Embedded Multi-Person Pedestrian Tracking and Detection

MSCV19 Capstone Project, Internal(CMU)

Team Member: Yongxin (Richard) Wang, Chunhui Liu
Advisor: Professor. Kris Kitani

09/20/2019

Introduction

- Problem
    - Detect and track multiple people
        - Tracking existing people
        - Handles new people and disappearing ones
- Our goal
    - Single stage network for detection and tracking
        - Extending SiameseRPN for Multi-Object Tracking

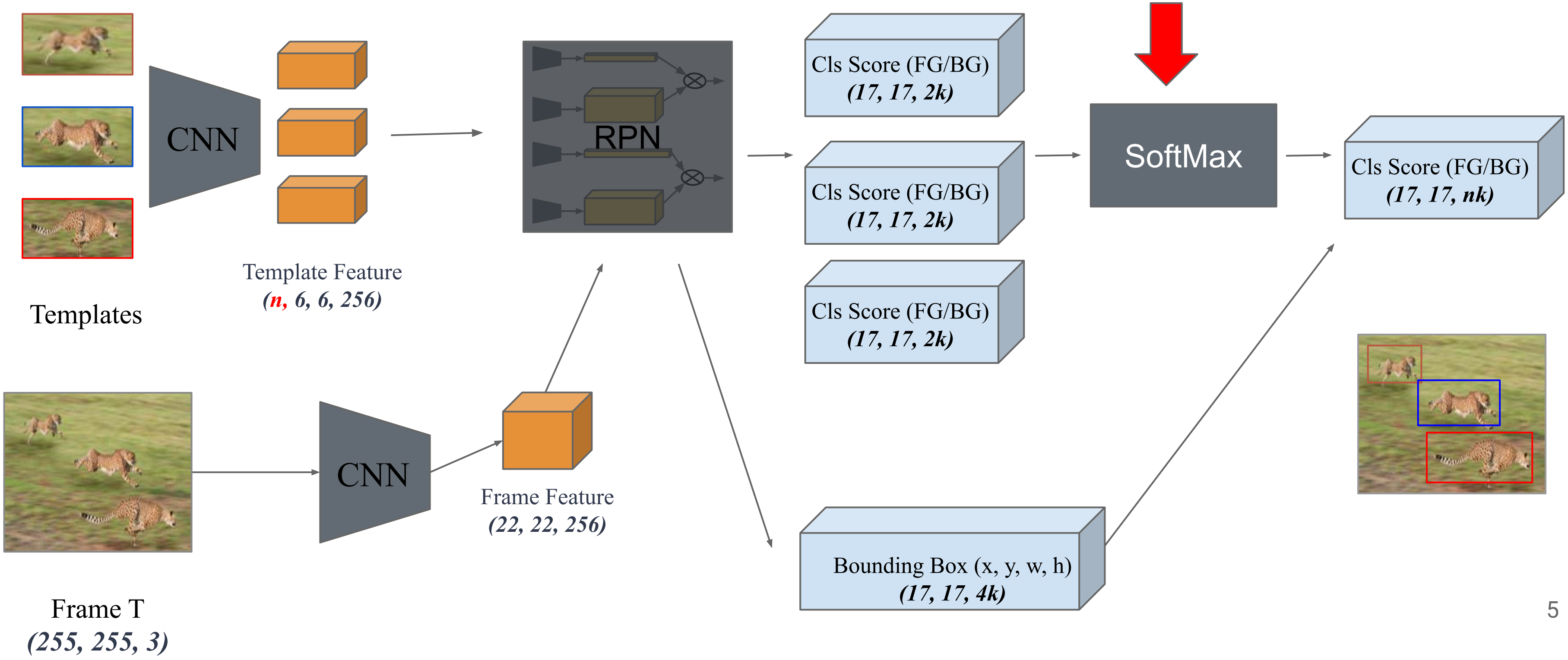
Outline

- From last semester
  - SiameseRPN for Multiple Object Tracking
- One Stage Network: Simultaneously detect and track
  - Richard: Simultaneous Tracking and Detection With Graph Neural Networks (GNN)
  - Chunhui: Track without Bells and Whistles

Extending Siamese RPN for Multiple Object Tracking

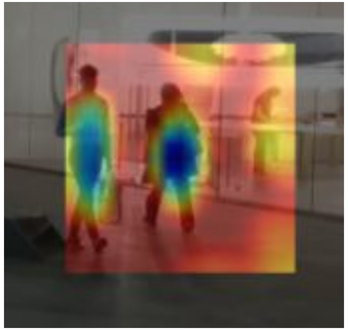
Li, Bo et al. “High Performance Visual Tracking with Siamese Region Proposal Network.” 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition 2018

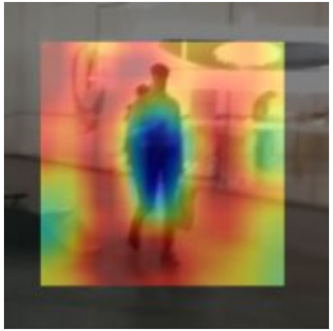
Pipeline: Connect all templates

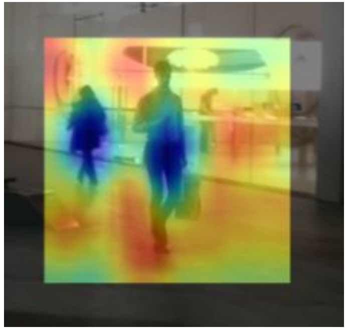


Visualization Response

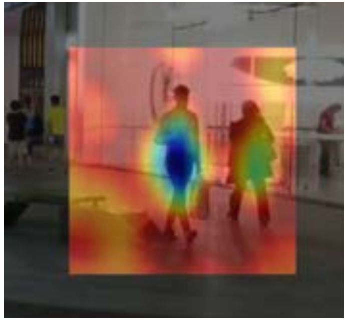
Template:

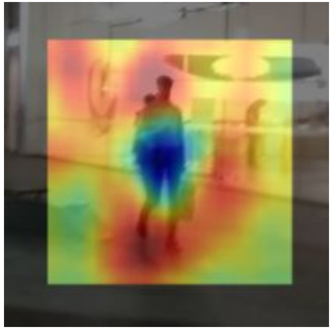


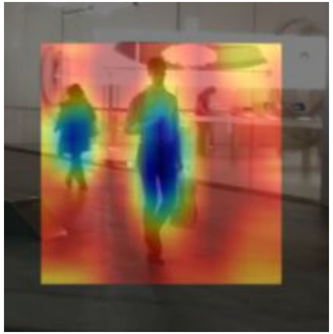




Template:

