University of Tehran

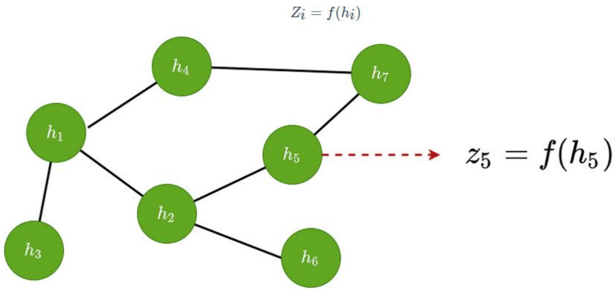
June 2021

Multiple Object Tracking - Graph

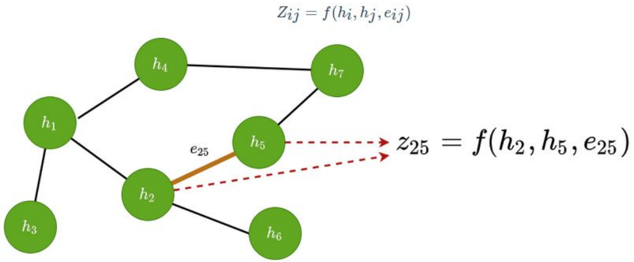
Ali Izadi

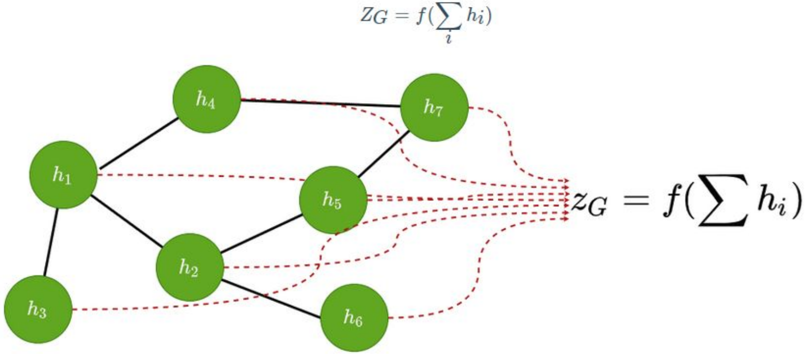
Types of problems have graph structured data

