# Graph Neural Network

**Haozhi Qi**

Spring 2020

Lecture 27 of CS 182/282A: Designing, Visualizing and Understanding Deep Neural Networks

# Outlines

- Why Graph Neural Network (GNN)?
- The relation between GNN, CNN and transformer.
- Applications of Graph Neural Network
  - On ‘classical’ network problems.
  - Emerging New Research Directions

# Outlines

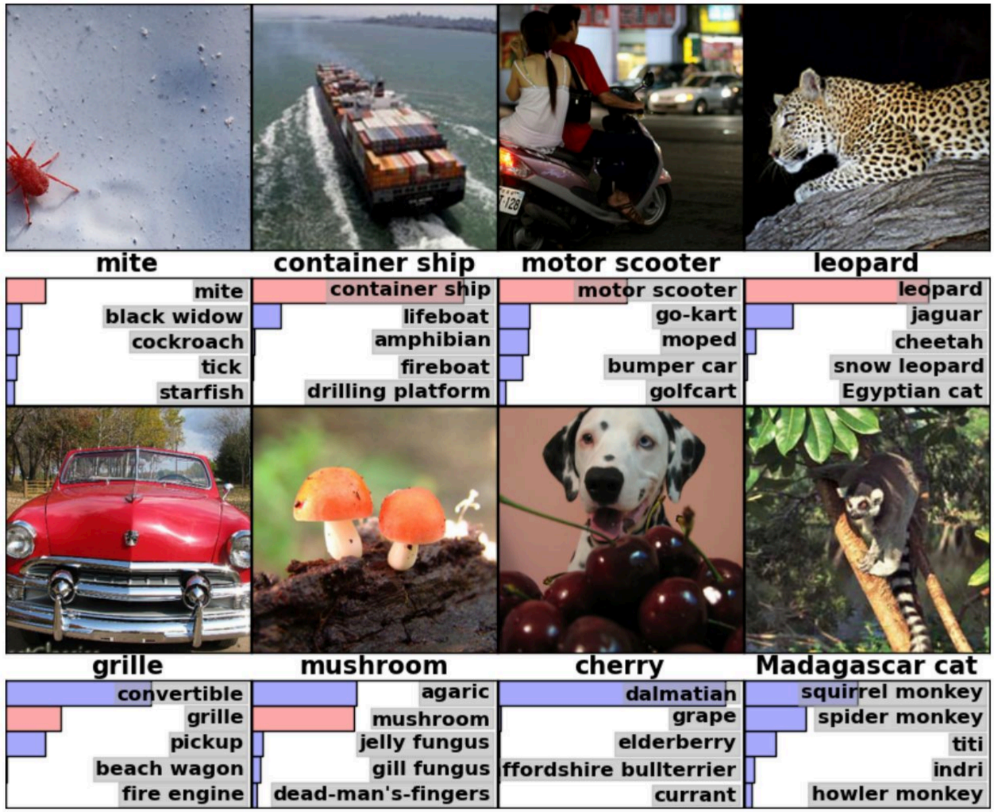
- **Why Graph Neural Network (GNN)?**
- GNN, and the relation between GNN, CNN and transformer.
- Applications of Graph Neural Network
  - On ‘classical’ network problems.
  - Emerging New Research Directions

# Deep Learning on Regular Data

- As you see in this course, deep learning achieves great success in
  - Images

IMAGENET

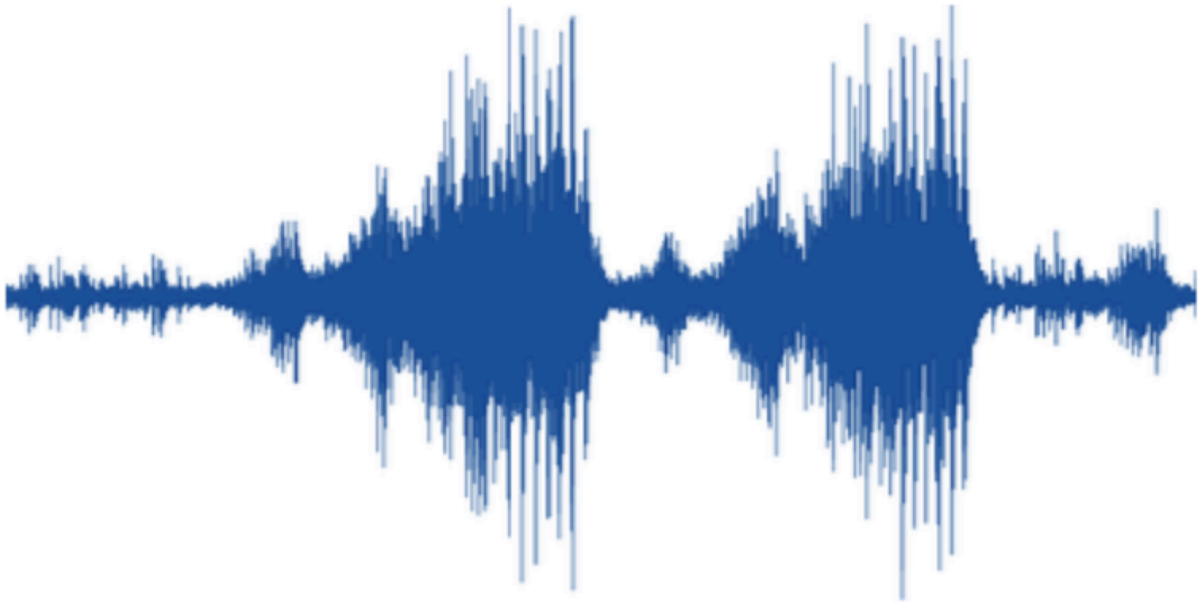
- 1,000 object classes (categories).
- Images:
  - 1.2 M train
  - 100k test.




