Deep Learning on Graphs in Natural Language Processing and Computer Vision

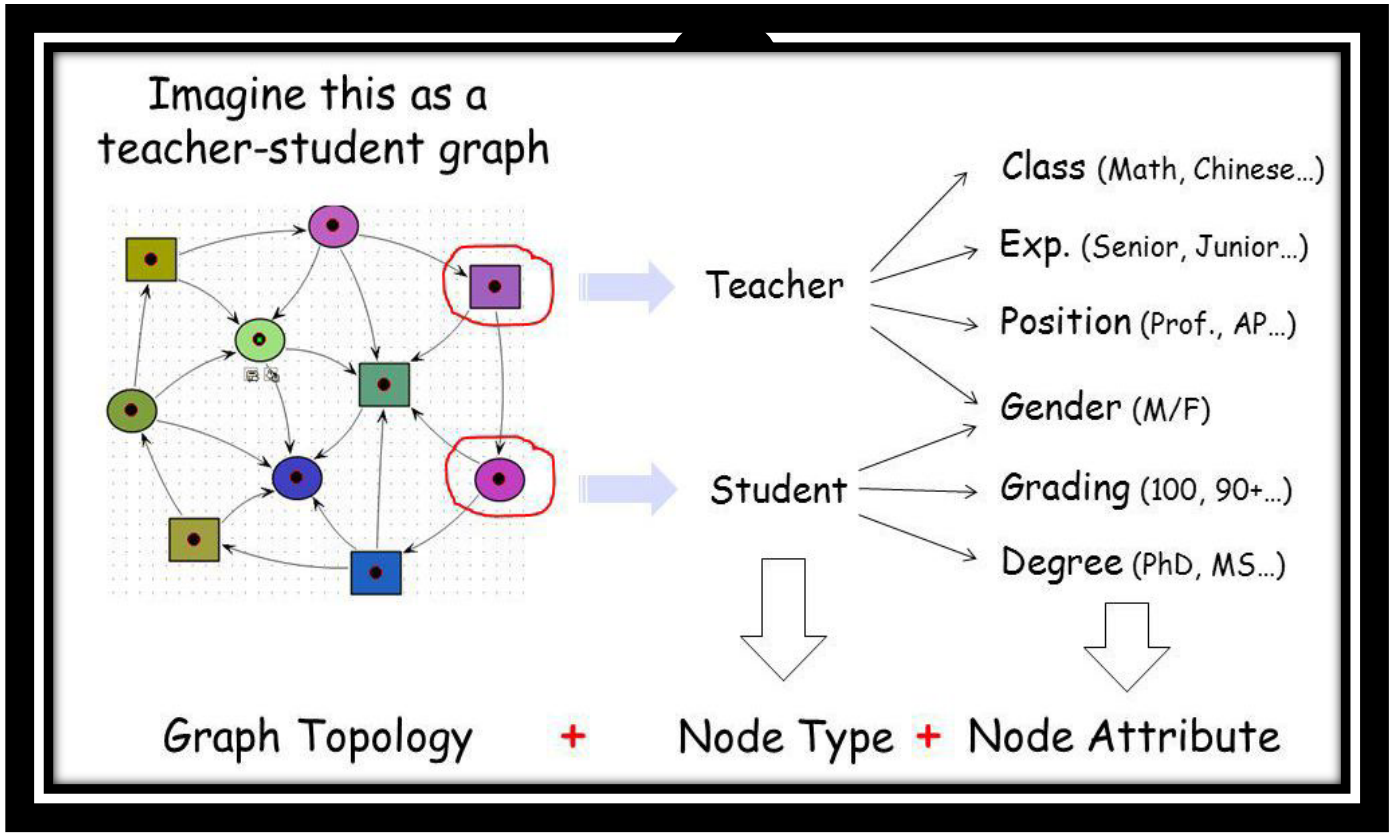
Lingfei Wu
IBM Research AI
IBM T. J. Watson Research Center

Joint work with Yu Chen, Mohammed J Zaki, Kai Shen, Fangli Xu, Siliang Tang, Jun Xiao and Yueting Zhuang

May 24, 2020

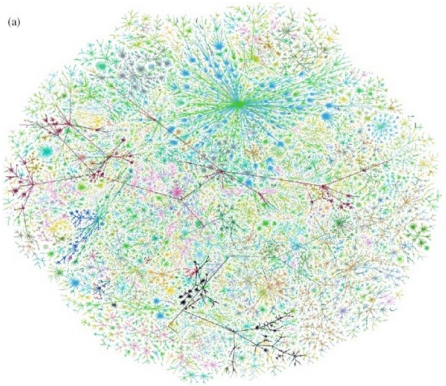
Why graphs?

- Graphs are a general language for describing and modeling complex systems

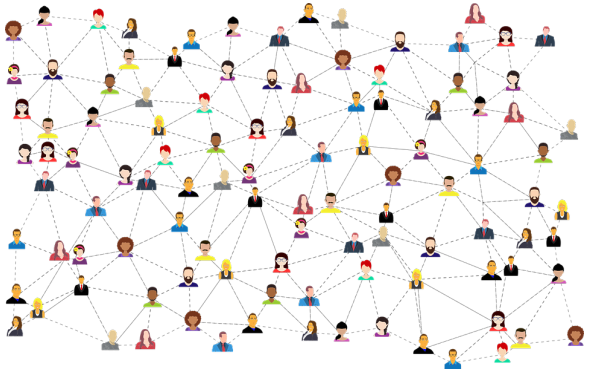


Graph!

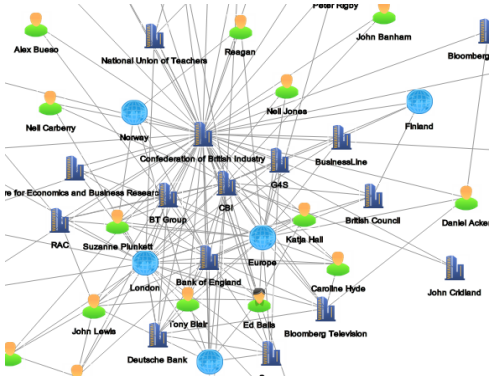
Graph-structured data are ubiquitous



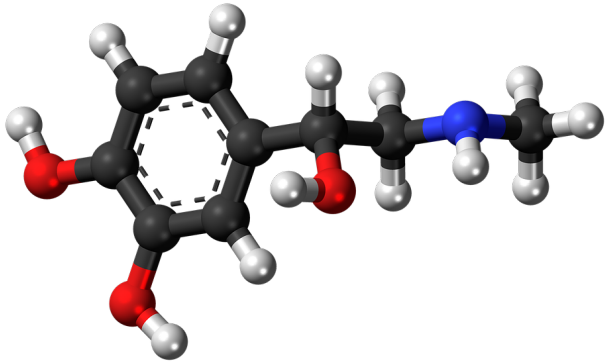
Internet



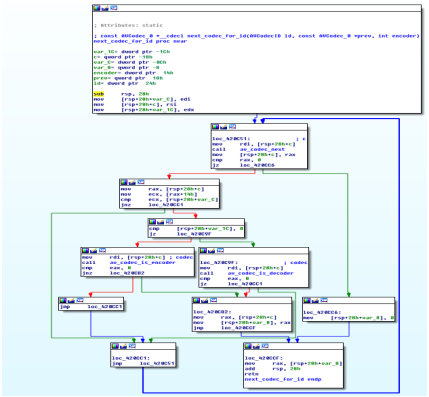
Social networks



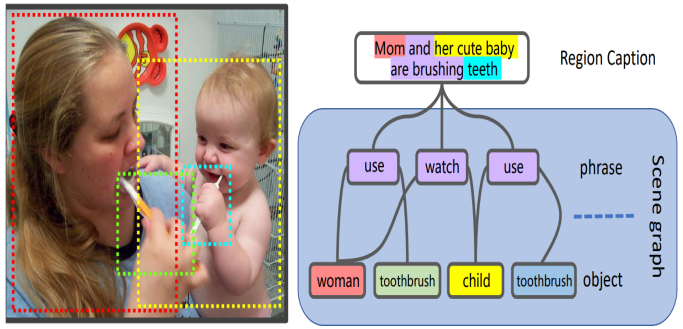
Networks of transactions



Biomedical graphs



Program graphs



Scene graphs

Machine(Deep) Learning with Graphs

Classical ML tasks in graphs:

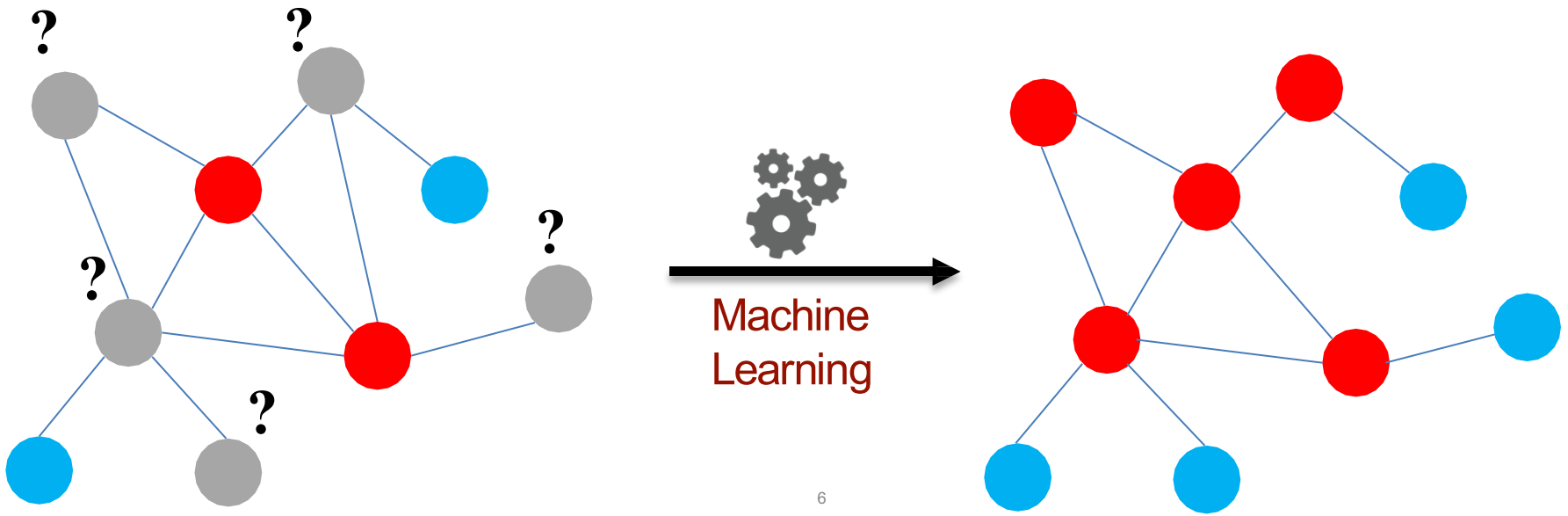
- Node classification
  - Predict a type of a given node
- Link prediction
  - Predict whether two nodes are linked
- Community detection
  - Identify densely linked clusters of nodes
- Graph matching (similarity)
  - How similar are two (sub)graphs

Recent ML tasks in graphs:

- Graph classification
  - Predict a type of a given graph
- Graph generation
  - Generate graphs from learned distribution
- Graph structure learning
  - Joint learn graph structure and graph embeddings
- Graph-to-X learning
  - Graph Inputs – X outputs

Deep Learning on Graphs
Basic Models

Node Classification Task



Node Classification: An Example

Classifying the function of proteins in the interactome!