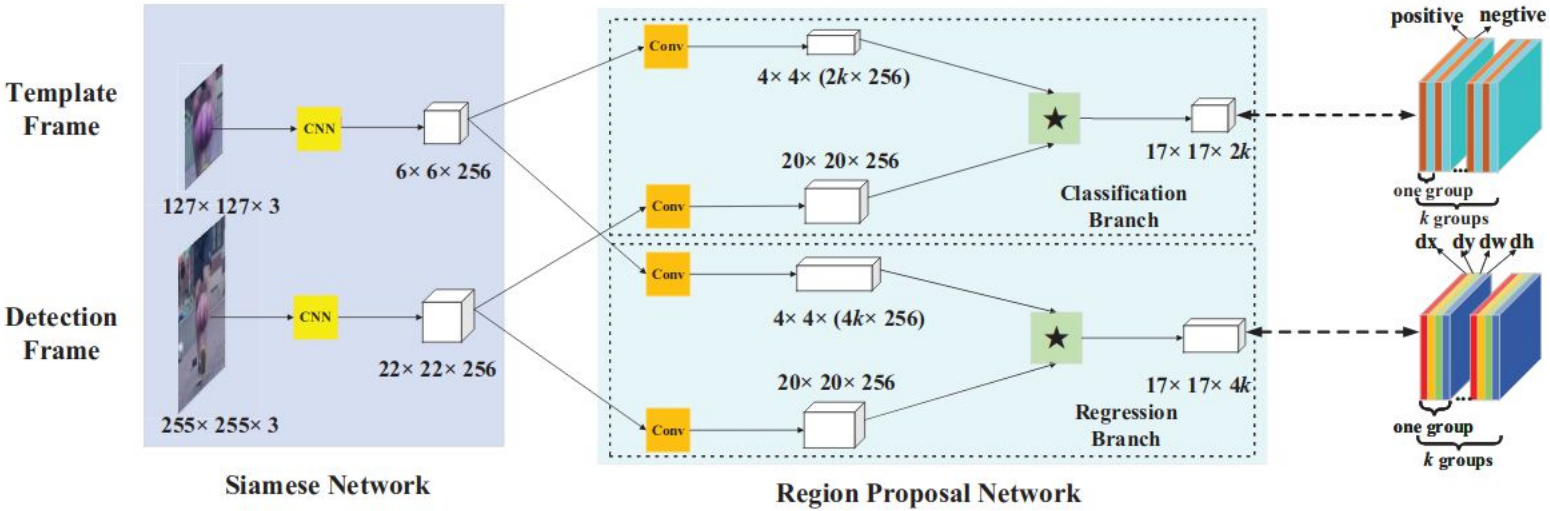






Steering away from SiameseRPN

- SiameseRPN as a single object tracker
  - Only tracks existing objects
  - Training/testing are not scalable
  - Difficult to tune
- Goal: Single stage network for detection and tracking



Outline

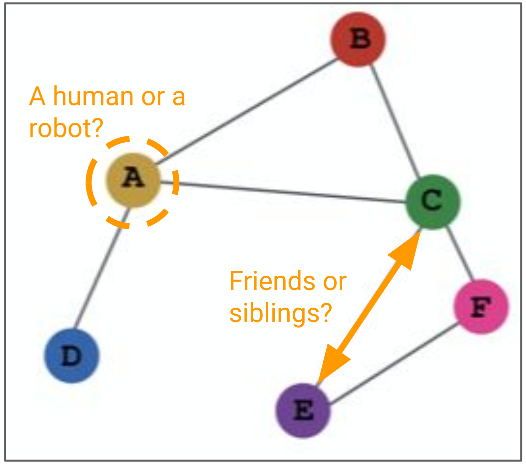
- ~~From last semester~~
    - ~~SiameseRPN for Multiple Object Tracking~~
- One Stage Network: Simultaneously detect and track
    - Richard: Simultaneous Tracking and Detection With Graph Neural Networks (GNN)
    - Chunhui: “Tracking without Bells and Whistles” in ICCV19

Simultaneous Tracking and Detection With Graph Neural Networks (GNN)

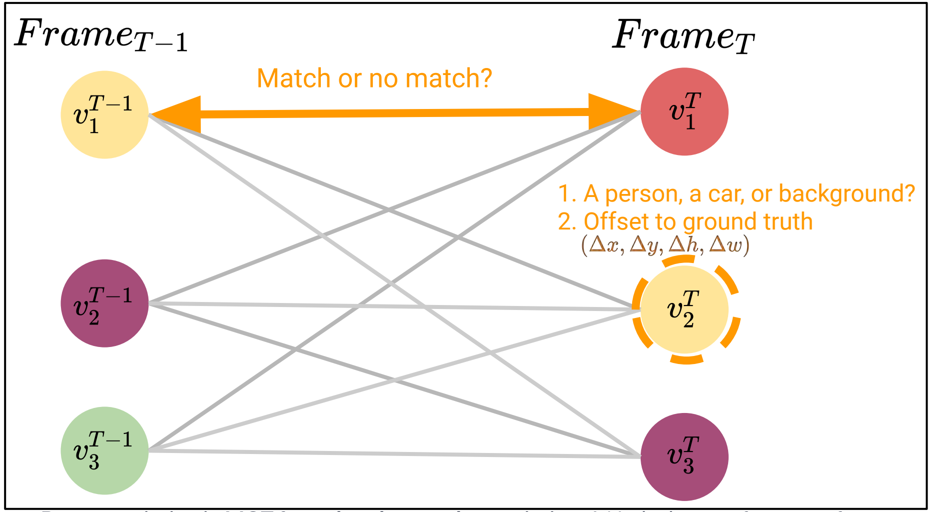
[1] Michael Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, Max Welling, "Modeling Relational Data with Graph Convolutional Networks", ESWC 2018 (Best Student Research Paper)

Brief Introduction to GNN

- Extract and **aggregate** node embeddings and edge embeddings based on local neighborhood
  - Embeddings can be used for downstream tasks, i.e. classification [1] and regression
  - Details about how GNN works can be found in Appendix A