Node classification



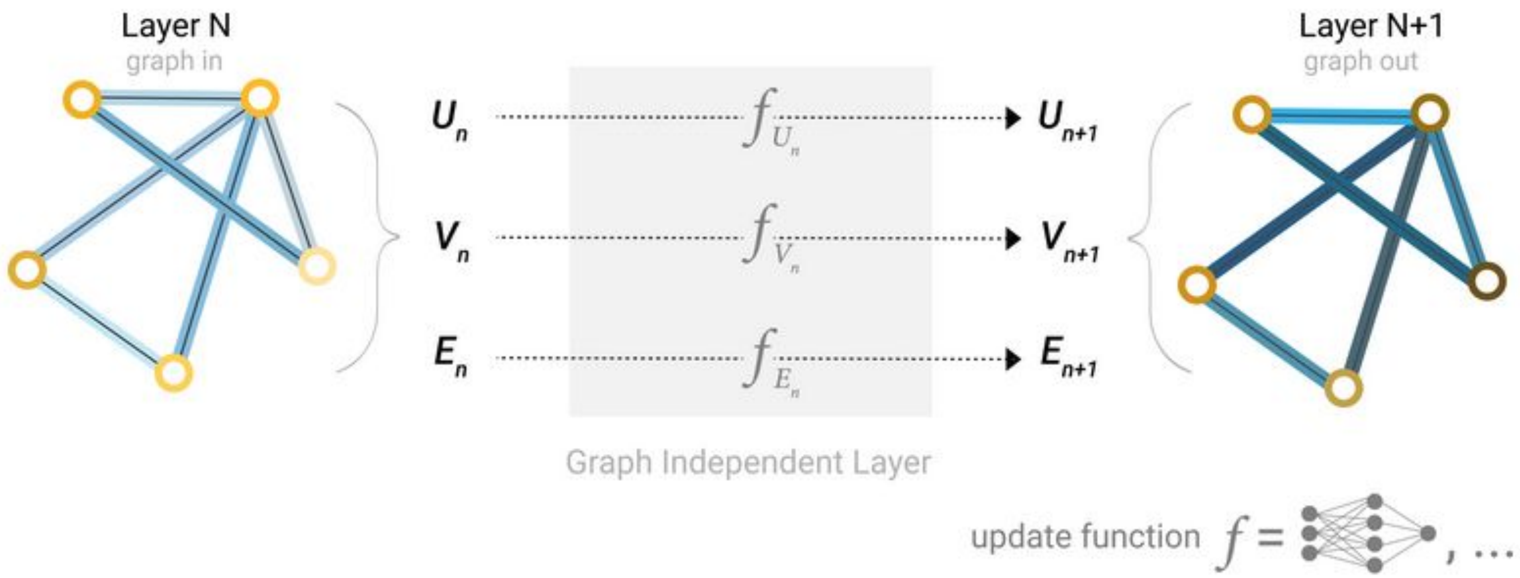
Edge classification





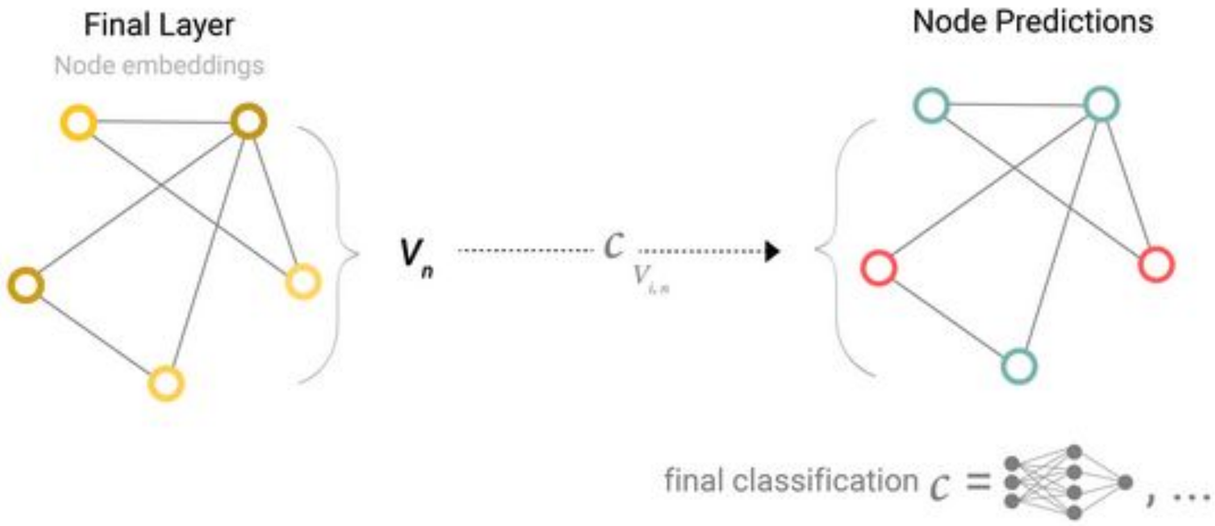
Graph classification

The simpleset Graph Neural Network (GNN)

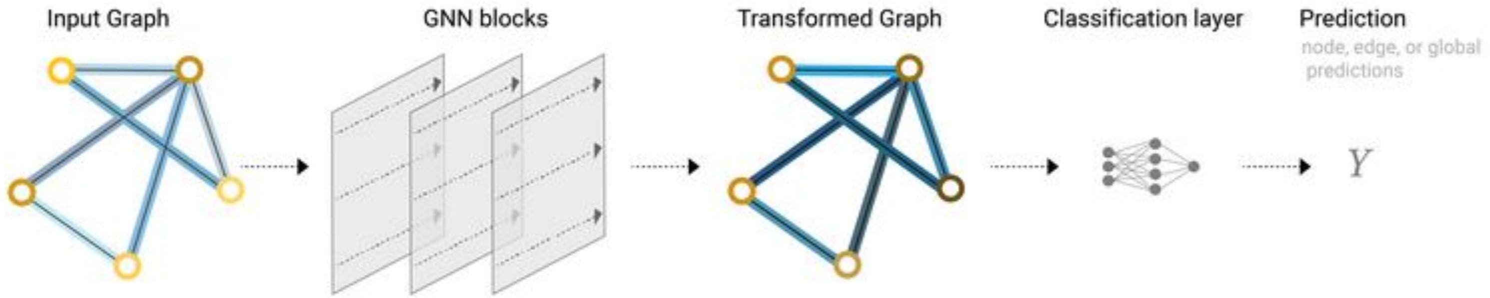


A single layer of a simple GNN. A graph is the input, and each component (V,E,U) gets updated by a MLP to produce a new graph. Each function subscript indicates a separate function for a different graph attribute at the n-th layer of a GNN model.