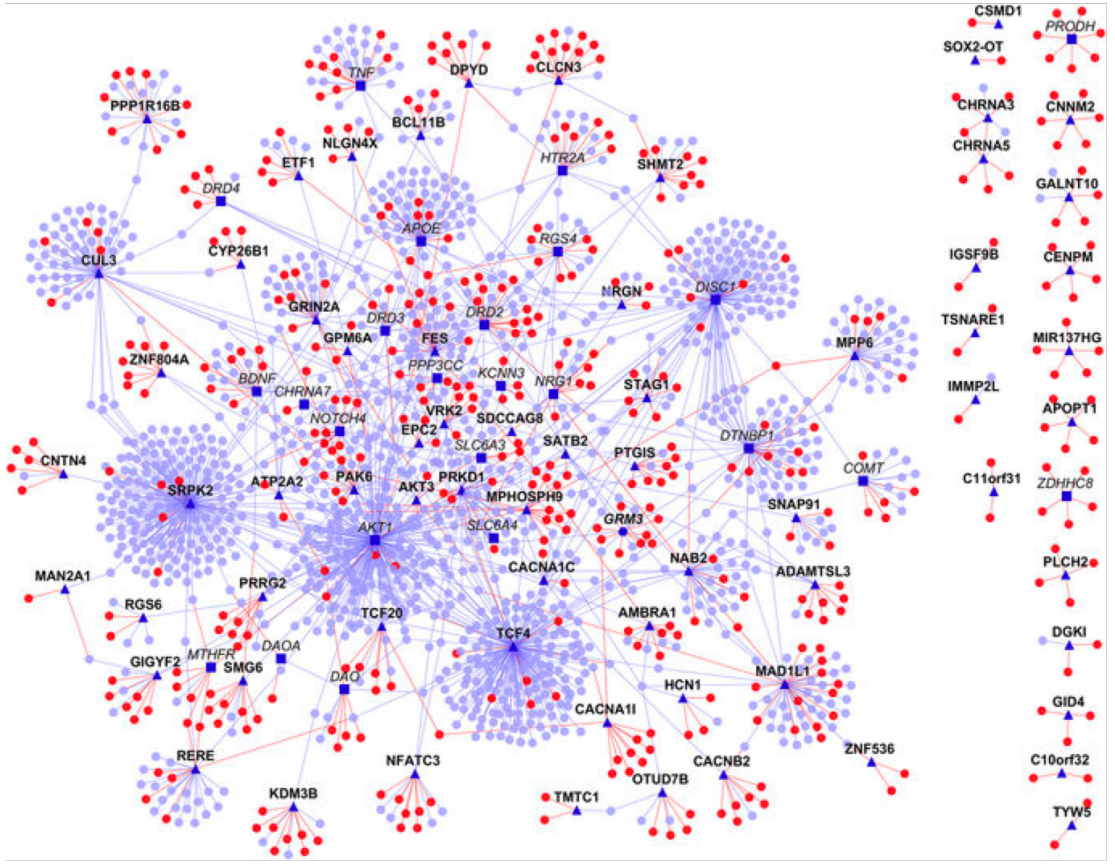
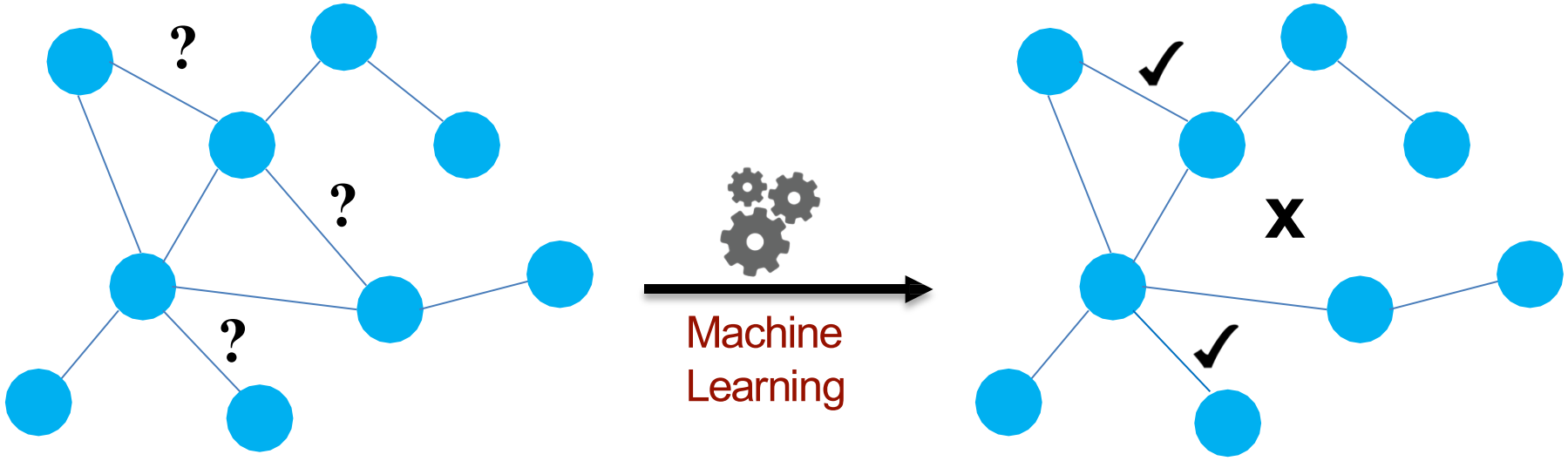


Image from: Ganapathiraju et al. 2016. Schizophrenia interactome with 504 novel protein–protein interactions. *Nature*.

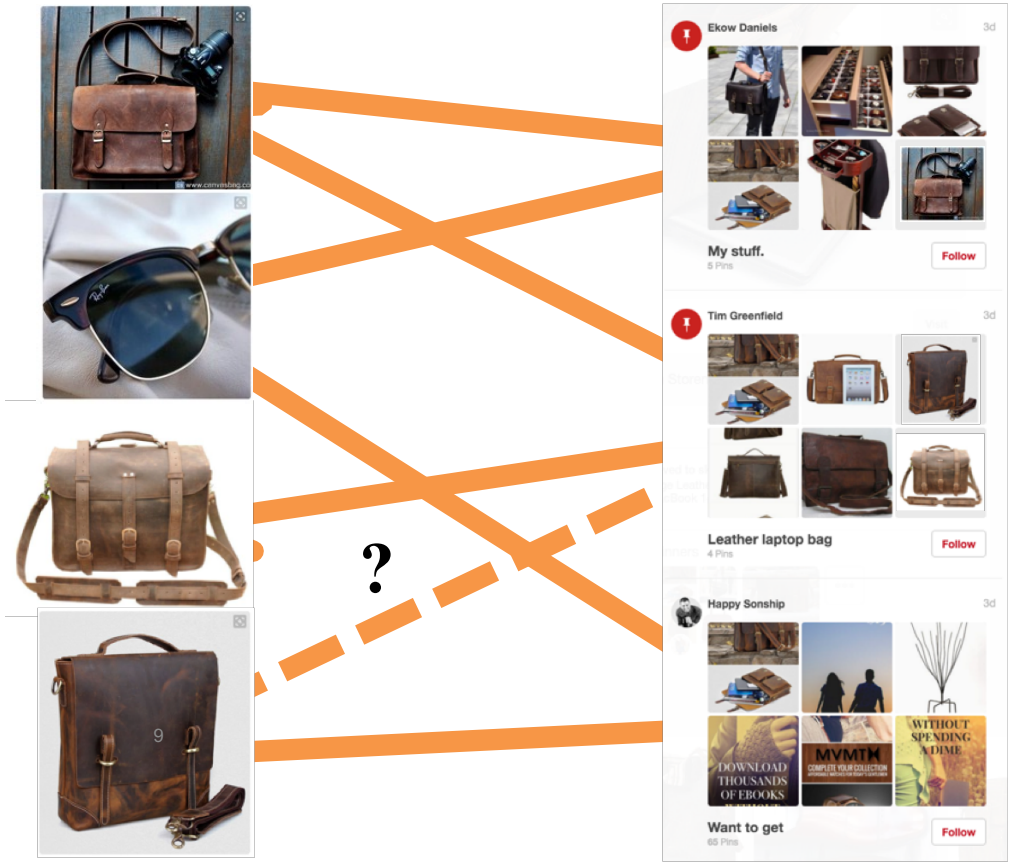
Link Prediction Task



8

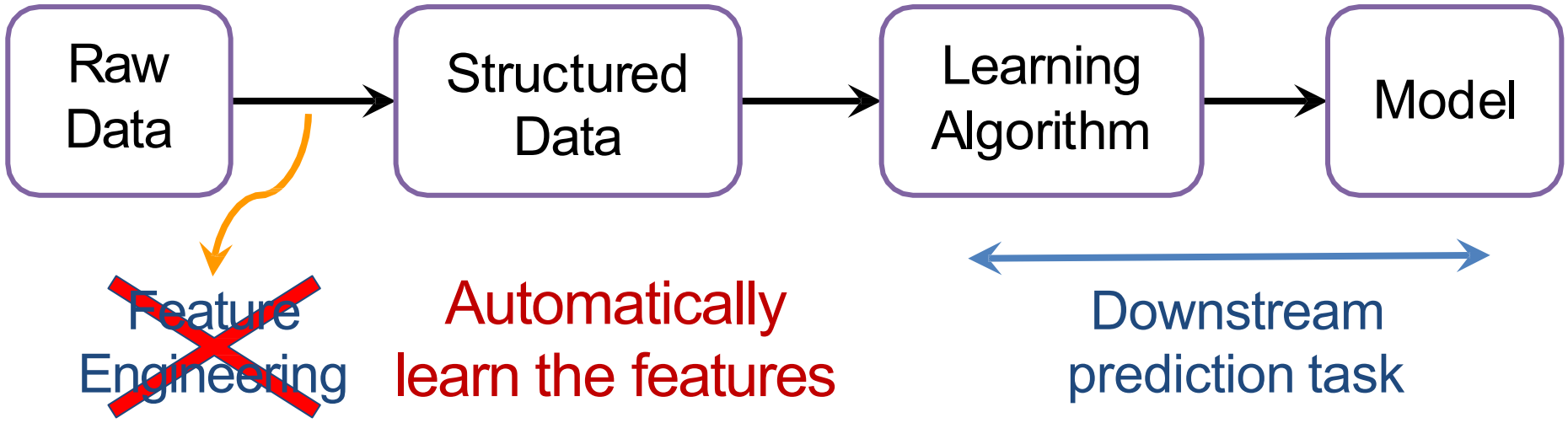
Link Prediction: An Example

Content recommendation is link prediction!