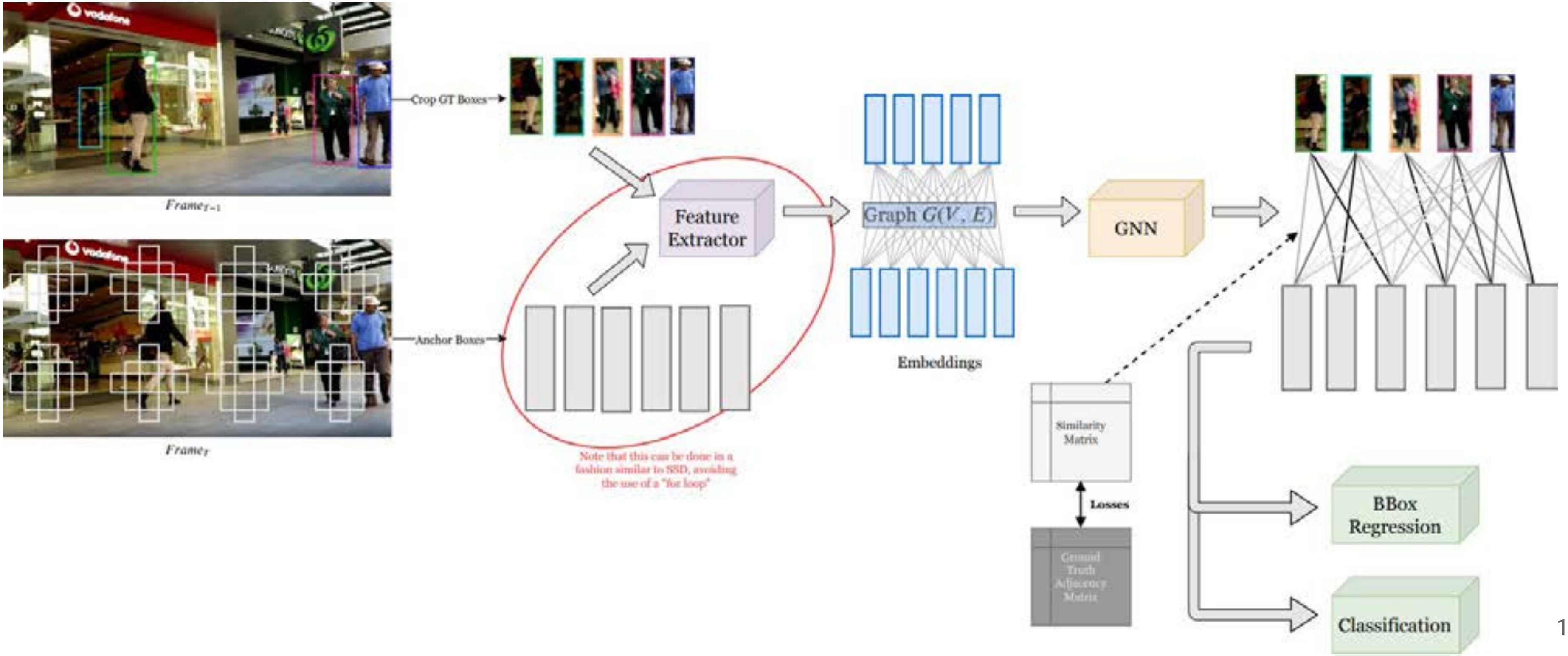


An online social network represented by a graph. Each node denotes a feature of an entity within the social network. Each edge denotes the relationship features between entities.



Data association in MOT formulated as graph association. A Node denotes features of a detection (if in $Frame_{T-1}$) or of an anchor box (if in $Frame_T$). An edge denotes relationship features between detections and anchor boxes.

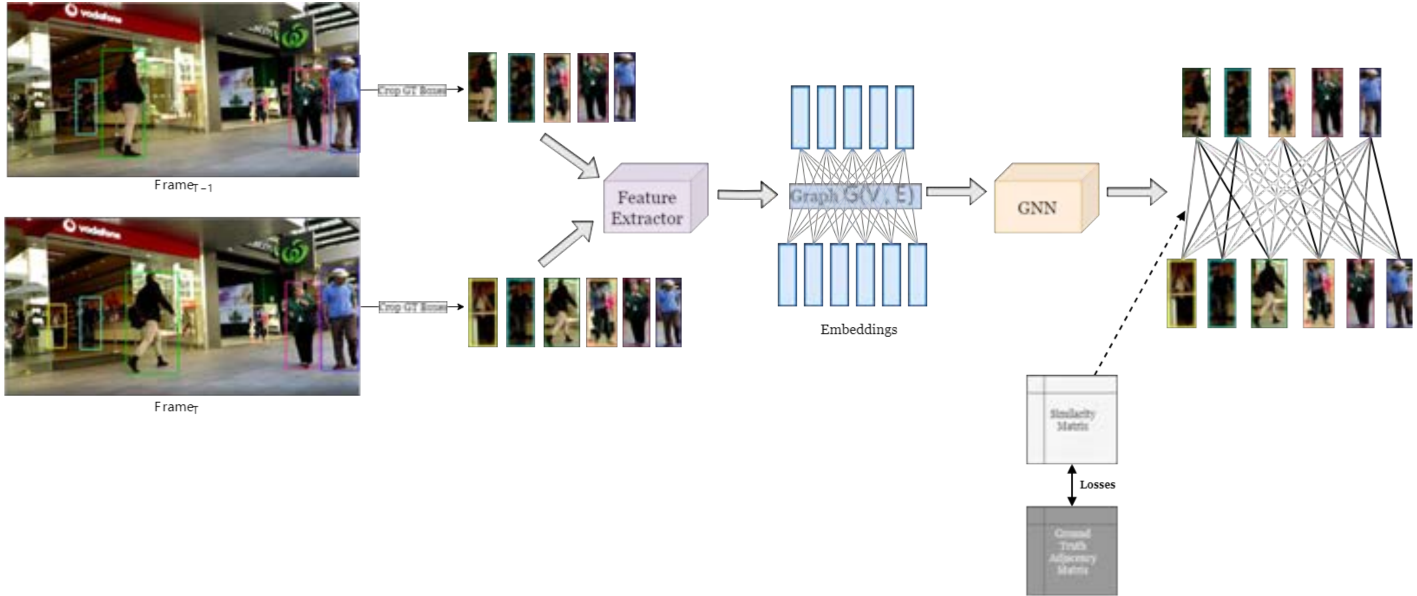
Our Idea


Crop GT Boxes
$Frame_{T-1}$
Anchor Boxes
$Frame_T$
Feature Extractor
Note that this can be done in a fashion similar to SSD, avoiding the use of a "for loop"
Graph $G(V, E)$
Embeddings
GNN
Similarity Matrix
Losses
Ground Truth Adjacency Matrix
BBox Regression
Classification

[2] Xiaolong Jiang and Peizhao Li and Yanjing Li and Xiantong Zhen, "Graph Neural Based End-to-end Data Association Framework for Online Multiple-Object Tracking", arXiv 1907.05315

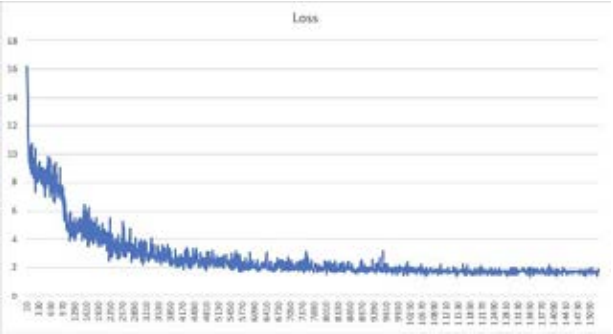
Proof of Concept

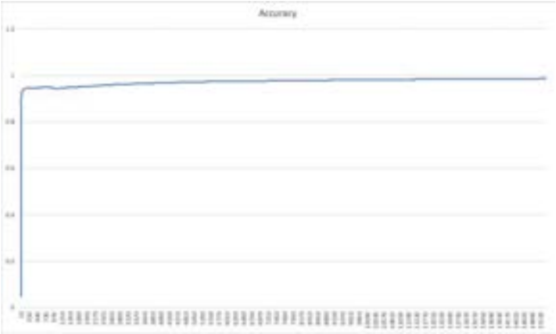
- A GNN matching network [2]
  - So far the most similar work to ours
  - Given detections between two frames, use GNN to match them



Training

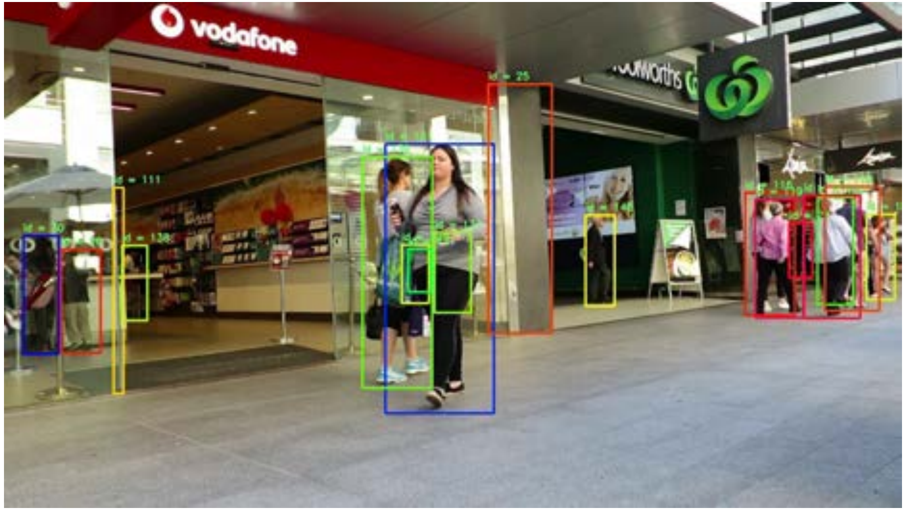
- Dataset: MOT17
- Training sequences ID’s: MOT17-09