Node Classification

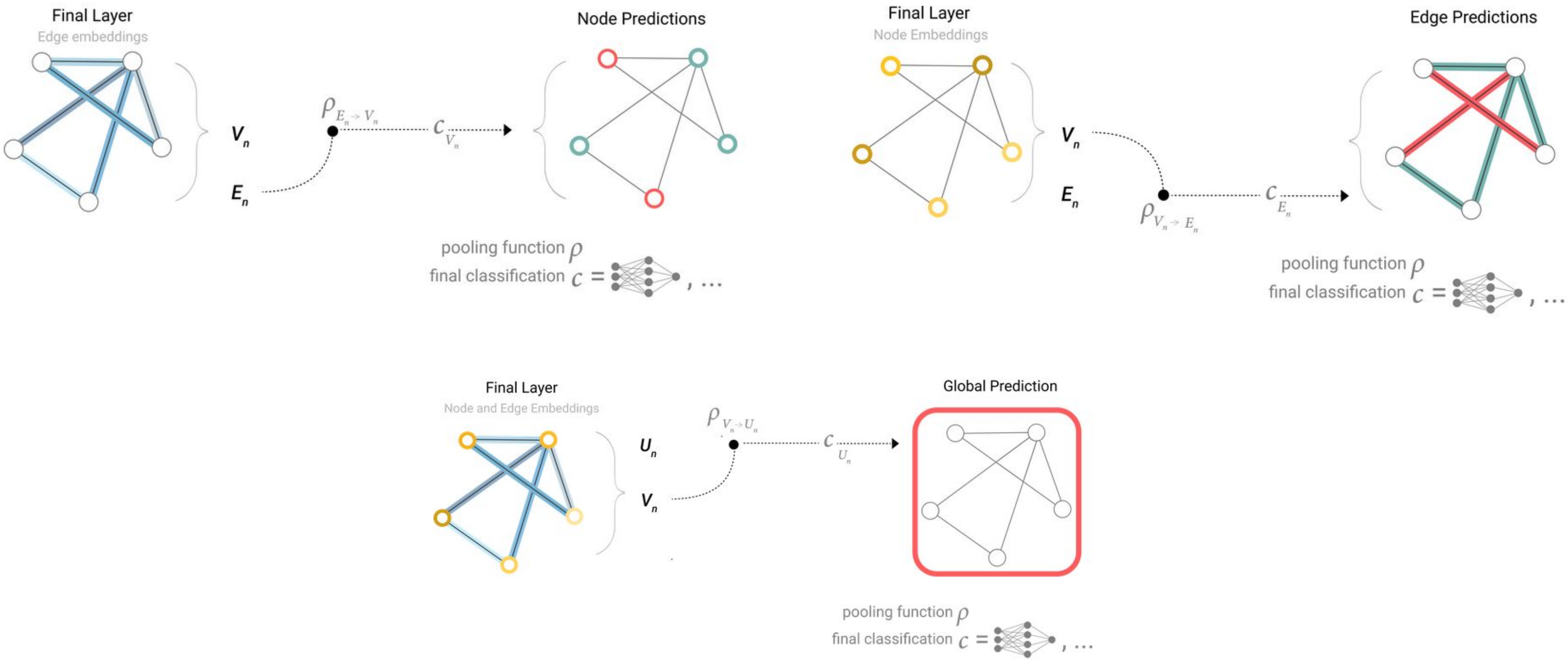


End to End Architecture



An end-to-end prediction task with a GNN model.