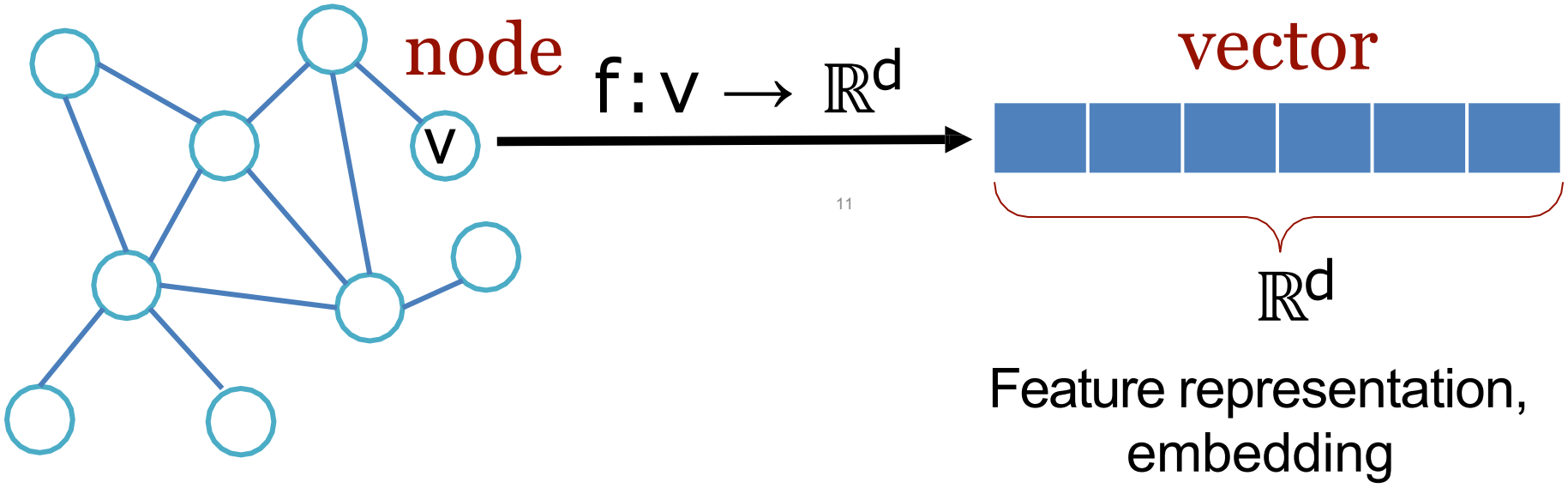
Machine Learning Lifecycle

- (Supervised) Machine Learning Lifecycle: feature learning is the key



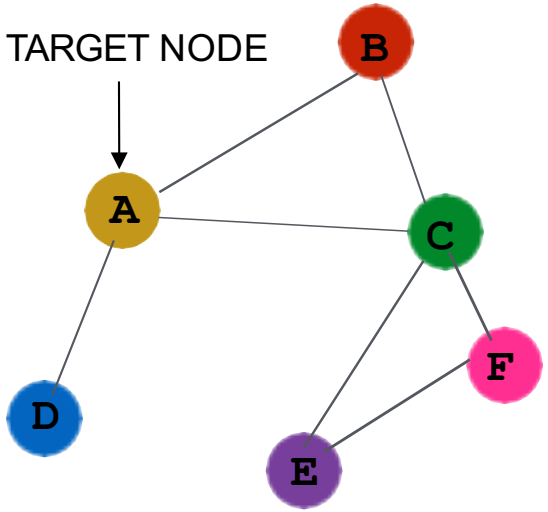
Feature Learning in Graphs

- Our Goal: Design efficient task-independent/ task-dependent feature learning for machine learning in graphs!

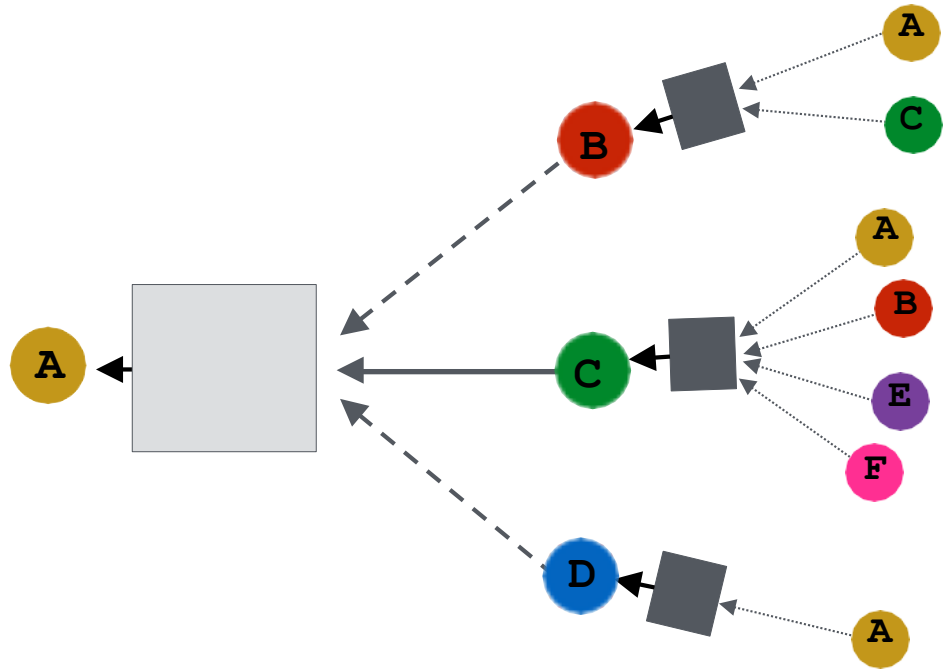


Graph Neural Networks

- Key idea: Generate node embeddings based on local neighborhoods.