Training statistics for sequence MOT17-09

Visual Result



Matching results for sequence MOT-09 with GT BBox

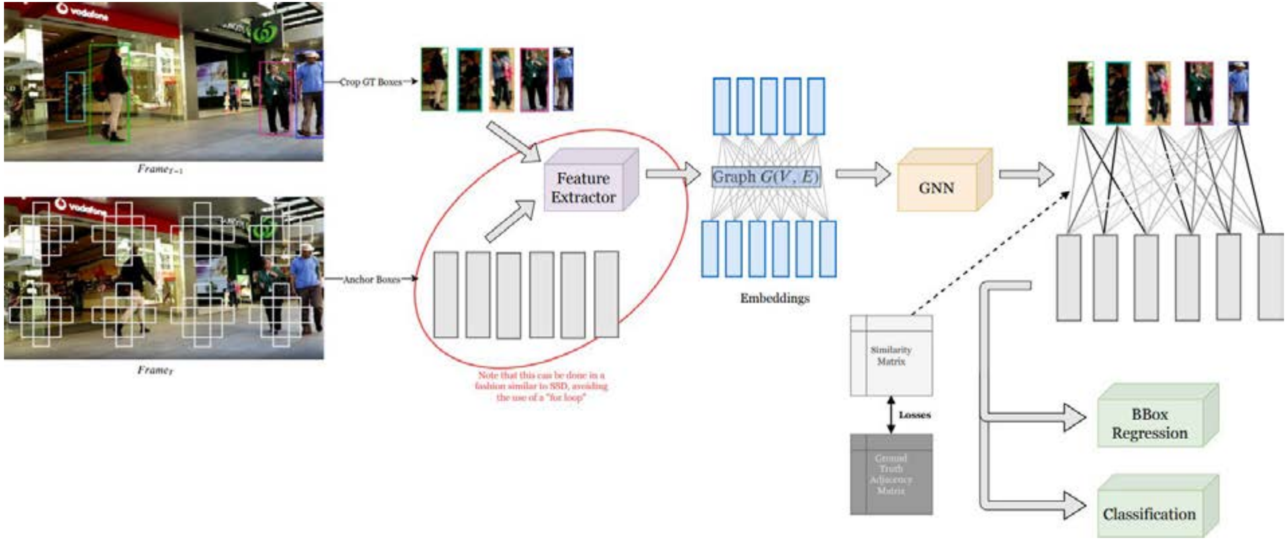
Matching results for sequence MOT-11 with GT BBox

Visual Result

Matching results for sequence MOT-11 with Faster RCNN detections

Next up

- Visual results confirmed that GNNs can be used for matching
- Quantitative results
    - Train on full MOT17 training set and evaluate on MOT17 test set
- Initialize implementation/training/experiments with our idea

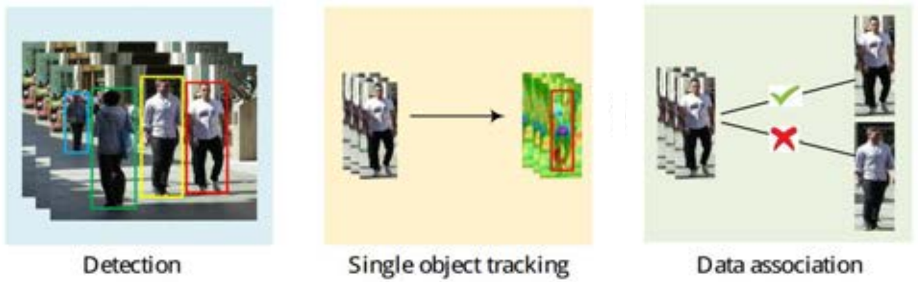


Tracking as Pure Detection

Philipp Bergmann et al. “Tracking without bells and whistles.” ICCV 2019

One Stage Tracking Network

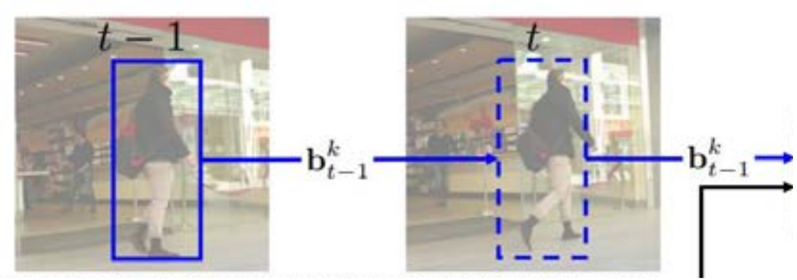
- Motivation:
  - Previous ideas: tracking by detection + association



- Solve tracking as a pure detection problem
  - Detection networks already how to correct anchors to the right bounding box.