Polling Operations

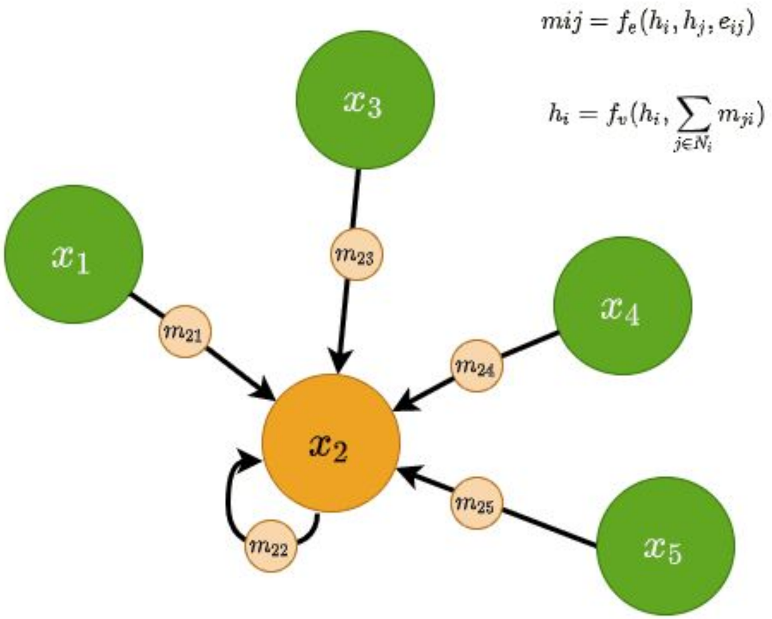


Convolutions on the graph based on the graph topology

Message Passing:

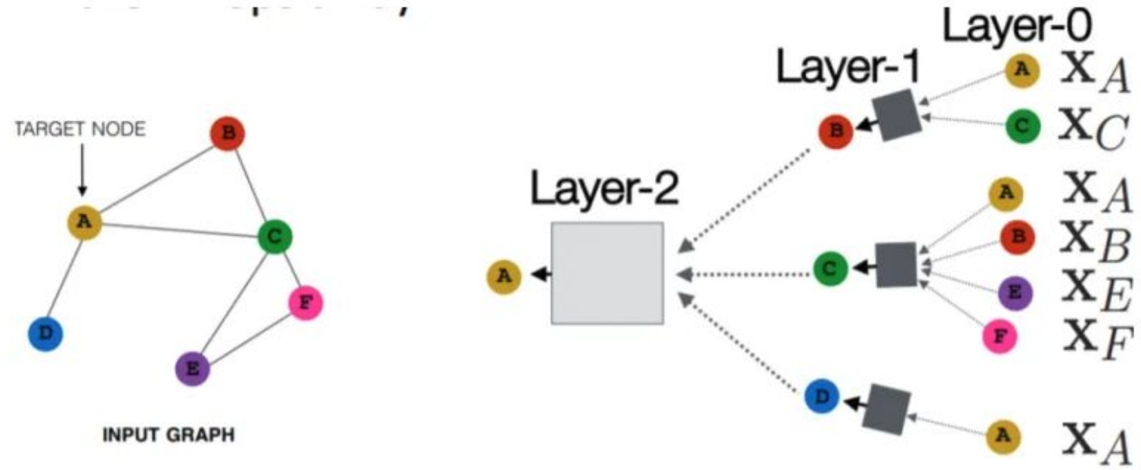
$$h_i = f_v(h_i, \sum_{j \in N_i} m_{ji})$$

1. The node’s feature vectors are **transformed** using some sort of projection.
2. They are **aggregated** by a permutation-invariant function.
3. The feature vector of each node is **updated** based on its current values and the aggregated neighbourhood representation



Convolutions on the graph based on the graph topology

By iteratively repeating the 1-hop localized convolutions K times (i.e., repeatedly ‘passing messages’), the receptive field of the convolution effectively includes all nodes upto K hops away.



Deep Model: Many Layers | Source

Embedding Computation

most popular ones:

- Graph Convolutional Networks (GCN)
  Kipf, T.N. and Welling, M., 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

- Graph Attention Networks (GAT)
  Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P. and Bengio, Y., 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903*.

Object Tracking + GNN

Weng, X., Wang, Y., Man, Y. and Kitani, K.M., 2020. **Gnn3dmot: Graph neural network for 3d multi-object tracking with 2d-3d multi-feature learning.** In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 6499-6508).