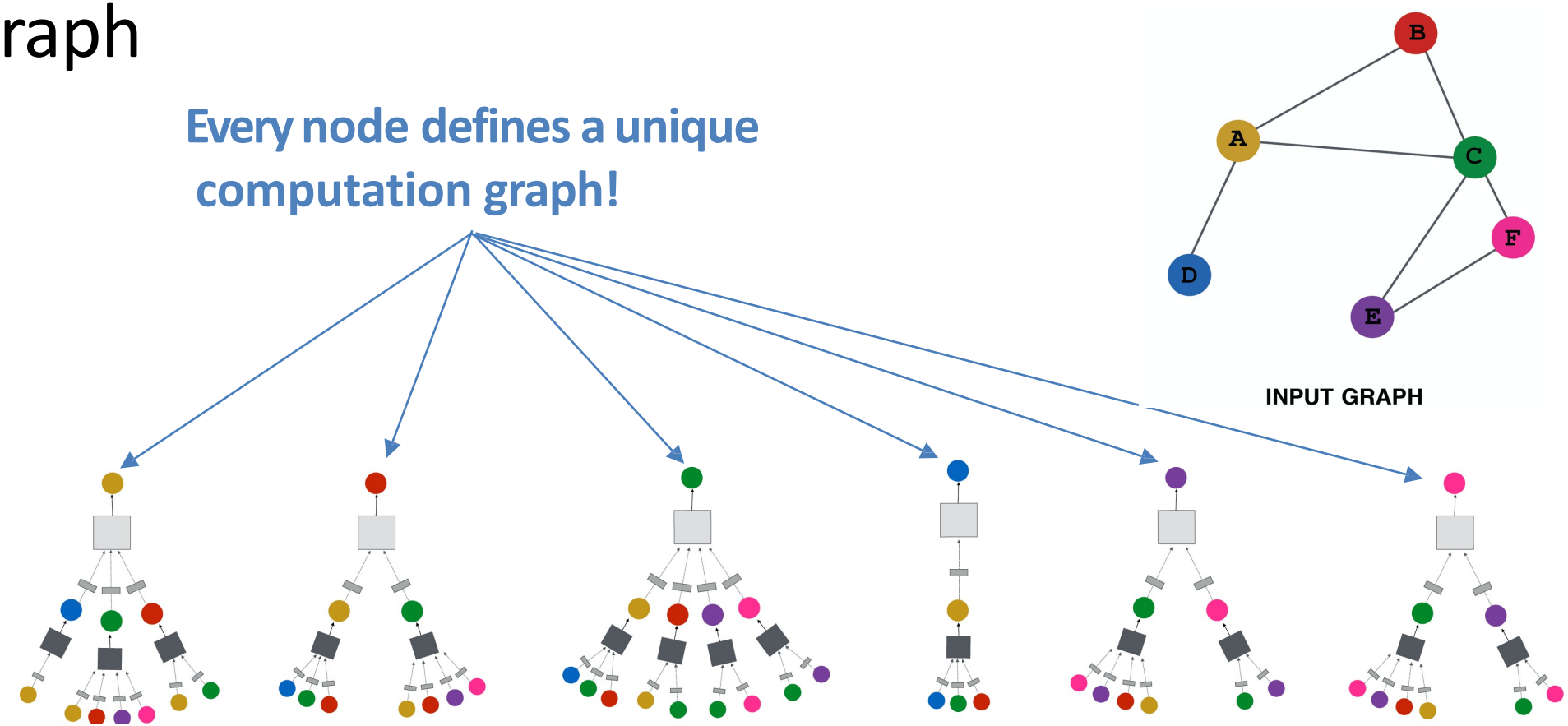


INPUT GRAPH



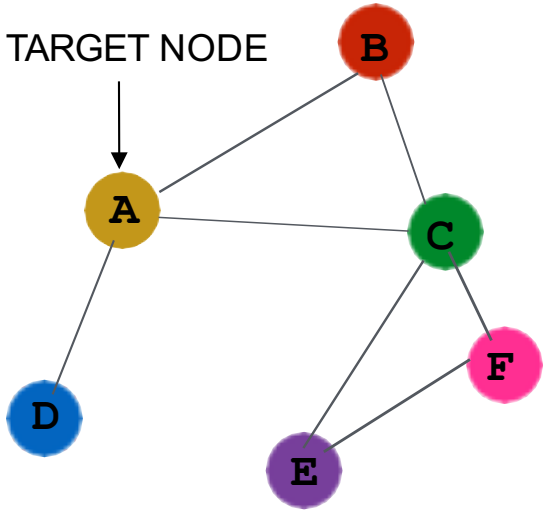
Neighborhood Aggregation

- Intuition: Network neighborhood defines a computation graph

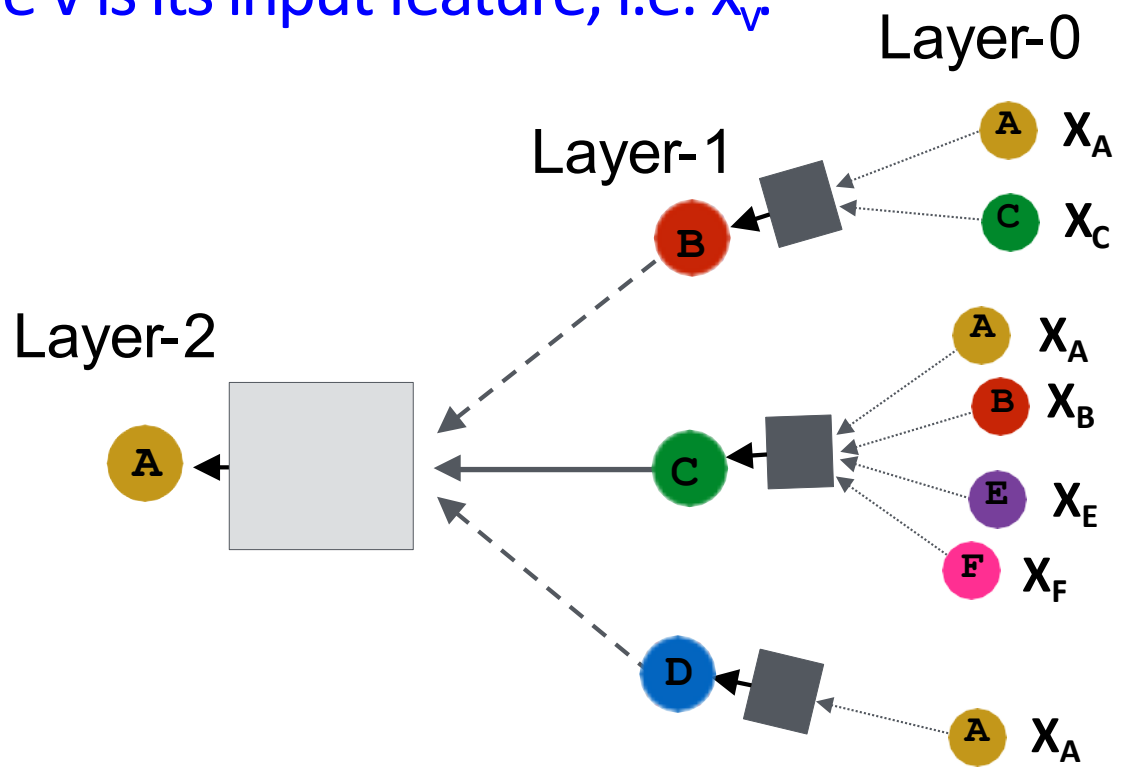


Neighborhood Aggregation

- Nodes have embeddings at each layer.
- Model can be arbitrary depth.
- "layer-0" embedding of node v is its input feature, i.e. x_v.

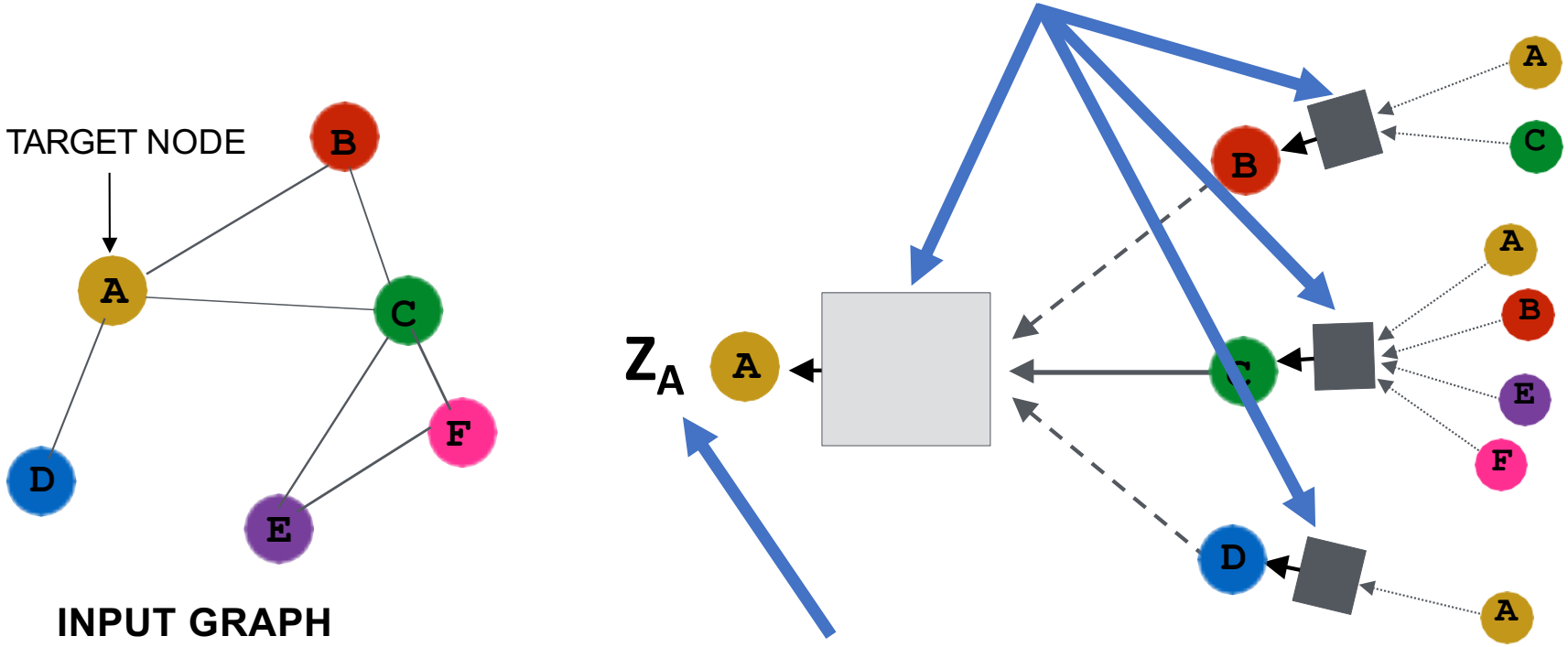


INPUT GRAPH



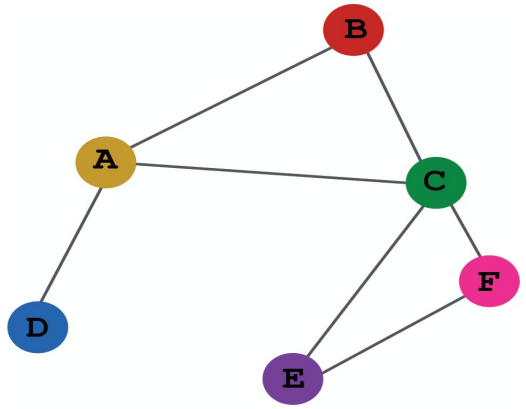
Overview of GNN Model

1) Define a neighborhood aggregation function.



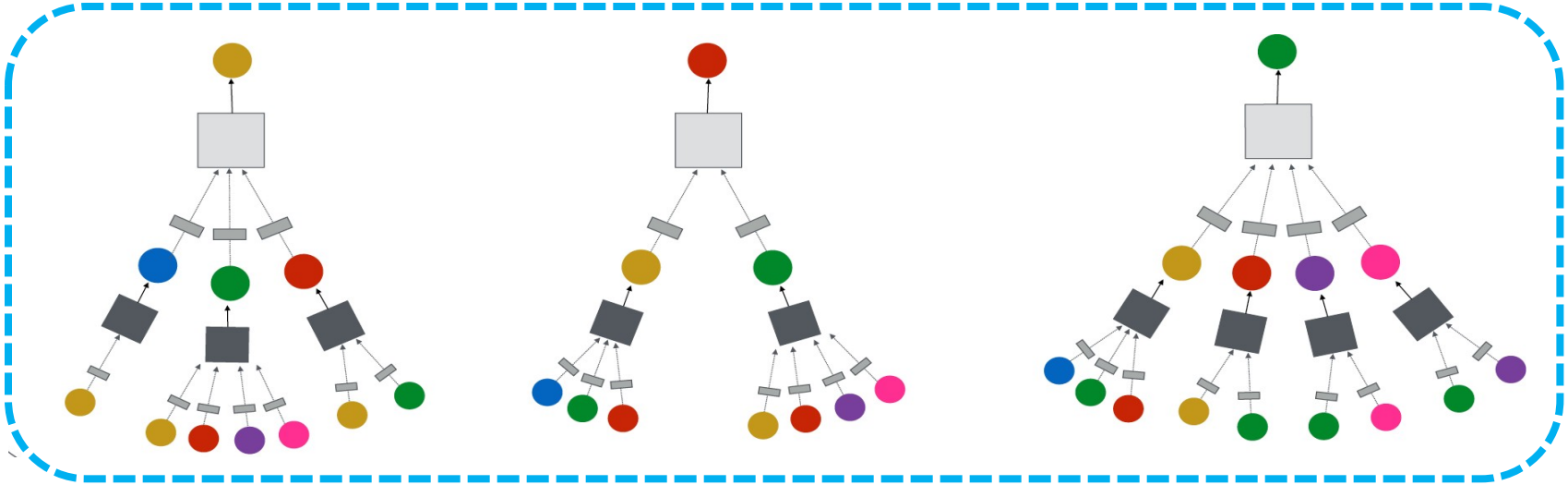
2) Define a loss function on the embeddings, $\mathbf{L(z_v)}$