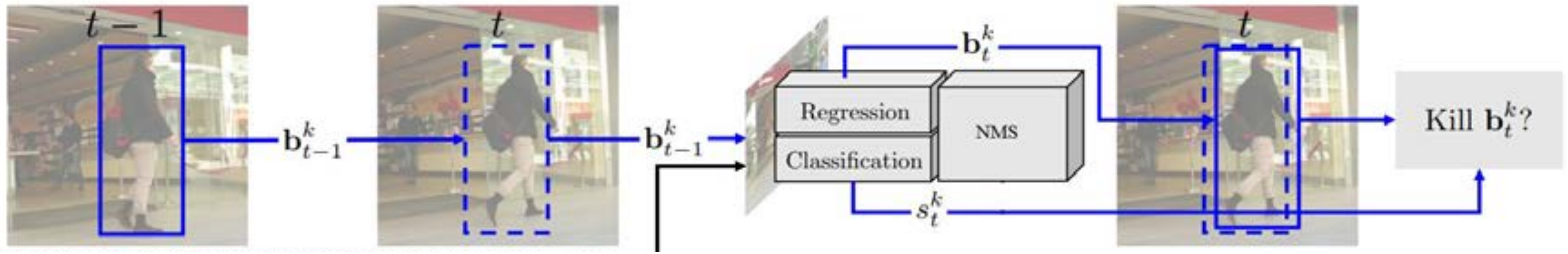
One Stage Tracking Network

- Solve tracking as a pure detection problem



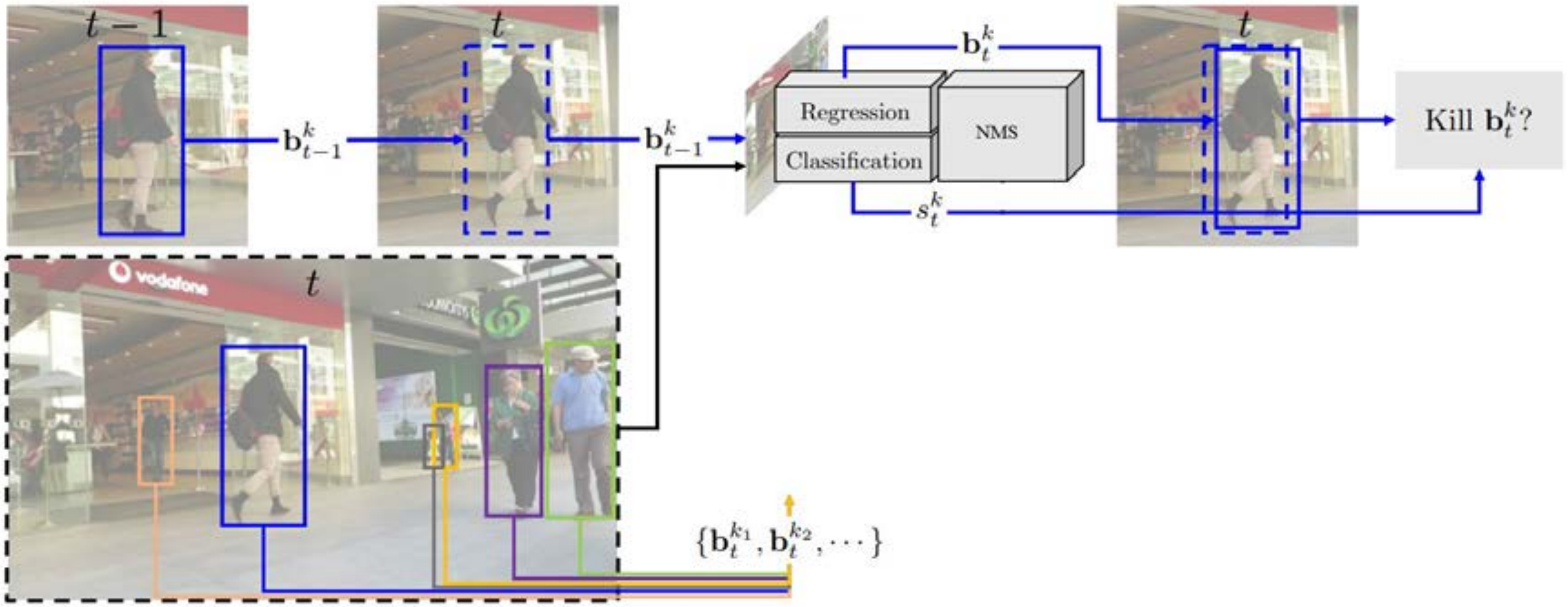
One Stage Tracking Network

- Solve tracking as a pure detection problem



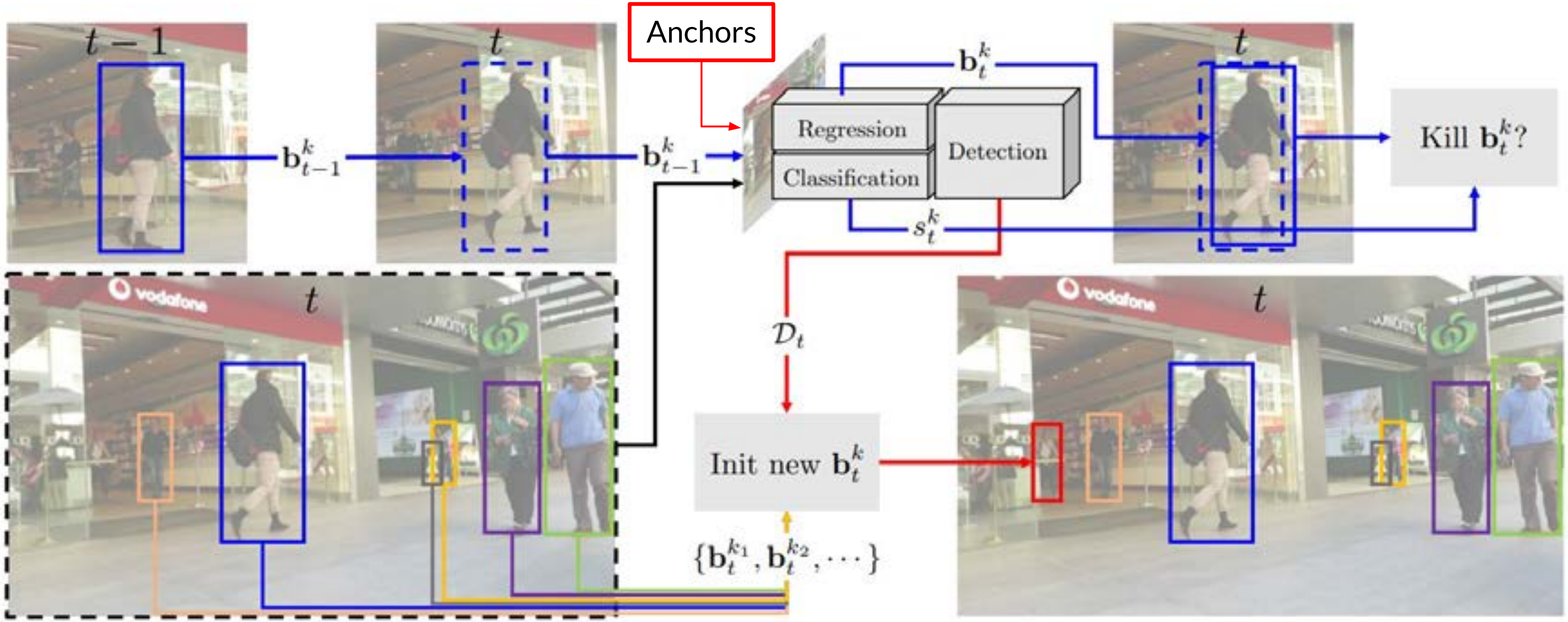
One Stage Tracking Network

- Solve tracking as a pure detection problem



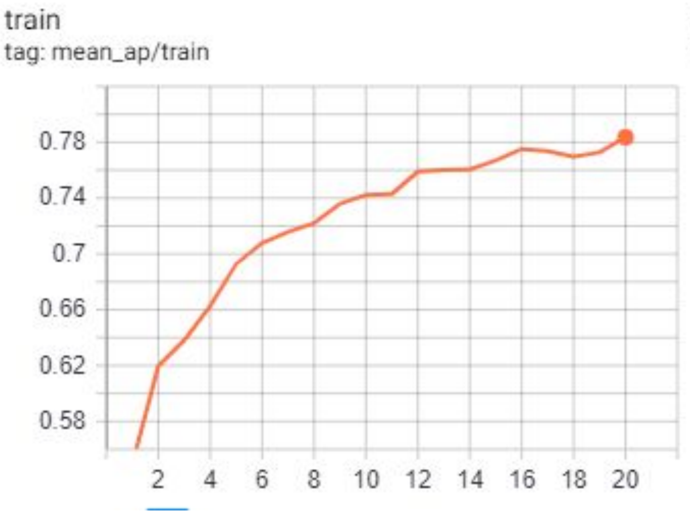
One Stage Tracking Network

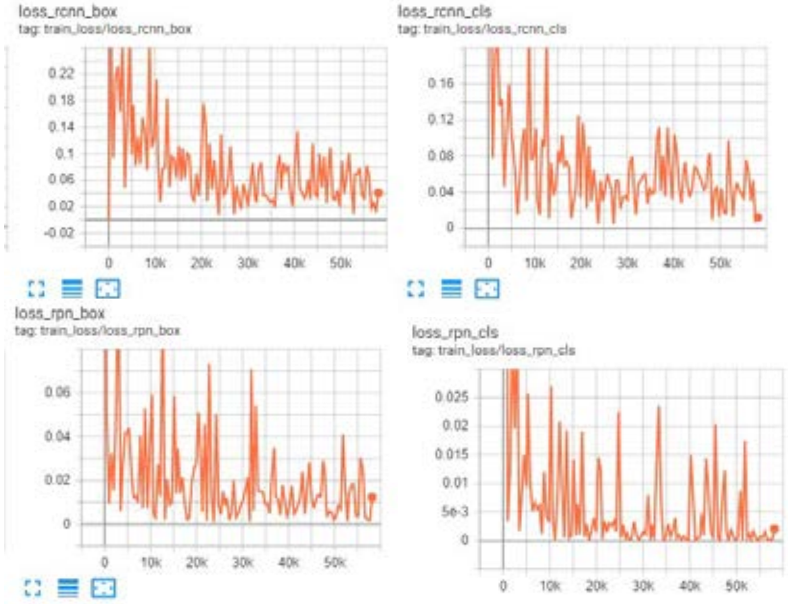
- Solve tracking as a pure detection problem



One Stage Tracking Network: Train and Test

- Train: As same as training a Faster RCNN on MOT17
  - Train on 9 sequences
  - Train time: 45min per epoch, 30 epoch to converge (~24 hours)





One Stage Tracking Network: Train and Test

CVPR 2019 Tracking Challenge Results

Download all results for this benchmark

Click on a measure to sort the table accordingly. See below for a more detailed description.

Detections: Public ▾ Filter

Showing only entries that use public detections!

Tracker	Avg Rank	↑MOTA		IDF1	MT	ML	FP	FN	ID Sw.	Frag	Hz	Detector
SRK_ODESA 1.	7.7	**54.8**	±19.3	**52.2**	**35.4%**	**19.2%**	33,814	**215,572**	3,750 (61.0)	5,493 (89.3)	1.2	Public
D. Borysenko, D. Mykheievskyi, V. Porokhonskyy. ODESA: Object Descriptor that is Smooth Appearance-wise for object tracking tasks. In (to be submitted to ECCV'20), .												
TracktorCV 2.	7.8	51.3	±18.7	47.6	24.9%	26.0%	16,263	253,680	2,584 (47.2)	4,824 (88.2)	2.7	Public
P. Bergmann, T. Meinhardt, L. Leal-Taixé. Tracking without bells and whistles. In ICCV, 2019.												
DD_TAMA19 3.	6.8	47.6	±20.3	48.7	27.2%	23.6%	38,194	252,934	2,437 (44.4)	3,887 (70.9)	0.2	Public