Contributions:

- **Instead** of obtaining features for each object **independently**, we propose a novel **feature interaction** mechanism by introducing the **Graph Neural Network**.
- **Joint** feature extractor to learn appearance and motion features from **2D and 3D space** simultaneously.

Approach

- Given M tracked objects $o_i \in O$ at frame t where $i \in \{1, 2, \cdots, M\}$
- N detected objects $d_j \in D$ in frame t+1 where $j \in \{1, 2, \cdots, N\}$

we want to learn **discriminative** feature from O and D and then find the **correct matching** based on the **pairwise feature similarity**.

entire network consists of:

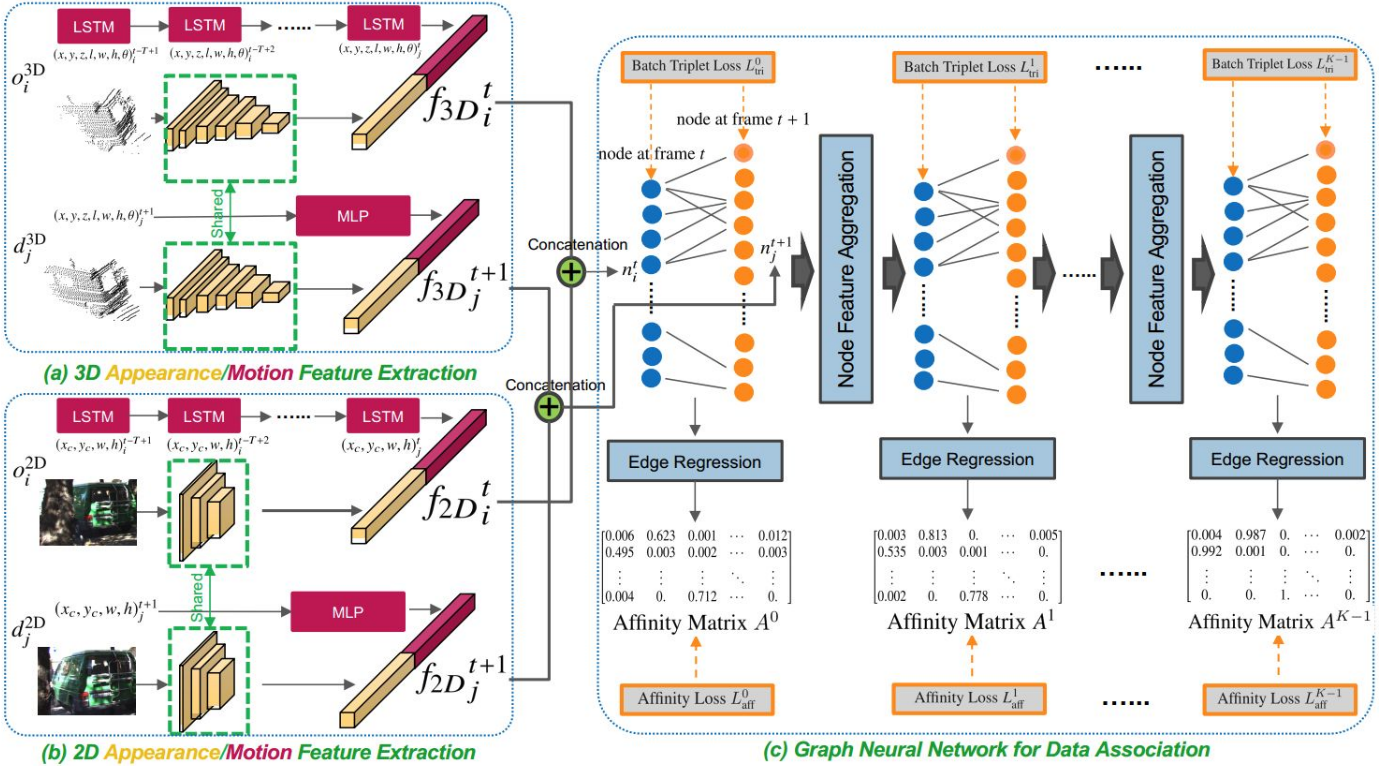
1) a 3D appearance and motion feature extractor
2) a 2D appearance and motion feature extractor

- Both 2D and 3D feature extractors are applied to all objects in O and D
  $o^{3D}=\{x, y, z, l, w, h, \theta, I\}$ and $o^{2D}=\{xc, yc, w, h, I\}$ where I=assigned ID

3) a **graph neural network** that takes the fused object feature as input and constructs a graph with **node being the object feature in frame t and t+1**.
4) **graph neural network iteratively** aggregates the node feature from the **neighborhood** and computes the **affinity matrix** for matching using **edge regression**.