Overview of GNN Model

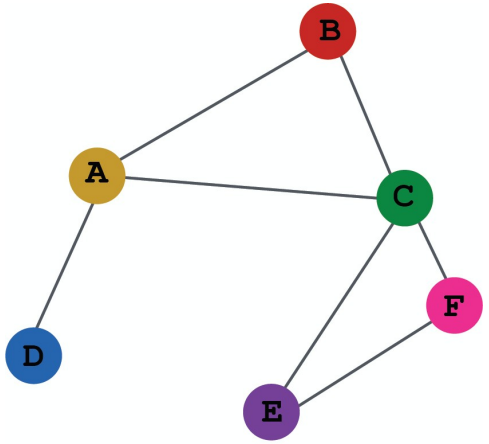


INPUT GRAPH

3) Train on a set of nodes, i.e., a batch of compute graphs



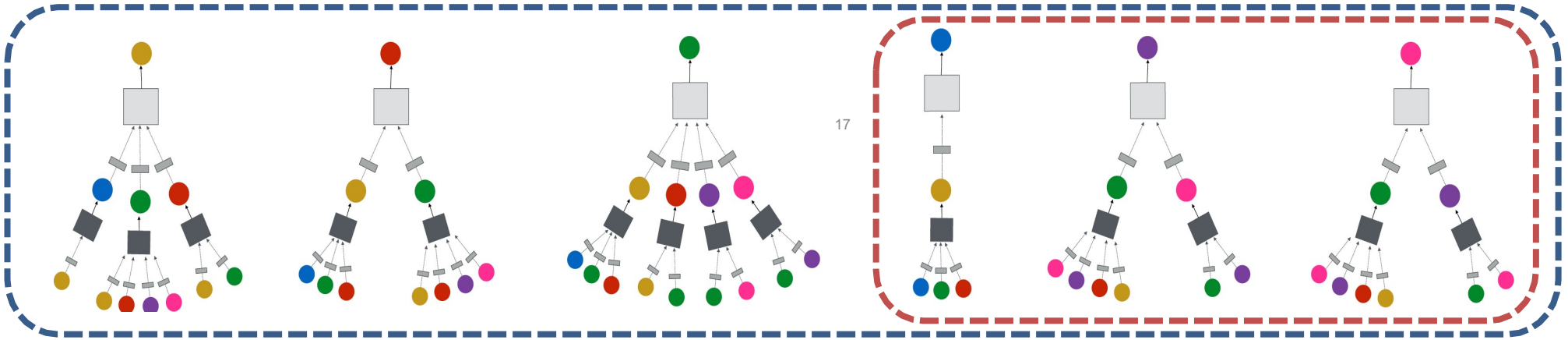
Overview of GNN Model



INPUT GRAPH

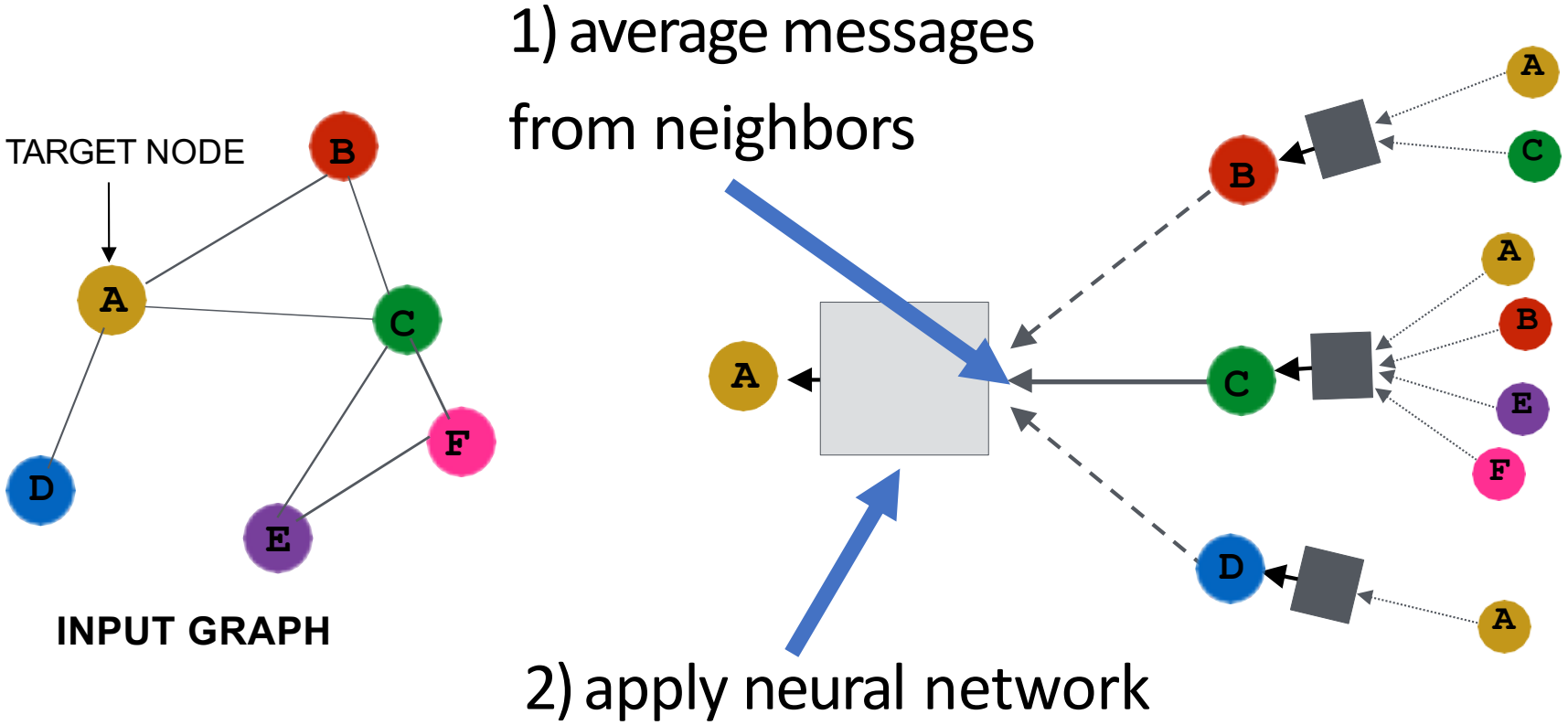
4) Generate embeddings for nodes as needed

Even for nodes we never trained on!



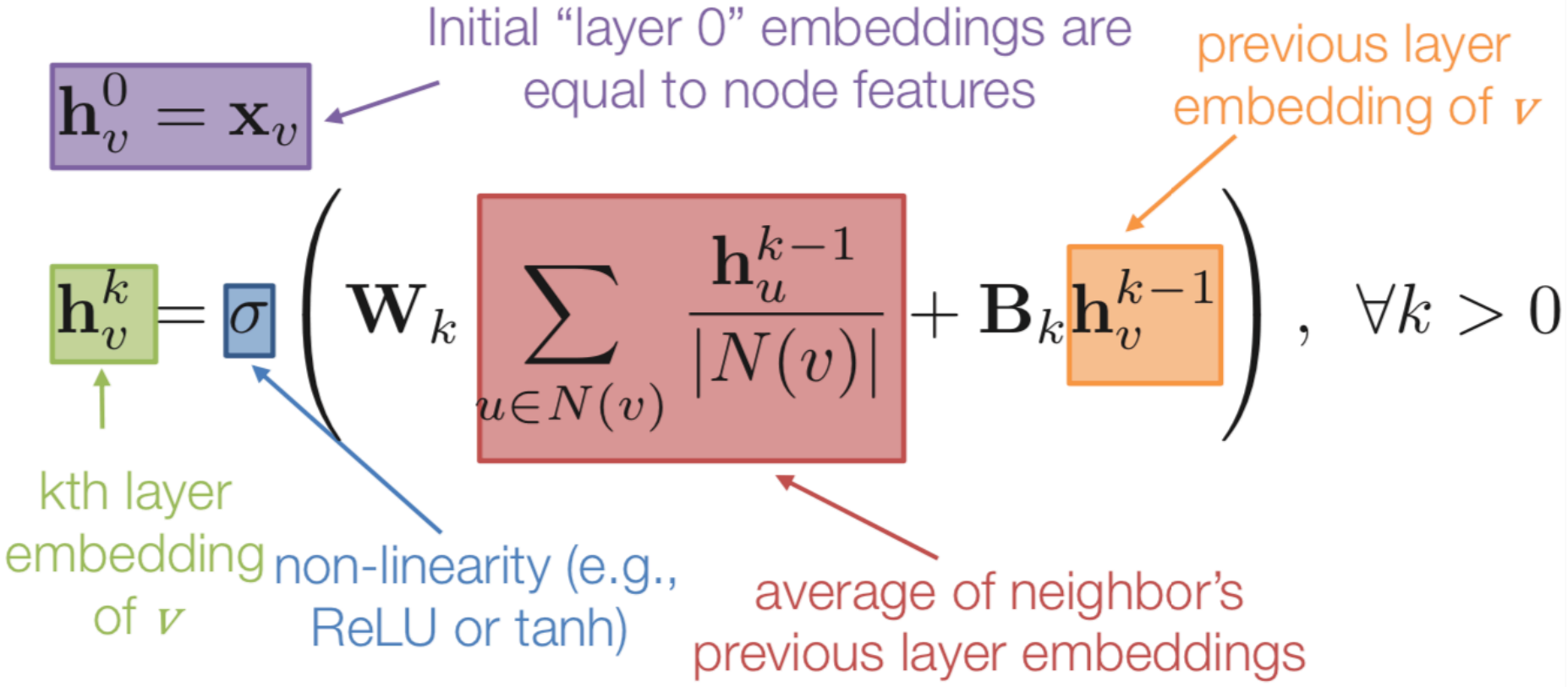
GNN Model: A Case Study

- Basic approach: Average neighbor information and apply a neural network.



GNN Model: A Case Study

- Basic approach: Average neighbor information and apply a neural network.



GNN Model: Quick Summary

- Key idea: generate node embeddings by aggregating neighborhood information.
  - Allows for parameter sharing in the encoder
  - Allows for inductive learning
- Other state of the art GNNs variants:
  - Graph convolutional networks
  - Graph attention networks
  - Graph isomorphism networks

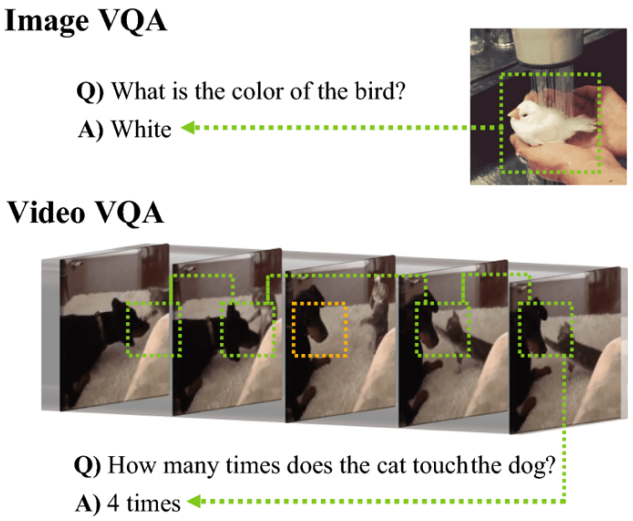
Graph2Seq + RL for Question Generation (ICLR'20)

Natural Question Generation: Background

- Natural question generation (QG) is a challenging yet rewarding task, that aims to generate questions given an input passage and a target answer.

SQuAD2.0
The Stanford Question Answering Dataset

- Many real applications:
  - Reading comprehension
  - Visual and video question answering
  - Dialog system



Natural Question Generation: Definition

- Input:
  - A text passage: $X^p = \{x_1^p, x_2^p, ..., x_N^p\}$
  - A target answer: $X^a = \{x_1^a, x_2^a, ..., x_L^a\}$

- Output:
  - The best natural language question: $\hat{Y} = \{y_1, y_2, ..., y_T\}$
    which maximizes the conditional likelihood: $\hat{Y} = \arg\max_Y P(Y|X^p, X^a)$

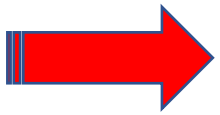
Existing State-of-the-art Methods

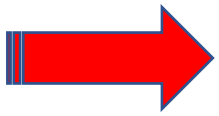
- Template-based approaches
  - Mostow & Chen, 2009; Heilman & Smith, 2010; Heilman, 2011
  - Rely on heuristic rules or hand-crafted templates
  - low generalizability and scalability

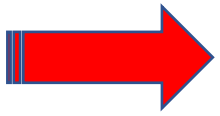
- Seq2Seq-based approaches
  - Du et al., 2017; Zhou et al., 2017; Song et al., 2018a; Kumar et al., 2018a
  - Fail to utilize the rich text structure information beyond the simple word sequence
  - Rely on cross-entropy based sequence training which has several limitations
  - Fail to effectively utilize the answer information

Known Issues of Existing Approaches

- Issue I: fail to consider global interactions between answer and context
- Issue II: fail to consider rich hidden structure information of word sequence
- Issue III: limitations of cross-entropy based objectives like exposure bias and inconsistency between train/test measurement