Y. Yoon, D. Kim, K. Yoon, Y. Song, M. Jeon. Online Multiple Pedestrian Tracking using Deep Temporal Appearance Matching Association. In arXiv:1907.00831, 2019.												
V_IOU 4.	8.7	46.7	±19.6	46.0	22.9%	24.4%	33,776	261,964	2,589 (48.6)	4,354 (81.8)	18.2	Public
E. Bochinski, T. Senst, T. Sikora. Extending IOU Based Multi-Object Tracking by Visual Information. In IEEE International Conference on Advanced Video and Signals-based Surveillance, 2018.												
Aaron 5.	6.2	46.5	±18.6	46.6	22.5%	24.6%	40,676	256,671	**2,315** (42.7)	2,968 (54.8)	14.9	Public
Anonymous submission												

One Stage Tracking Network: Train and Test

- Speed Test: 0.4 FPS (~2.5 second per frame)
  - Detection: ~ 0.4 s
  - Tracking: ~ 2s
    - Camera Motion Alignment: 1s
    - Motion Post Processing: 1e-6 s
    - Regression and track: 0.05s
    - NMS: ~0.8s
  - New object and ReID: ~ 0.1s

One Stage Tracking Network: Train and Test

- Speed Test: 0.4 FPS (~2.5 second per frame)
  - <u>Detection: ~ 0.4 s</u>
  - Tracking: ~ 2s
    - Camera Motion Alignment: 1s
    - Motion Post Processing: 1e-6 s
    - <u>Regression and track: 0.05s</u>
    - <u>NMS: ~0.8s</u>
  - New object and ReID: ~ 0.1s

Speed up using Yolo V3

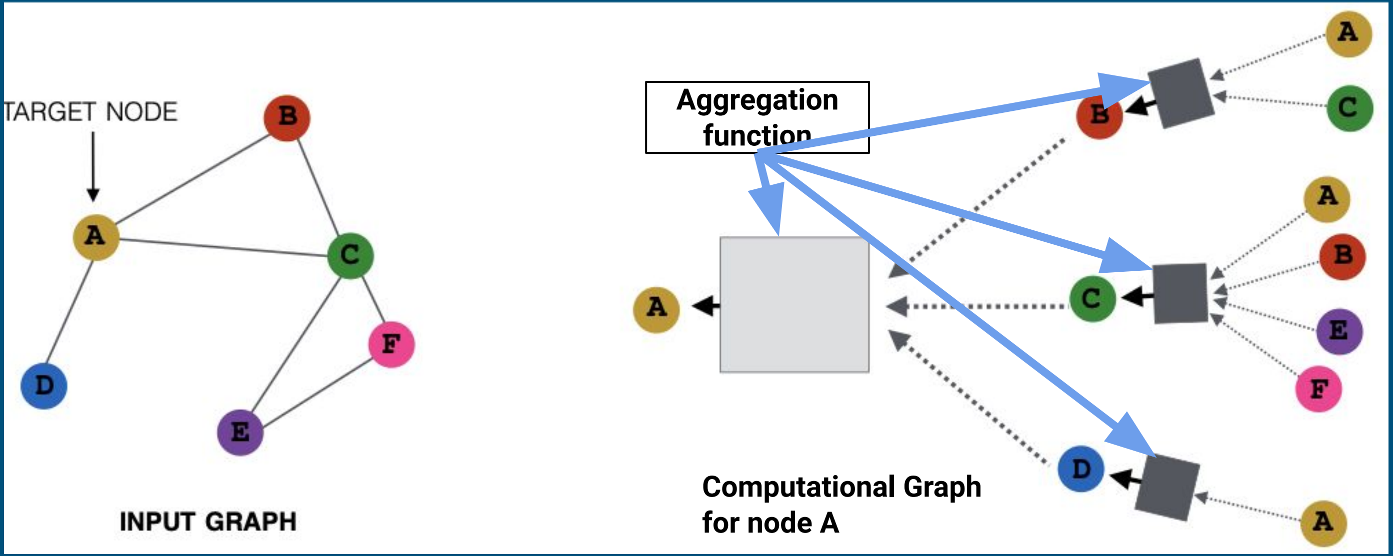
Analysis and Improve

Timeline

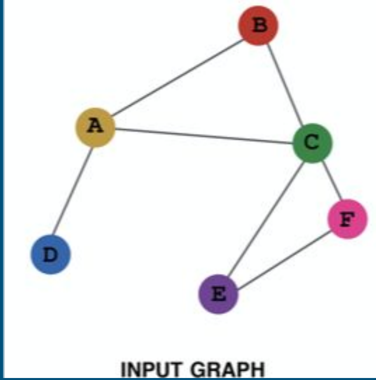
- ~~9/5 - 9/19:~~
    - ~~Collect Idea, Running baseline code (re-train and evaluate).~~
- 9/19 - 10/3:
    - Richard: GNN, results
    - Chunhui: Retrain YoloV3 on MOT17 Dataset, results
- 10/3 - 10/17:
    - Richard: Merge GNN with SSD for one-stage network
    - Chunhui: Embedding bank for ID switch
- 10/17 - 10/31:
    - Richard: One-stage network: train and test, ablation study
    - Chunhui: Speed test and deployment
- 10/31 - 11/14:
    - Wrap up