(A. Krizhevsky et. al., 2012)

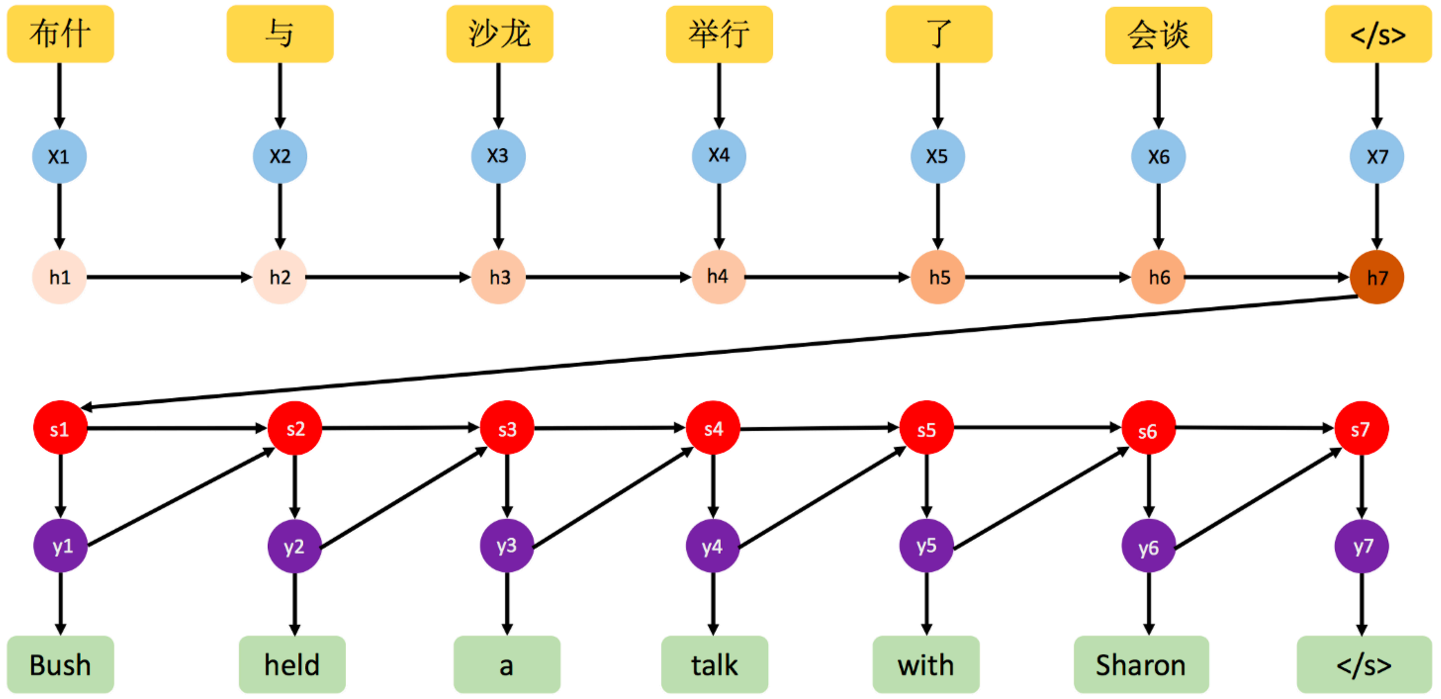
# Deep Learning on Regular Data

- As you see in this course, deep learning achieves great success in
  - Images
  - Speeches