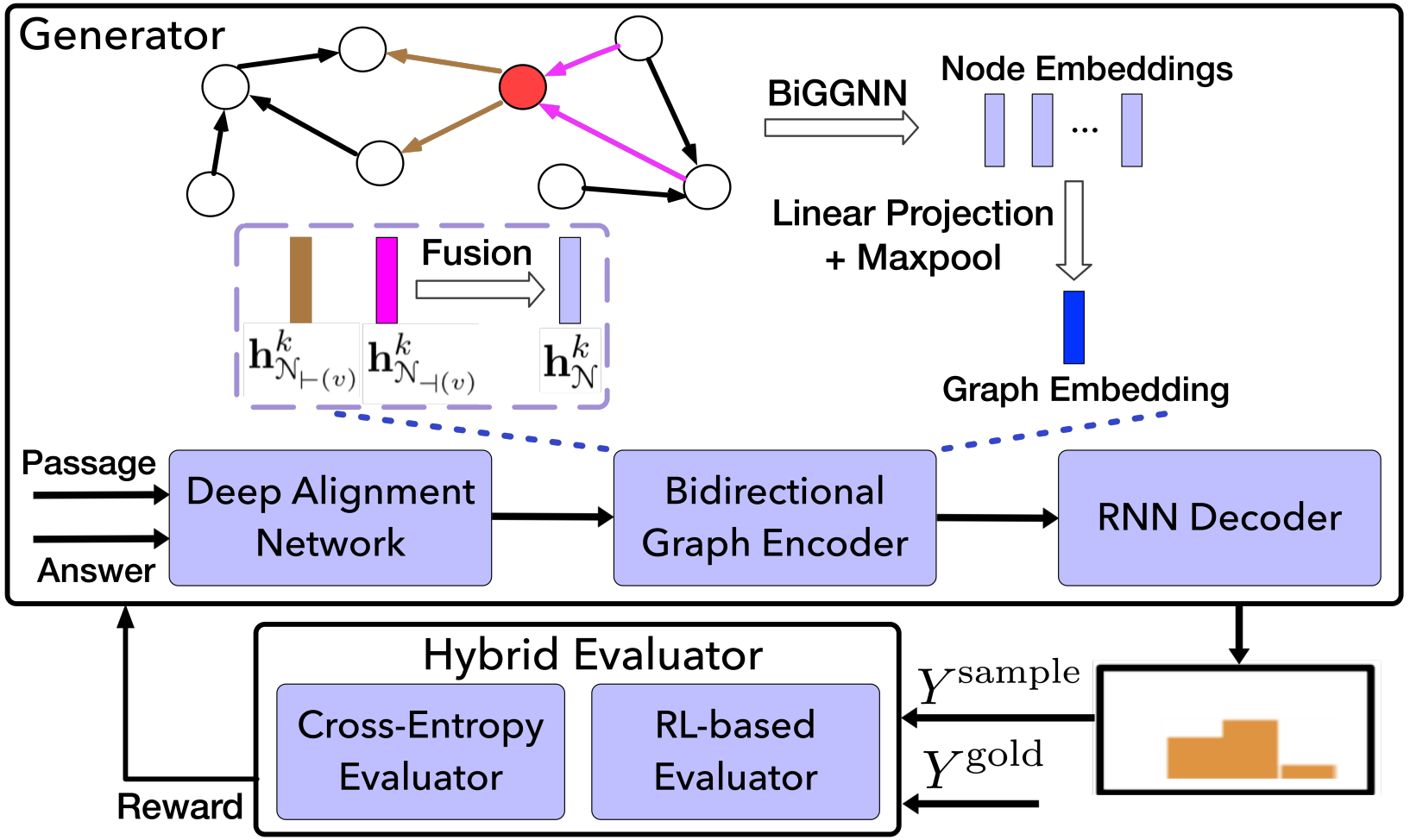






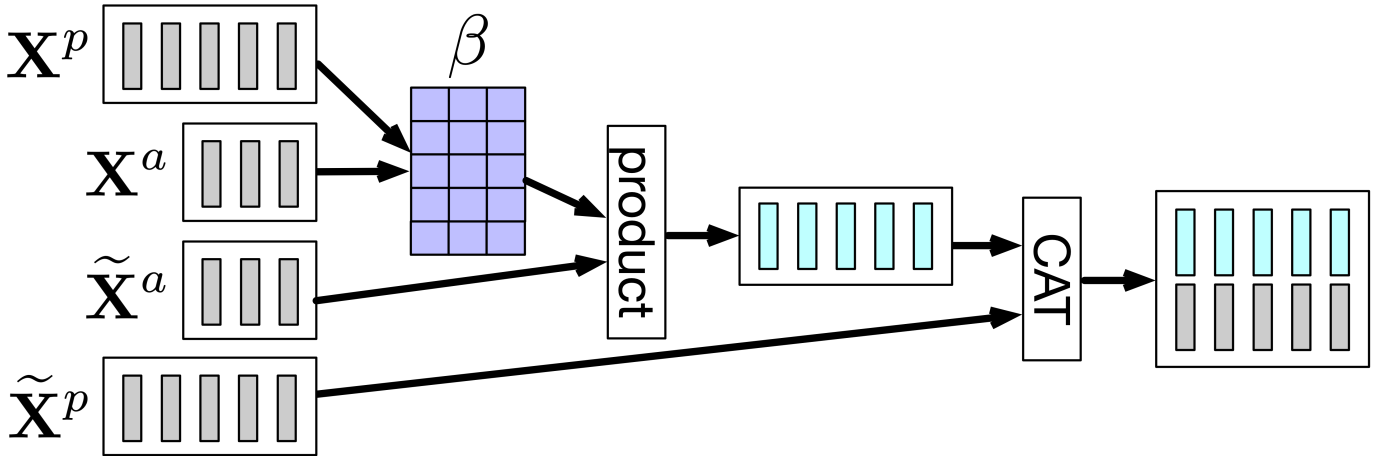
- Solution I: Deep alignment network to align answer and context
- Solution II: Novel Graph2Seq model for considering hidden structure information in sequence
- Solution III: Novel Reinforcement Learning Loss for enforcing syntactic and semantic coherent of generated text

RL-based Graph2Seq for QG: System Overview



Deep Answer Alignment

A deep alignment network for incorporating the answer information into passages at both the word level and the contextualized hidden state level



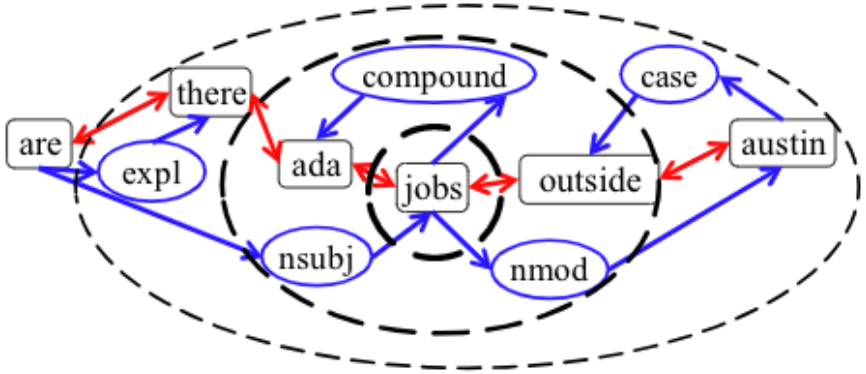
p denotes passage
a denotes answer

$$\widetilde{\mathbf{H}}^p = \text{Align}(\mathbf{X}^p, \mathbf{X}^a, \widetilde{\mathbf{X}}^p, \widetilde{\mathbf{X}}^a) = \text{CAT}(\widetilde{\mathbf{X}}^p; \mathbf{H}^p) = \text{CAT}(\widetilde{\mathbf{X}}^p; \widetilde{\mathbf{X}}^a \beta^T)$$

$$\beta \propto \exp\left(\text{ReLU}(\mathbf{W}\mathbf{X}^p)^T \text{ReLU}(\mathbf{W}\mathbf{X}^a)\right)$$

Graph Construction: Static VS Dynamic

- Syntax-based static graph
  - A directed and unweighted passage graph based on dependency parsing



- Semantics-aware dynamic graph
  - Dynamically build a directed and weighted graph to model semantic relationships among passage words

$$\mathbf{A} = \mathrm{ReLU}(\mathbf{U}\tilde{\mathbf{H}}^p)^T \mathrm{ReLU}(\mathbf{U}\tilde{\mathbf{H}}^p)$$

$$\bar{\mathbf{A}} = \mathrm{kNN}(\mathbf{A})$$

$$\mathbf{A}^{\dashv}, \mathbf{A}^{\vdash} = \mathrm{softmax}(\{\bar{\mathbf{A}}, \bar{\mathbf{A}}^T\})$$

$\tilde{\mathbf{H}}^p$ is the passage representation

Bidirectional Graph2seq Generator (I)

Node aggregation for the syntax-based static graph

$$\mathbf{h}^k_{\mathcal{N}_{\dashv(v)}} = \text{MEAN}(\{\mathbf{h}^{k-1}_v\} \cup \{\mathbf{h}^{k-1}_u, \forall u \in \mathcal{N}_{\dashv(v)}\})$$
$$\mathbf{h}^k_{\mathcal{N}_{\vdash(v)}} = \text{MEAN}(\{\mathbf{h}^{k-1}_v\} \cup \{\mathbf{h}^{k-1}_u, \forall u \in \mathcal{N}_{\vdash(v)}\})$$

Node aggregation for the semantics-based dynamic graph

$$\mathbf{h}^k_{\mathcal{N}_{\dashv(v)}} = \sum_{\forall u \in \mathcal{N}_{\dashv(v)}} \mathbf{a}^{\dashv}_{v,u} \mathbf{h}^{k-1}_u, \quad \mathbf{h}^k_{\mathcal{N}_{\vdash(v)}} = \sum_{\forall u \in \mathcal{N}_{\vdash(v)}} \mathbf{a}^{\vdash}_{v,u} \mathbf{h}^{k-1}_u$$

Bidirectional Graph2seq Generator (II)

Fuse the aggregated node embeddings from both directions

$$\mathbf{h}_{\mathcal{N}}^{k} = \text{Fuse}(\mathbf{h}_{\mathcal{N}_{\dashv(v)}}^{k}, \mathbf{h}_{\mathcal{N}_{\vdash(v)}}^{k})$$

$$\text{Fuse}(\mathbf{a}, \mathbf{b}) = \mathbf{z} \odot \mathbf{a} + (1 - \mathbf{z}) \odot \mathbf{b}$$

$$\mathbf{z} = \sigma(\mathbf{W}_z[\mathbf{a}; \mathbf{b}; \mathbf{a} \odot \mathbf{b}; \mathbf{a} - \mathbf{b}] + \mathbf{b}_z)$$

Update the node embeddings using fused information

$$\mathbf{h}_{v}^{k} = \text{GRU}(\mathbf{h}_{v}^{k-1}, \mathbf{h}_{\mathcal{N}}^{k})$$

Hybrid Evaluator

- Regular cross-entropy based training objectives have limitations
  - Exposure bias
  - Evaluation discrepancy between training and testing
- We apply a mixed objective function combining both the cross-entropy loss and RL loss
  - Ensure the generation of syntactically and semantically valid text

$$\mathcal{L}_{lm} = \sum_t -\log P(y_t^*|X, y_{<t}^*) + \lambda \,\text{covloss}_t \qquad \mathcal{L}_{rl} = (r(\hat{Y}) - r(Y^s)) \sum_t \log P(y_t^s|X, y_{<t}^s)$$

$$\mathcal{L} = \gamma \mathcal{L}_{rl} + (1-\gamma)\mathcal{L}_{lm}$$

- Two-stage training strategy:
  - Train the model with cross-entropy loss
  - Finetune the model by optimizing the mixed objective function

Automatic Evaluation Results

Table 1: Automatic evaluation results on the SQuAD test set.