Appendix A: more details about GNN

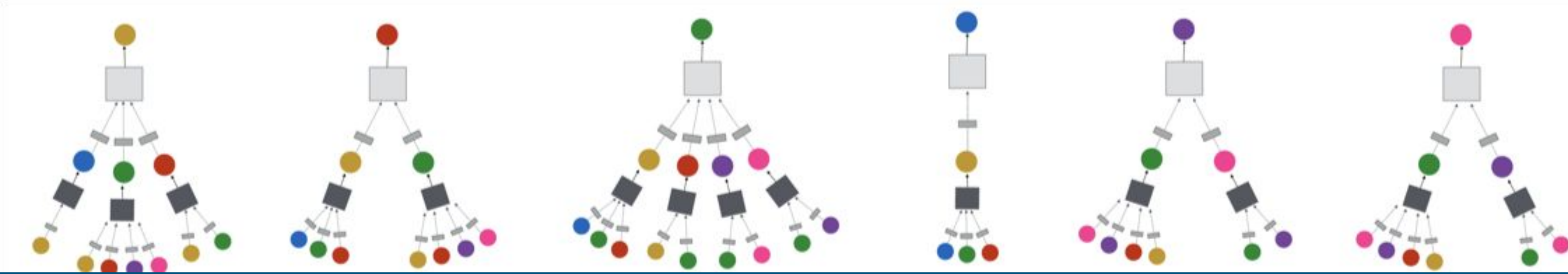
- Key Idea in GNN: Neighborhood Aggregation
  - Generate node embeddings based on local neighborhoods
  - Nodes aggregate information from their neighbors using neural networks



Appendix A: more details about GNN

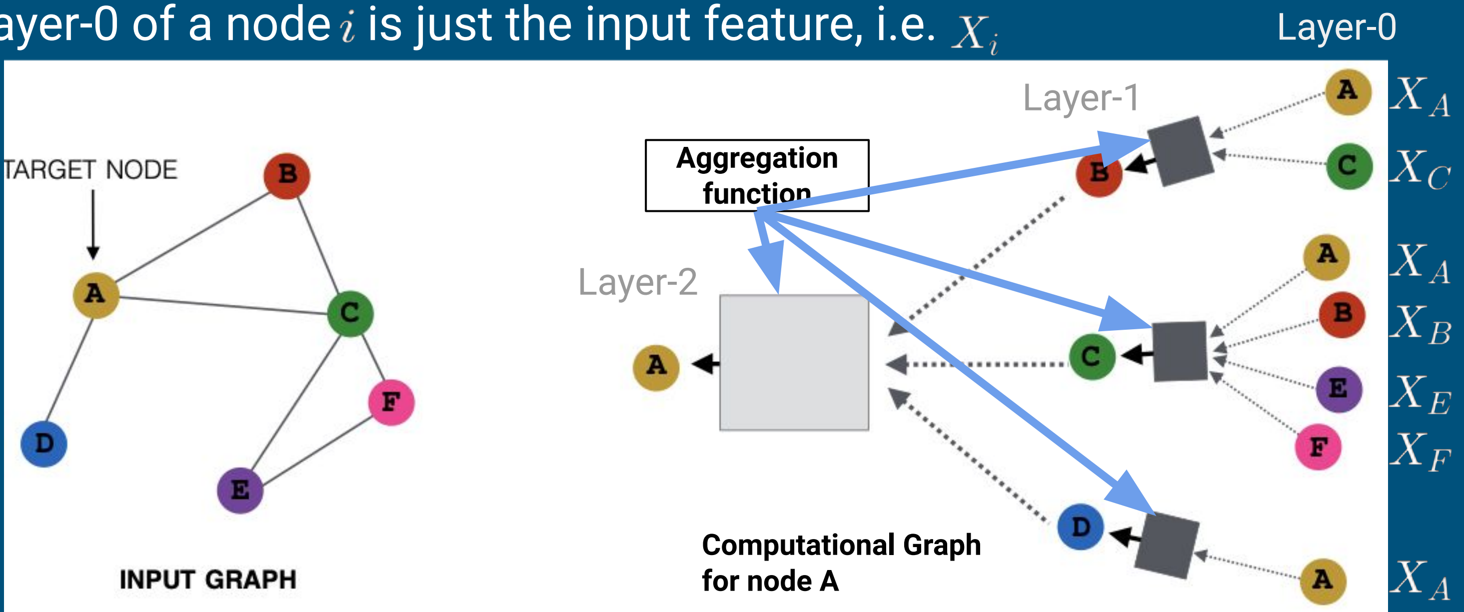


- Neighborhood Aggregation defines a computational graph
- Each node will have a computational graph for its aggregation



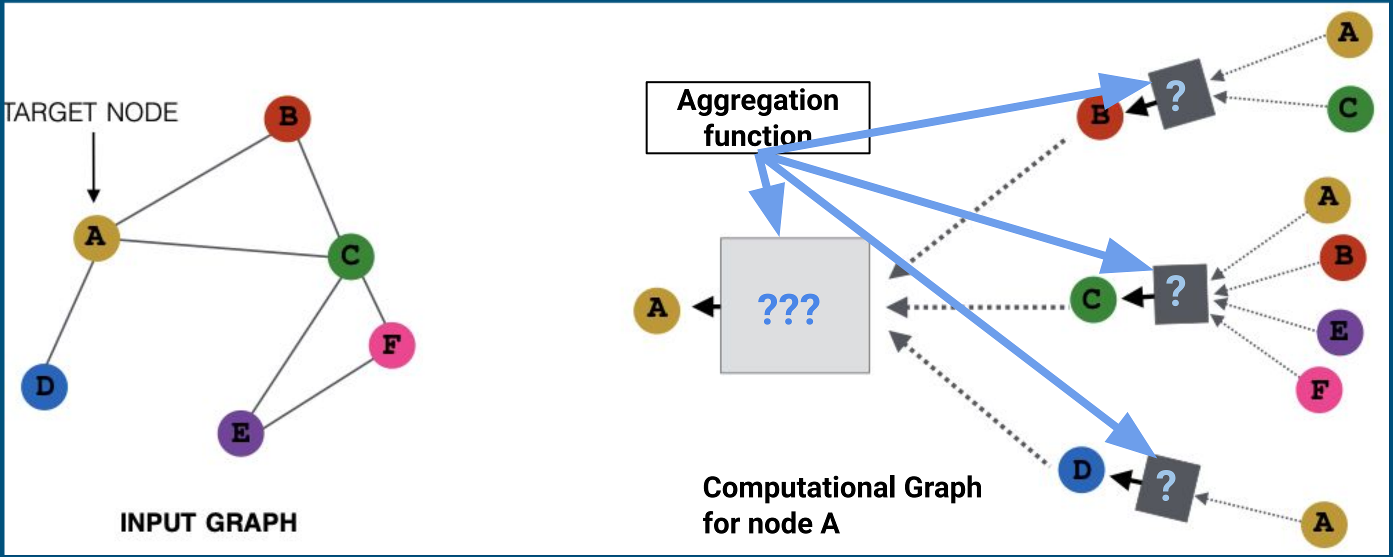
Appendix A: more details about GNN

- Nodes have embeddings at each GNN layer - one layer means one aggregation
- GNN can have an arbitrary number of layers (aggregations)
- Layer-0 of a node i is just the input feature, i.e. X_i