# Deep Learning on Regular Data

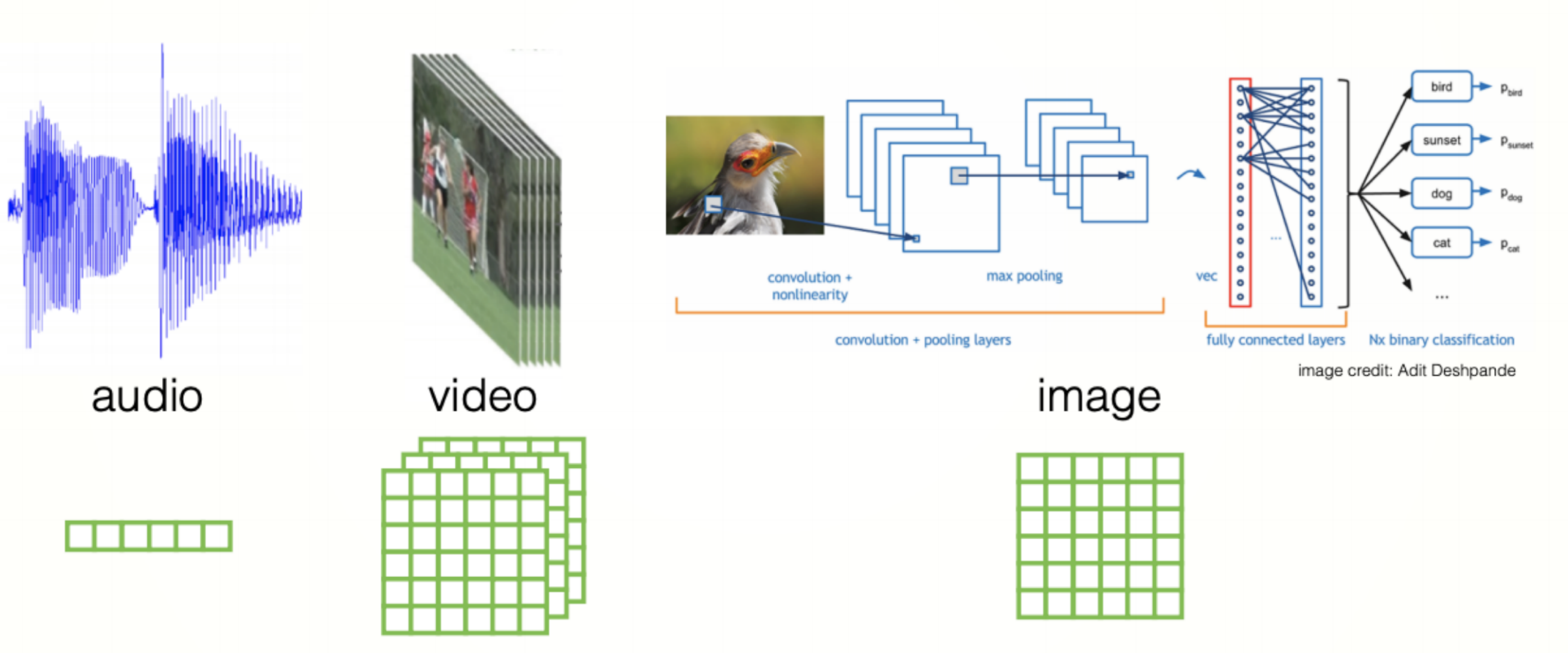
- As you see in this course, deep learning achieves great success in
  - Images
  - Speeches
  - NLP




(Sutskever et al., 2014)

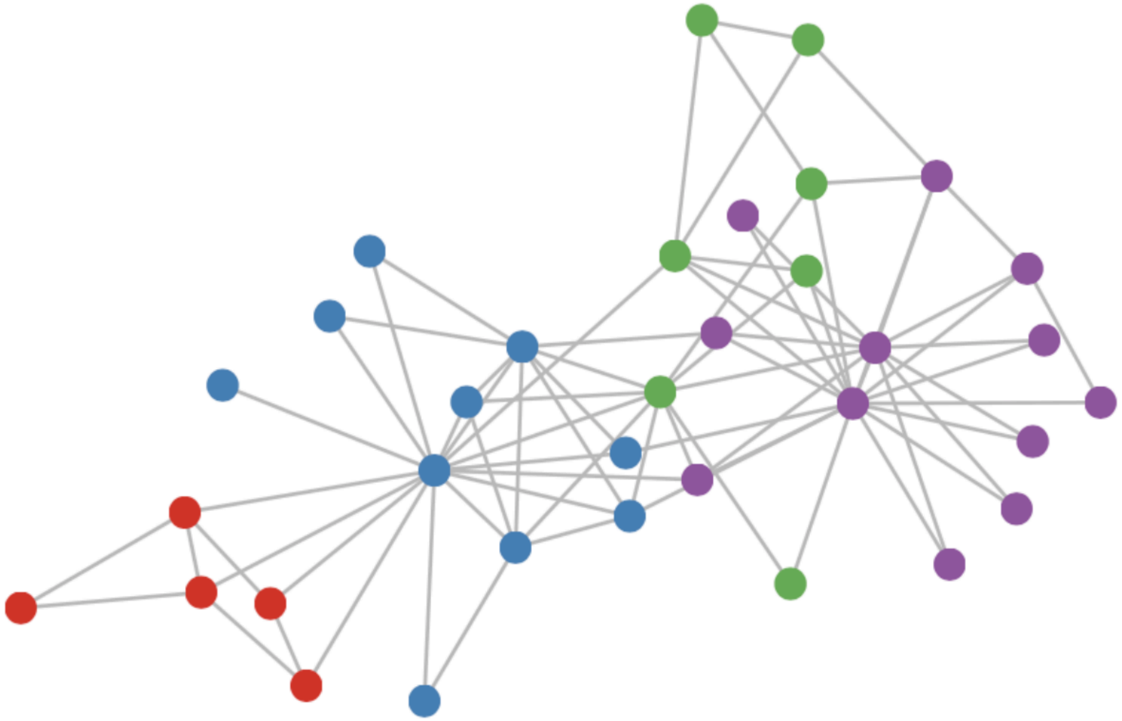
# Deep Learning on Regular Data

- (Convolutional) Neural Network explicitly exploit:
  - Translational invariance (local correlation)
  - Hierarchical compositionality