Methods	Split-1				Split-2			
	BLEU-4	METEOR	ROUGE-L	Q-BLEU1	BLEU-4	METEOR	ROUGE-L	Q-BLEU1
SeqCopyNet	–	–	–	–	13.02	–	44.00	–
NQG++	–	–	–	–	13.29	–	–	–
MPQG+R*	14.39	18.99	42.46	52.00	14.71	18.93	42.60	50.30
AFPQA	–	–	–	–	15.64	–	–	–
s2sa-at-mp-gsa	15.32	19.29	43.91	–	15.82	19.67	44.24	–
ASs2s	16.20	19.92	43.96	–	16.17	–	–	–
CGC-QG	–	–	–	–	17.55	21.24	44.53	–
G2S$_{dyn}$+BERT+RL	17.55	21.42	45.59	55.40	18.06	21.53	45.91	55.00
G2S$_{sta}$+BERT+RL	**17.94**	**21.76**	**46.02**	**55.60**	**18.30**	**21.70**	**45.98**	**55.20**

Human Evaluation and Ablation Study Results

Table 2: Human evaluation results ($\pm$ standard deviation) on the SQuAD split-2 test set. The rating scale is from 1 to 5 (higher scores indicate better results).

Methods	Syntactically correct	Semantically correct	Relevant
MPQG+R*	4.34 (0.15)	4.01 (0.23)	3.21 (0.31)
G2S$_{sta}$+BERT+RL	4.41 (0.09)	4.31 (0.12)	3.79 (0.45)
Ground-truth	**4.74 (0.14)**	**4.74 (0.19)**	**4.25 (0.38)**

Table 3: Ablation study on the SQuAD split-2 test set.

Methods	BLEU-4	Methods	BLEU-4
G2S$_{dyn}$+BERT+RL	18.06	G2S$_{dyn}$	16.81
G2S$_{sta}$+BERT+RL	18.30	G2S$_{sta}$	16.96
G2S$_{sta}$+BERT-fixed+RL	18.20	G2S$_{dyn}$ w/o DAN	12.58
G2S$_{dyn}$+BERT	17.56	G2S$_{sta}$ w/o DAN	12.62
G2S$_{sta}$+BERT	18.02	G2S$_{sta}$ w/o BiGGNN, w/ Seq2Seq	16.14
G2S$_{sta}$+BERT-fixed	17.86	G2S$_{sta}$ w/o BiGGNN, w/ GCN	14.47
G2S$_{dyn}$+RL	17.18	G2S$_{sta}$ w/ GGNN-forward	16.53
G2S$_{sta}$+RL	17.49	G2S$_{sta}$ w/ GGNN-backward	16.75

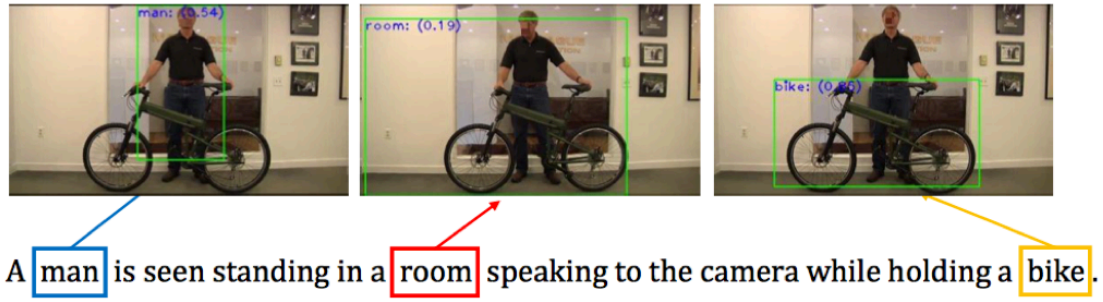
Spatial-Temporal Graph2Seq for Grounded Video Description(IJCAI'20)

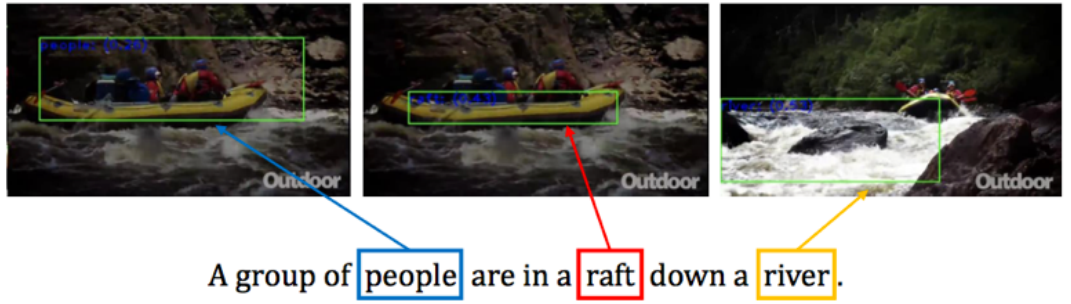
Grounded Video Description Task

Input: A video clip

Output: A sentence describing the content of the video.

The sentence's noun-phrases can be grounded with the bounding boxes in the videoframes.



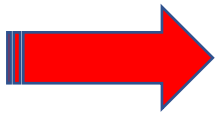


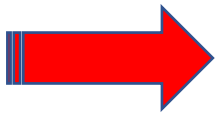
Existing State-of-the-art Methods

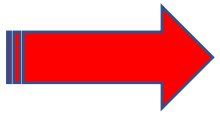
- Grounded Video Description [Zhou et al., 2019]
  - First work that link the words with the region proposals to generate descriptions more grounded
  - The self-attention method modeling the regions contains noise.
  - And all regions are attended equally and individually.
- Exploring visual relationship for image captioning [Yao et al., 2018]
  - They use a pre-trained relation classifier to build a semantic graph to model the regions.
  - The graph could be noisy
- STAGE: Spatial-Temporal Attention on Graph Entities for Video Action Detection [Tomei et al., 2019]
  - They use a self-attention method to build a semantic graph
  - They just use a single frame to model the whole video

Known Issues of Existing Approaches

- Issue I: Encode region proposals independently and fail to consider hidden structural information among the region proposals
- Issue II: Self-attention-based methods need to handle noisy or fake relationships among these objects
- Issue III: the explicit structural features of objects (e.g. spatial, temporal, semantic) are overlooked

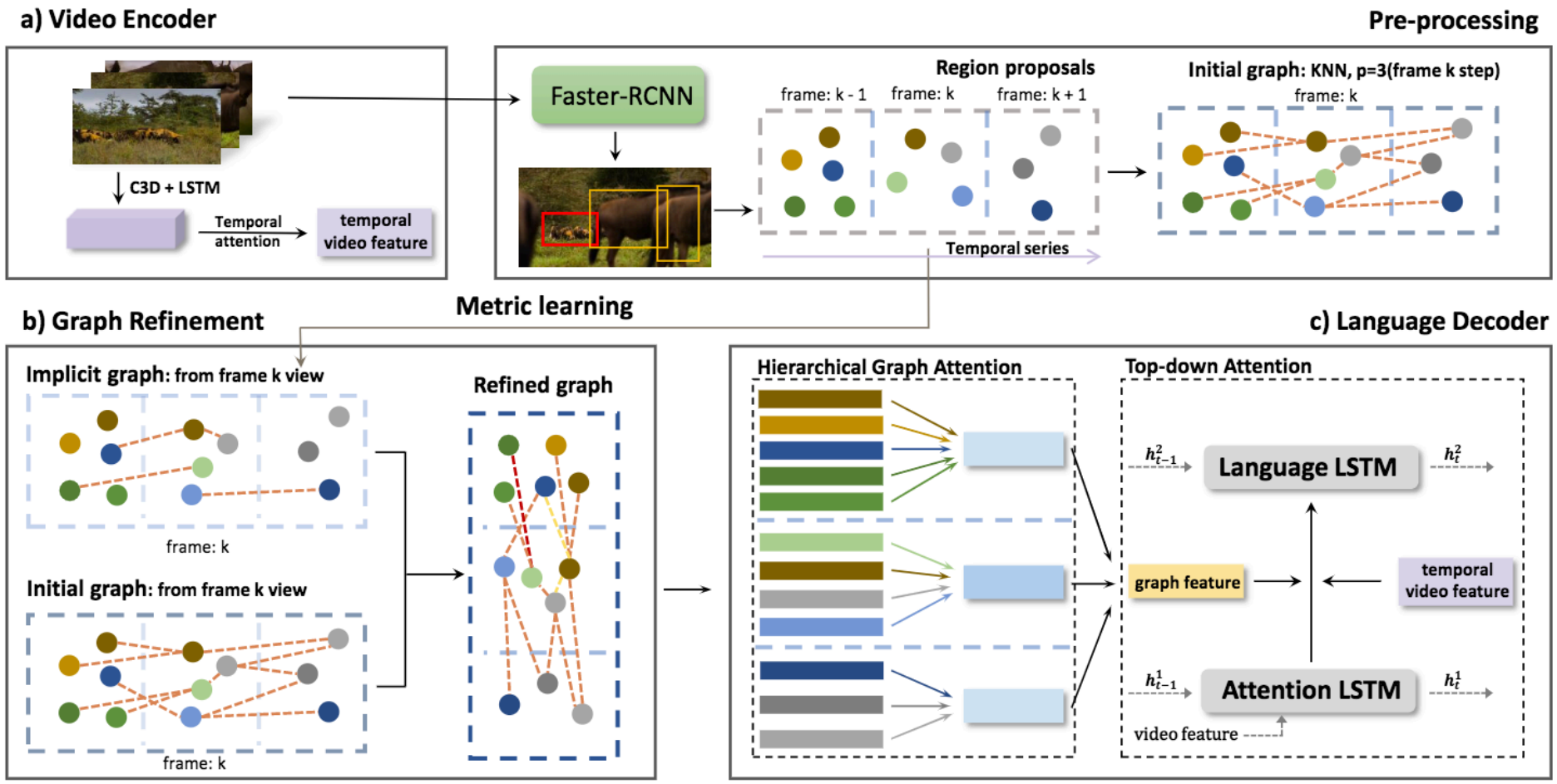




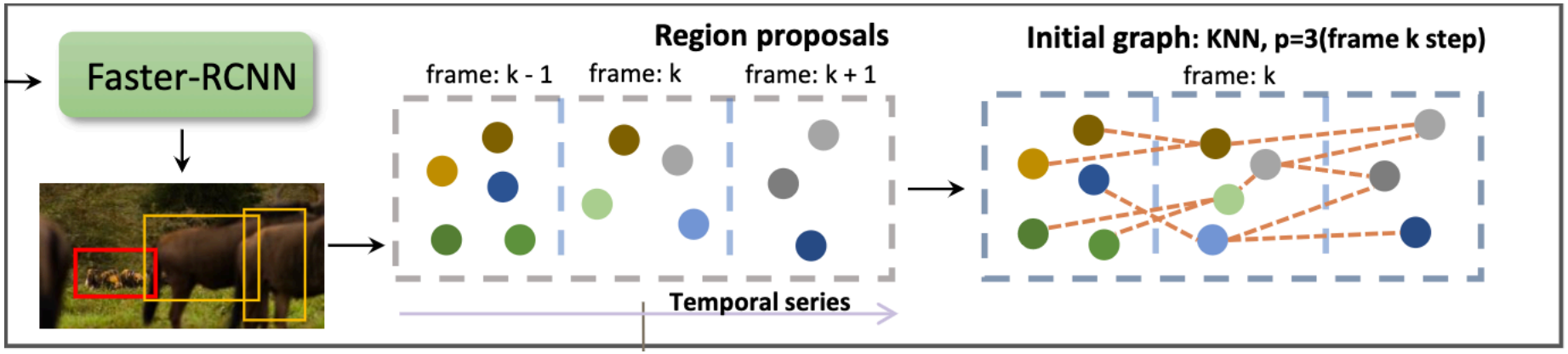


- Solution I: Graph2Seq model to learn relationships among region proposals between input video and output text sequence
- Solution II: KNN graph construction and graph refinement help learn optimized graph among the objects
- Solution III: Exploit spatial-temporal information and leverage hierarchical attention for focusing on different levels of semantics

Hierarchical Attention based Spatial-Temporal Graph2Seq for GVD: System Overview



Spatial-temporal Sequence Graph Construction



- Without external knowledge: KNN. For each node $r_i \in v_f$, we will find p nearest nodes in v_{f-1}, v_f, v_{f+1} and add edges between them.