Proposed Network

3D appearance network: Point Cloud
2D appearance network: ResNet

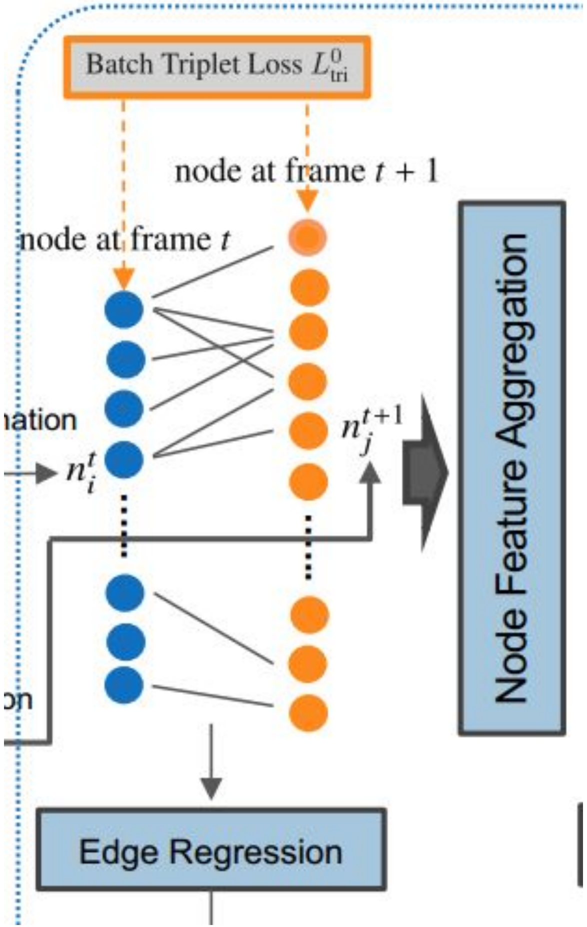


Graph Neural Network (Graph Construction)

- M features for tracked objects in frame t
- N features for detected objects in frame t+1

- We construct the edge only between the pair of nodes in different frames.

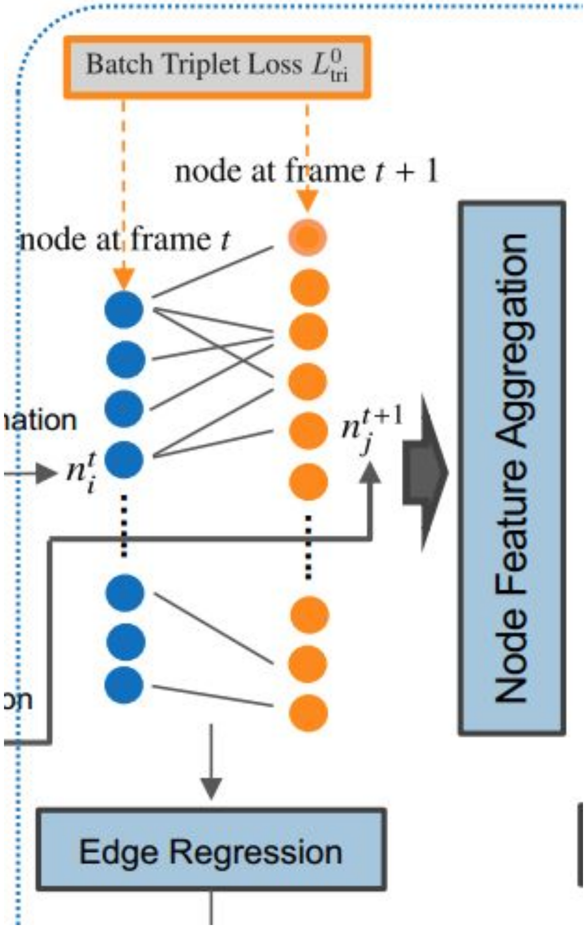
- for any tracked object o_i in frame t, the possible matched detection d_j in frame t+1 is most likely located in the **nearby location.**
  Therefore, we **construct the edge only if** two nodes' detection centers **have distance less than** Dist3D max meters in 3D space and Dist2D max pixels in the image.