# Graph-structured data in real world

- Social Network Analysis

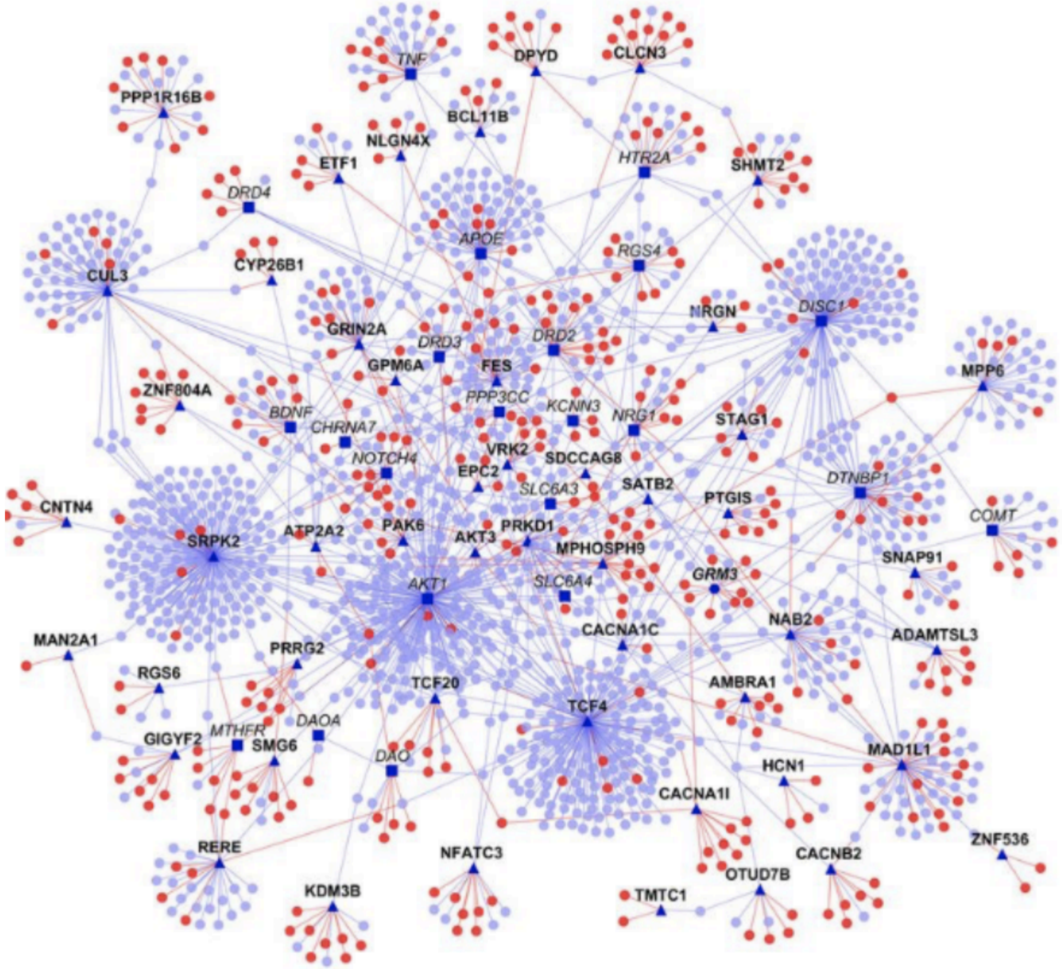


# Graph-structured data in real world

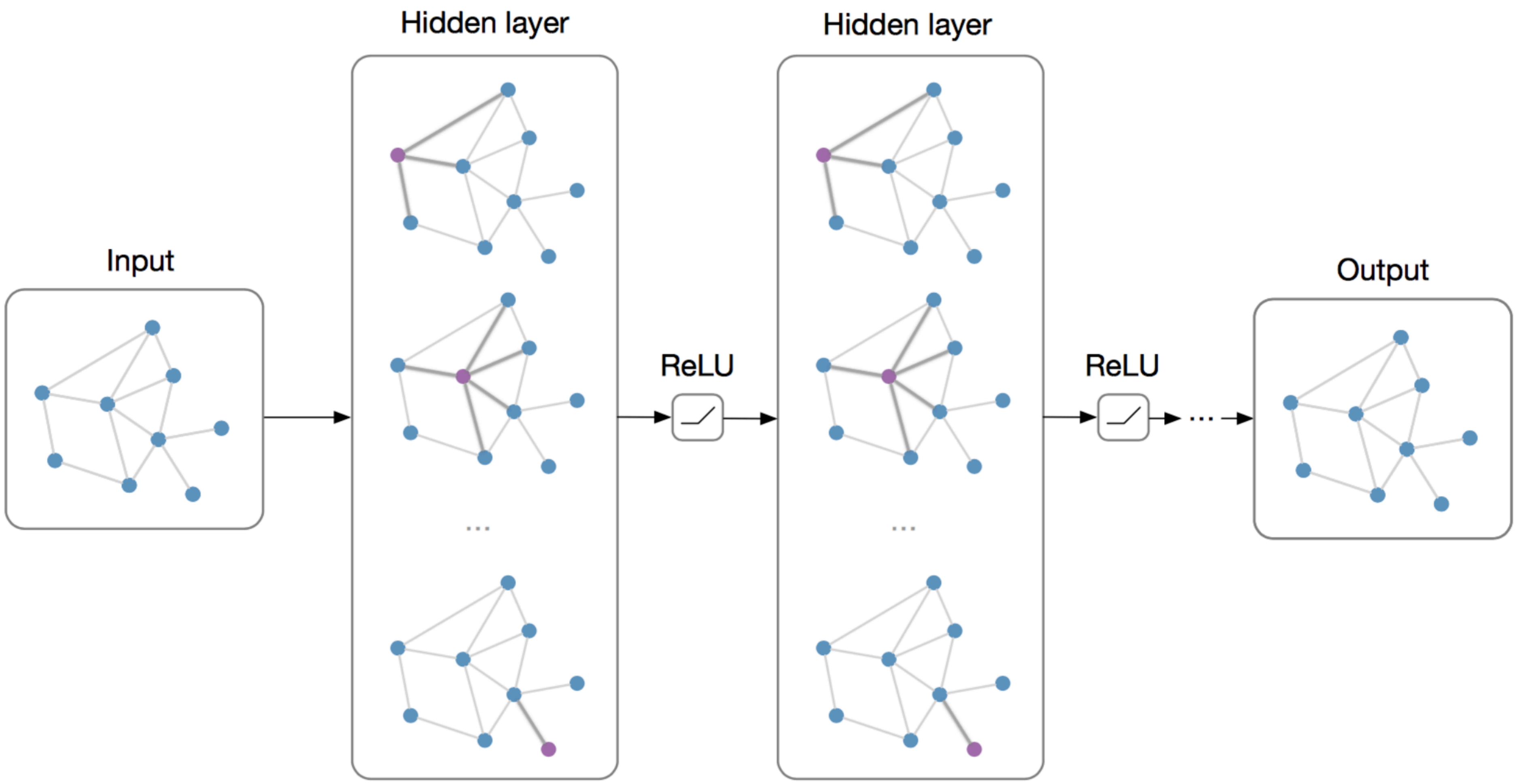
- Knowledge Graph

# Graph-structured data in real world

- Protein Interaction Network



# Graph Neural Networks

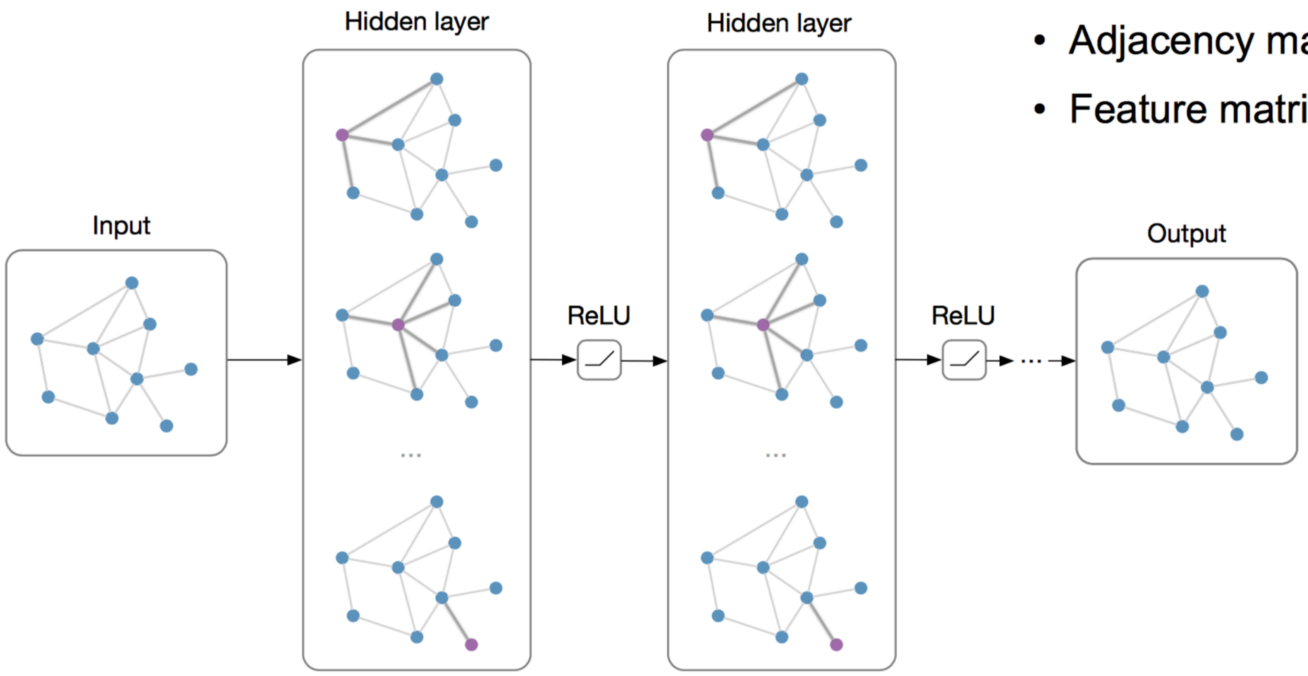


# Graph Neural Networks

- Pass massages between pairs of nodes and agglomerate
- Pass massages between nodes to refine node (and possibly edge) representations