- With external knowledge method: Relation Graph. Replace the KNN algorithm with relation classifier pre-trained on dataset like Visual Genome

Graph Learning and Refinement

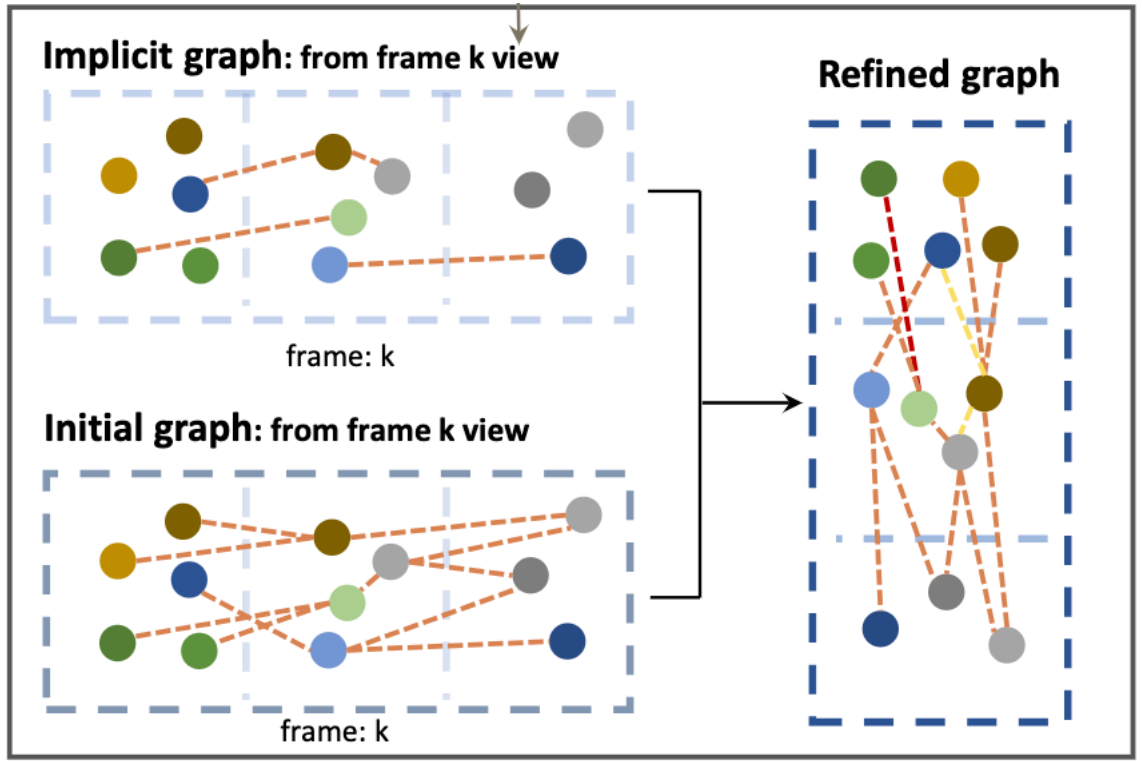
- Refine graph structure and fuse the learnt graph and initial graph

$$s_{i,j} = \frac{1}{m}\sum_{k=1}^{m}\cos(w^k \odot r_i, w^k \odot r_j)$$

$$A_{hidden}: \quad A_{i,j} = \begin{cases} s_{i,j} & if \quad s_{i,j} > \varepsilon \\ 0 & if \quad s_{i,j} \leq \varepsilon \end{cases}$$

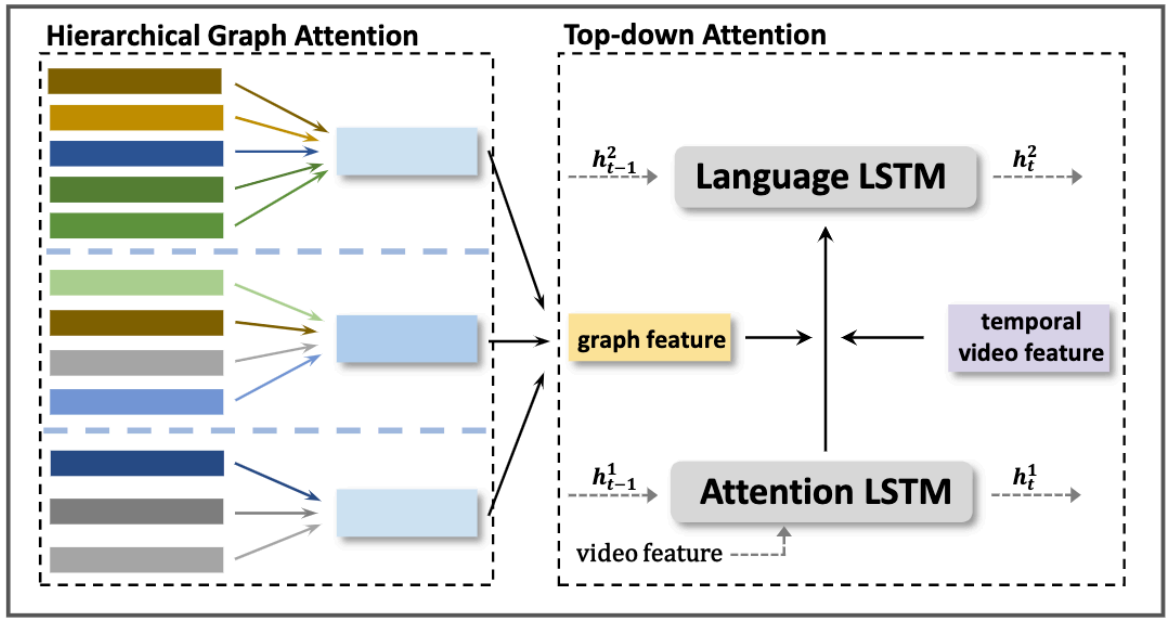
$$\tilde{A}_{dir} = \lambda\, \hat{A}_{initial} + (1 - \lambda)\, \hat{A}_{hidden}$$

$$\tilde{A} = \frac{\tilde{A}_{dir} + \tilde{A}_{dir}^T}{2}$$



Hierarchical Graph Attention for Text Generation

- Sub-graph level:
  - represent the sub-graph as a vector: $\tilde{R}_i = MeanPooling(\tilde{r}_{k:l})$
  - calculate sub-graph attention score: $\alpha = \{\alpha_1, \ldots, \alpha_F\}$
- Node level:
  - Calculate the node score in each sub-graph parallelly: $\beta_i = \{\beta_{i,1}, \ldots, \beta_{i,N_i}\}$



Automatic Experimental Results

Table 1: Results on Grounded ActivityNet-Entities test set. Notations: B@4-BLEU@4, C-CIDEr, M-METEOR, S-SPICE, M.Trans-Masked Transformer, TempAttn-BiLSTM+TempoAttn. All accuracies are in %.

Method	B@4	C	M	S	F1_all	F1_loc
M. Trans	2.41	46.1	10.6	13.7	-	-
Temp-Attn	2.17	42.2	10.2	11.8	-	-
ZhouGVD	2.35	45.5	11.0	14.7	7.59	25.0
KNN-HAST	2.61	48.5	**11.3**	**15.1**	7.64	**26.5**
RG-HAST	**2.65**	**49.3**	11.2	**15.2**	**7.66**	26.1

Effect on Different KNN Initial Graph

Table 2: Results on Grounded ActivityNet-Entities val set.

p (KNN)	B@4	C	M	S	F1_all	F1_loc
5	2.70	49.4	11.2	15.2	7.04	23.5
10	**2.80**	**49.6**	**11.3**	**15.3**	**7.22**	**24.9**
20	2.76	49.4	**11.3**	15.2	6.91	23.4
30	2.71	49.3	11.2	14.9	6.89	23.5
40	2.68	48.9	11.1	15.1	6.70	23.2

Ablation Study Results

Table 3: Results on Ablation Model on ActivityNet val set.

Method	B@4	C	M	S	F1_all	F1_loc
ZhouGVD	2.59	47.5	11.2	15.1	7.11	24.1
KNN-HAST	**2.80**	**49.4**	**11.3**	**15.3**	**7.22**	**24.9**
-init.	2.60	47.4	11.0	14.7	6.65	22.4
-refine	2.70	48.1	11.1	14.8	6.83	23.5
-hie. attn.	2.70	48.8	11.2	15.0	6.91	23.7

Conclusion

Take-home Messages from This Talk

- Deep Learning on Graphs is a fast-growing area today.
- Since graph can naturally encode complex information, it could bridge a gap by combining both empirical domain knowledges and the power of deep learning.

- Many on-going research projects of GNN in Graph4AI:
  - Scalable Node Embedding/Graph-level Embedding
  - Deep Graph Structure Learning
  - Graph Matching Networks
  - Dynamic/Incremental Graph Embedding
  - Graph2seq, Graph2Tree, Graph2Graph

Call for Papers on The Second International Workshop on Deep Learning on Graphs: Methods and Applications (DLG-KDD'20) August 24th, 2020, San Diego, CA, USA (in conjunction with KDD 2020, https://deep-learning-graphs.bitbucket.io/dlg-kdd20/)

- **Topic of interest (including but not limited to):**
- Graph neural networks on node-level, graph-level embedding
- Dynamic/incremental graph-embedding
- Graph neural networks on graph matching
- Deep graph structured learning
- Deep generative models for graph generation/semantic-preserving transformation
- Graph2seq, graph2tree, and graph2graph models
- Deep reinforcement learning on graphs
- Adversarial machine learning on graphs

- **And with particular focuses but not limited to these application domains:**
- learning and reasoning (machine reasoning, inductive logic programming, theory proving)
- natural language processing (information extraction, semantic parsing, text generation, machine comprehension)
- reinforcement learning (multi-agent learning, compositional imitation learning)
- program synthesis and analysis
- bioinformatics (drug discovery, protein generation, etc.)

Thank you!