Graph Neural Network (Edge Regression)

- M × N affinity matrix A based on the pairwise similarity of the features:

$$A_{ij} = \mathrm{Sigmoid}(\sigma_2(\mathrm{ReLU}(\sigma_1(n_i^t - n_j^{t+1})))),$$



Graph Neural Network (Node Feature Aggregation)

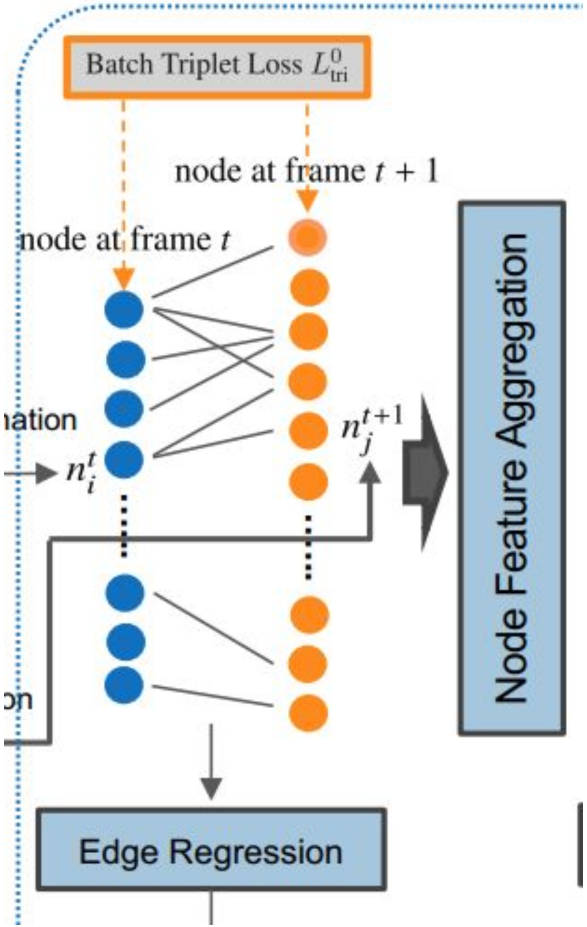
- iteratively update the node feature by aggregating features from the neighborhood.

$$\text{(Type 1)} \quad n_i^{t\prime} = \sum_{j \in \mathcal{N}(i)} \sigma_3(n_j^{t+1}),$$

$$\text{(Type 2)} \quad n_i^{t\prime} = \sigma_4(n_i^t) + \sum_{j \in \mathcal{N}(i)} \sigma_3(n_j^{t+1}),$$

$$\text{(Type 3)} \quad n_i^{t\prime} = \sigma_4(n_i^t) + \sum_{j \in \mathcal{N}(i)} \sigma_3(n_j^{t+1} - n_i^t),$$

$$\text{(Type 4)} \quad n_i^{t\prime} = \sigma_4(n_i^t) + \sum_{j \in \mathcal{N}(i)} \sigma_3(A_{ij}(n_j^{t+1} - n_i^t)),$$



Graph Neural Network (Loss for training)

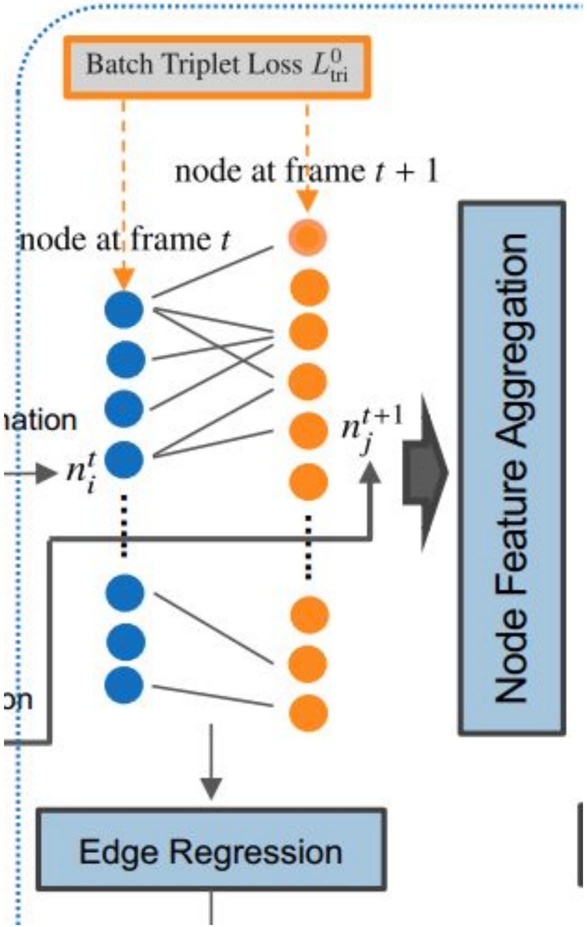
- Our proposed network employs two losses in all K layers during training:
  (1) batch triple loss Ltri;
  (2) affinity loss