**Notation:** $\mathcal{G} = (\mathbf{A}, \mathbf{X})$

- Adjacency matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$
- Feature matrix $\mathbf{X} \in \mathbb{R}^{N \times F}$

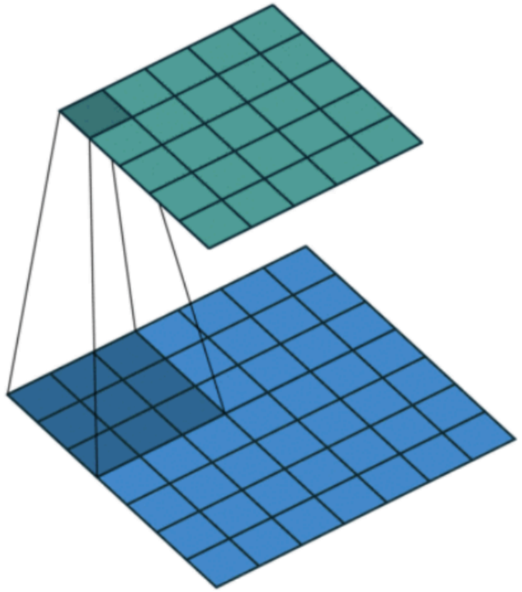


# Outlines

- Why Graph Neural Network (GNN)?
- **GNN, and the relation between GNN, CNN and transformer.**
- Applications of Graph Neural Network
  - On ‘classical’ network problems.
  - Emerging New Research Directions