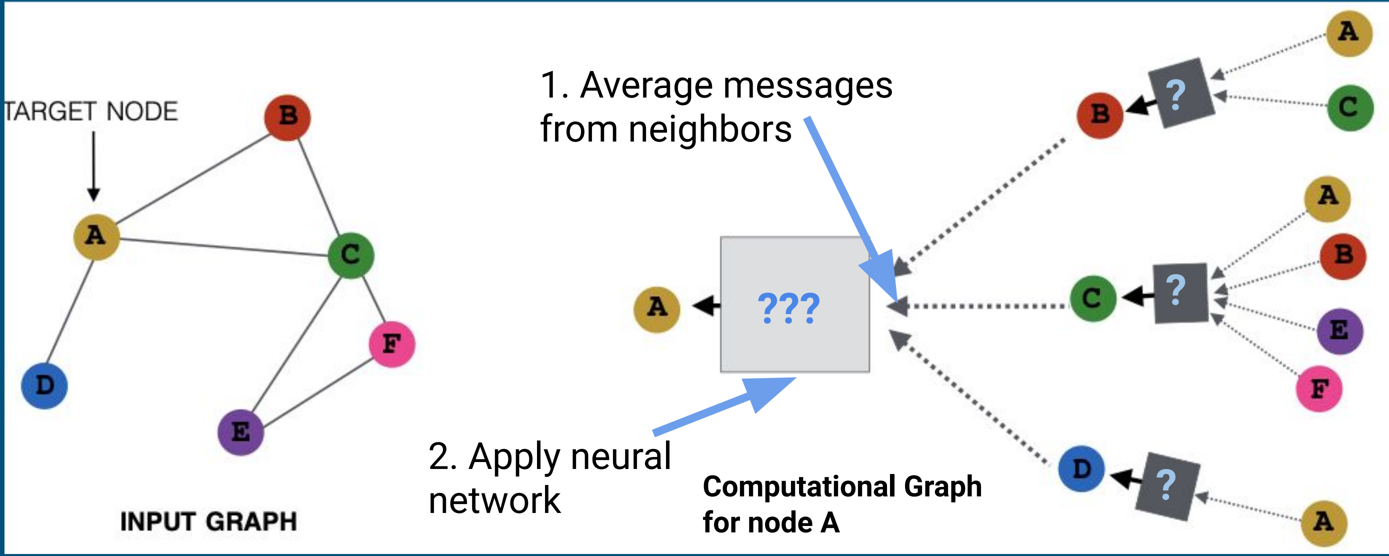
Appendix A: more details about GNN

- But what are aggregation functions?



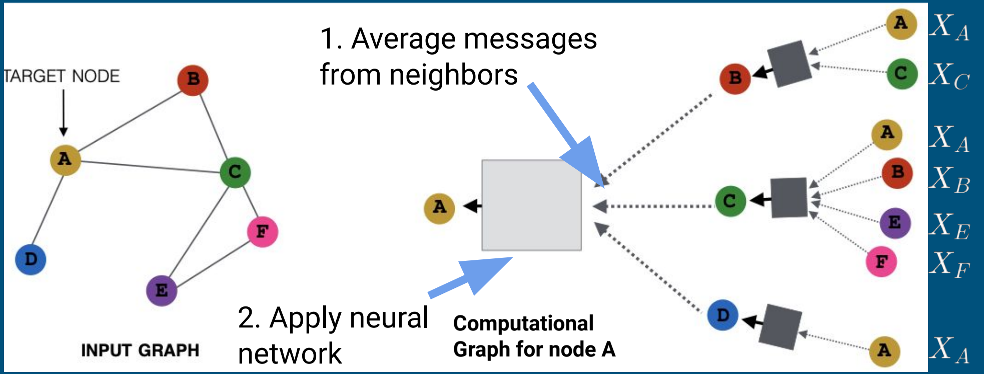
Appendix A: more details about GNN

- A basic approach: average the messages from neighbors and apply neural networks



Appendix A: more details about GNN

- Basic Approach: average messages from neighbors and apply neural networks



Layer-0 embedding of node ***v***

$$\boxed{h_v^0 = X_v}$$

Previous layer embedding of ***u***

Average neighbor's previous layer embeddings

$$h_v^k = \sigma\left(\mathbf{W}_k \sum_{u\in N(v))} \frac{h_u^{k-1}}{|N(v)|} + \mathbf{B}_k h_v^{k-1}\right)$$

Layer-***k*** embedding of node ***v***

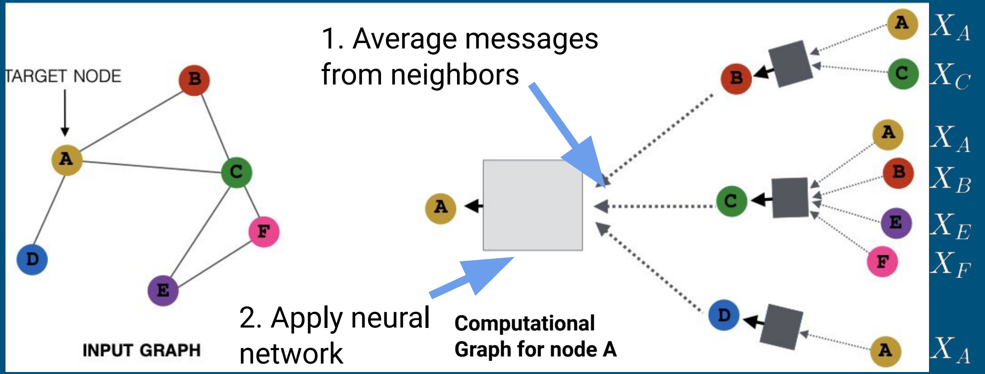
Activation Func (ReLU)

Cardinality of Neighbors around ***v***

Previous layer embedding of ***v***

Appendix A: more details about GNN

- Training the GNN



Layer-0 embedding of node ***v***

$$\boxed{h_v^0 = X_v}$$

Trainable Parameters

$$h_v^k = \sigma\left(\boxed{\mathbf{W}_k} \sum_{u \in N(v))} \frac{h_u^{k-1}}{|N(v)|} + \boxed{\mathbf{B}_k} h_v^{k-1}\right)$$

$$\boxed{z_v^K = h_v^K}$$

Layer-***K*** embedding of node ***v*** after ***K*** layers of neighborhood aggregation

- Feed z_v^K to Loss Function and apply Gradient Descent to train $\mathbf{W}_k$ and $\mathbf{B}_k$