$$L = \sum_{k=0}^{K-1}(L^k_{\text{tri}} + L^k_{\text{aff}}).$$

$$L_{\text{tri}} = \max\Big(||n_i^t - n_j^{t+1}|| - \min_{\substack{d_s \in D \\ id_i \neq id_s}} ||n_i^t - n_s^{t+1}|| - \min_{\substack{o_r \in O \\ id_r \neq id_j}} ||n_r^t - n_j^{t+1}|| + \alpha,\ 0\Big),$$

$$L_{\text{bce}} = \frac{-1}{MN}\sum_i^M \sum_i^N A^g_{ij}\log A_{ij} + (1 - A^g_{ij})\log(1 - A_{ij}).$$

$$L_{\text{ce}} = \frac{-1}{M}\sum_i^M A^g_{ij}\log\Big(\frac{\exp A_{ij}}{\sum_i^M \exp A_{ij}}\Big).$$

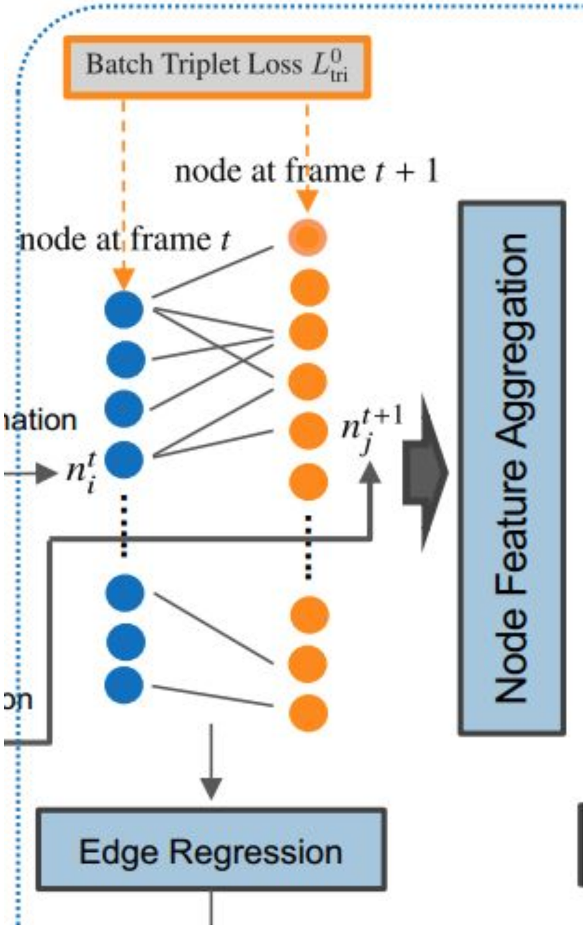


Graph Neural Network (Tracking Management)

false positive detection + false negative (missing detection)

controlling the birth and death of the objects:

- If a new object is able to find the match in **Birmin** frames continuously, we will then assign an ID to this object and add it to the set of tracked objects O.
- If a tracked object cannot find the matched detection in **Agemax** frames, we believe that this object has disappeared and will delete it from the set of tracked objects O.



https://distill.pub/2021/understanding-gnns/
https://neptune.ai/blog/graph-neural-network-and-some-of-gnn-applications
https://distill.pub/2021/gnn-intro/
https://theaisummer.com/gnn-architectures/

Thank you.