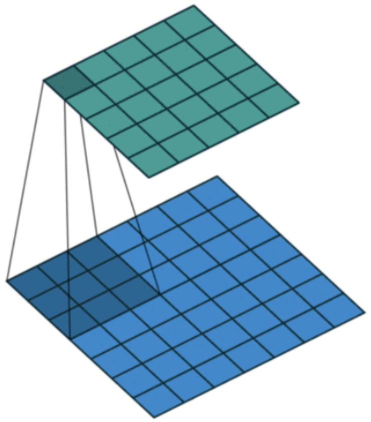
# Recap: CNN forward pass rule

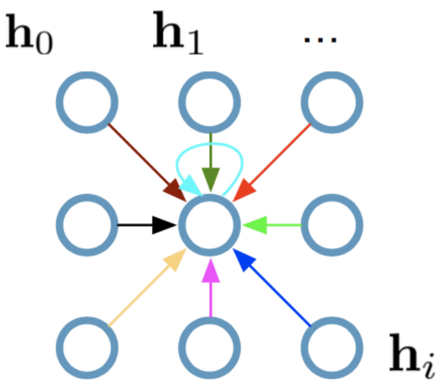
**Single CNN layer with 3x3 filter:**



# Recap: CNN forward pass rule

**Single CNN layer with 3x3 filter:**






**Update for a single pixel:**

- Transform messages individually $\mathbf{W}_i \mathbf{h}_i$
- Add everything up $\sum_i \mathbf{W}_i \mathbf{h}_i$

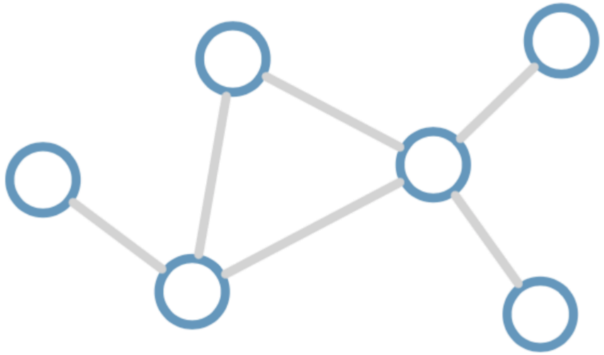
$\mathbf{h}_i \in \mathbb{R}^F$ are (hidden layer) activations of a pixel/node

**Full update:**

$$\mathbf{h}_4^{(l+1)} = \sigma\left(\mathbf{W}_0^{(l)}\mathbf{h}_0^{(l)} + \mathbf{W}_1^{(l)}\mathbf{h}_1^{(l)} + \cdots + \mathbf{W}_8^{(l)}\mathbf{h}_8^{(l)}\right)$$

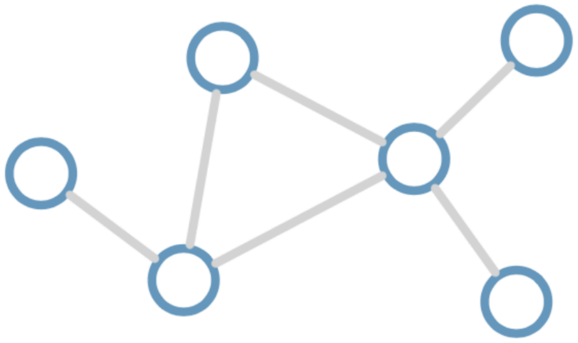
# From CNN forward to Graph Forward

Consider this undirected graph:

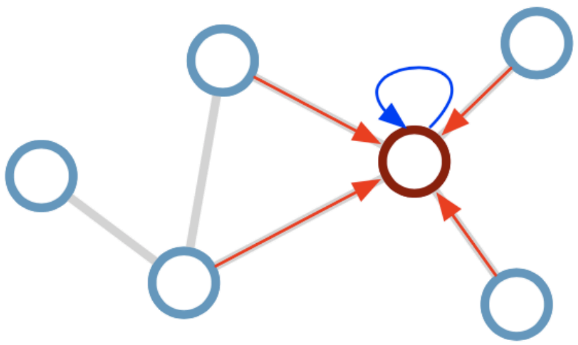


# From CNN forward to Graph Forward

Consider this
undirected graph:

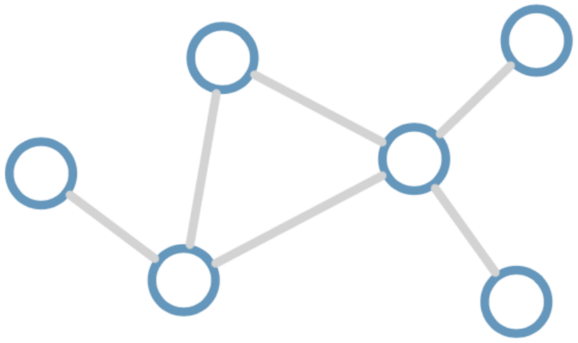


Calculate update
for node in red:

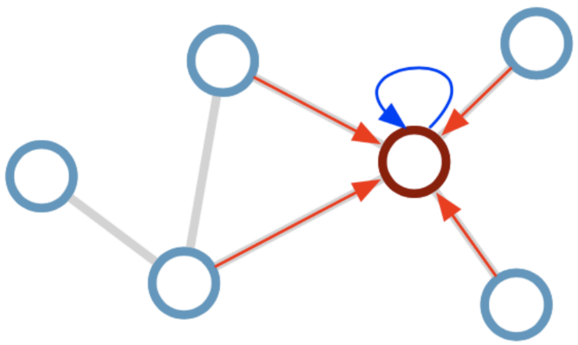


# From CNN forward to Graph Forward

Consider this
undirected graph:



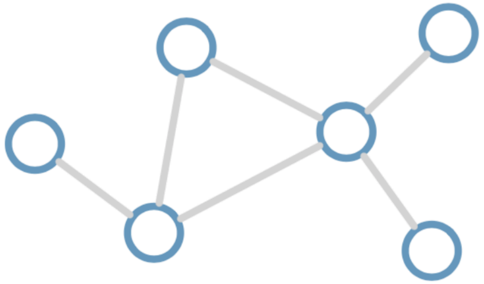
Calculate update
for node in red:



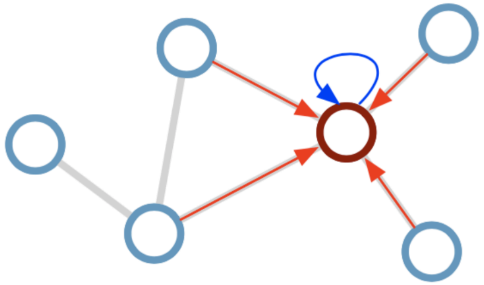
* slide from Thomas Kipf, **University of Amsterdam**

# From CNN forward to Graph Forward

Consider this undirected graph:



Calculate update for node in red:



**Desirable properties:**

- Weight sharing over all locations
- Invariance to permutations
- Linear complexity O(E)
- Applicable both in transductive and inductive settings

**Update rule:**

$$\mathbf{h}_i^{(l+1)} = \sigma\left(\mathbf{h}_i^{(l)}\mathbf{W}_0^{(l)} + \sum_{j\in\mathcal{N}_i}\frac{1}{c_{ij}}\mathbf{h}_j^{(l)}\mathbf{W}_1^{(l)}\right)$$

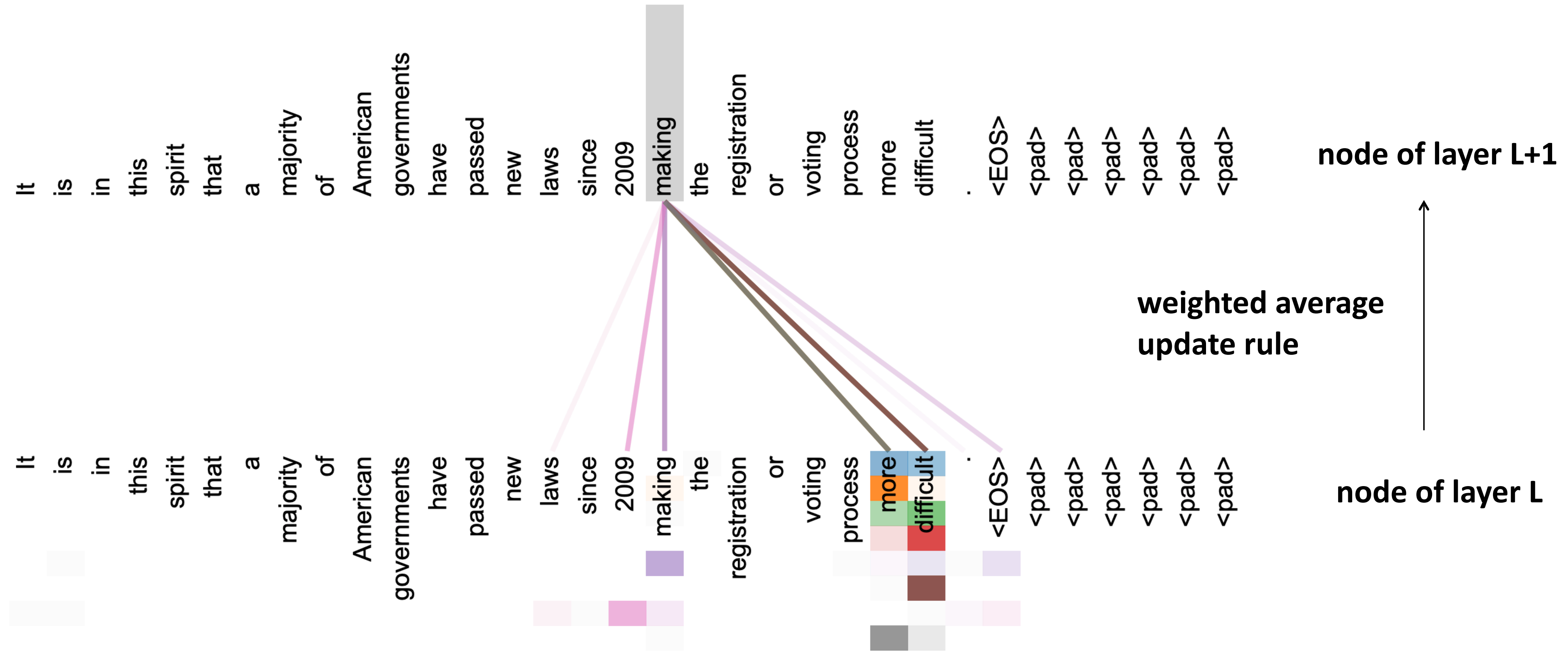
**Scalability: subsample messages** [Hamilton et al., NIPS 2017]

$\mathcal{N}_i$ : neighbor indices

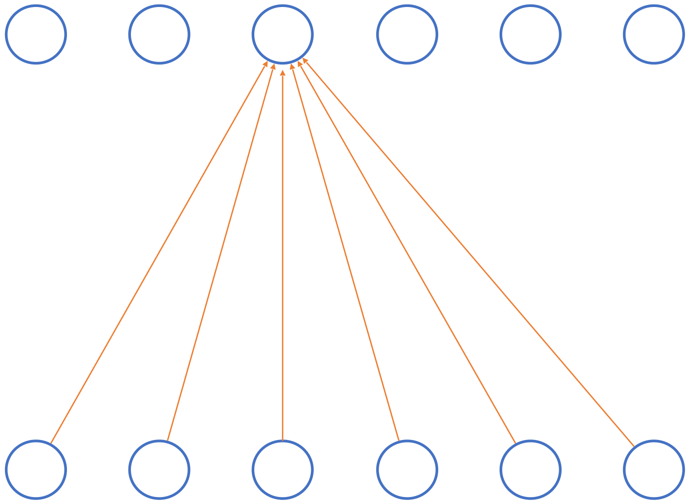
$c_{ij}$ : norm. constant (fixed/trainable)

* slide from Thomas Kipf, **University of Amsterdam**

# Relationship between transformer and GNN



# Relationship between transformer and GNN



$$h_i^{\ell+1} = \text{Attention}\left(Q^\ell h_i^\ell\ , K^\ell h_j^\ell\ , V^\ell h_j^\ell\right),$$

$$i.e.,\ h_i^{\ell+1} = \sum_{j \in \mathcal{S}} w_{ij}\left(V^\ell h_j^\ell\right),$$

$$\text{where } w_{ij} = \text{softmax}_j\left(Q^\ell h_i^\ell \cdot K^\ell h_j^\ell\right),$$

# Outlines

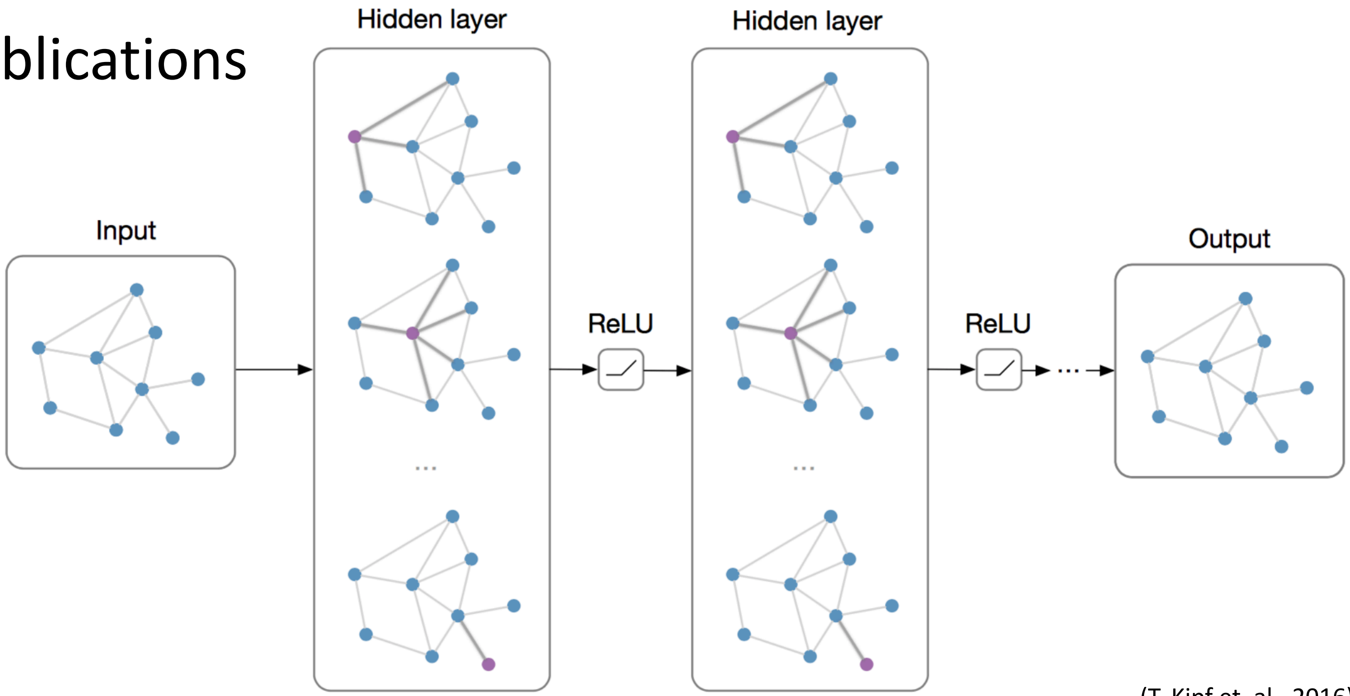
- Why Graph Neural Network (GNN)?
- GNN, and the relation between GNN, CNN and transformer.
- **Applications of Graph Neural Network**
  - On ‘classical’ network problems.
  - Emerging New Research Directions

# GNN on network classification

- node classification
- graph classification
- link prediction

# GNN on node classification

- Citation Network Node Classification
  - Each node is a publication / document
  - Edges stand for citation
  - only some of the publication have labels
  - infer the category of other publications




(T. Kipf et. al., 2016)

# GNN on network classification

- Molecular Classification

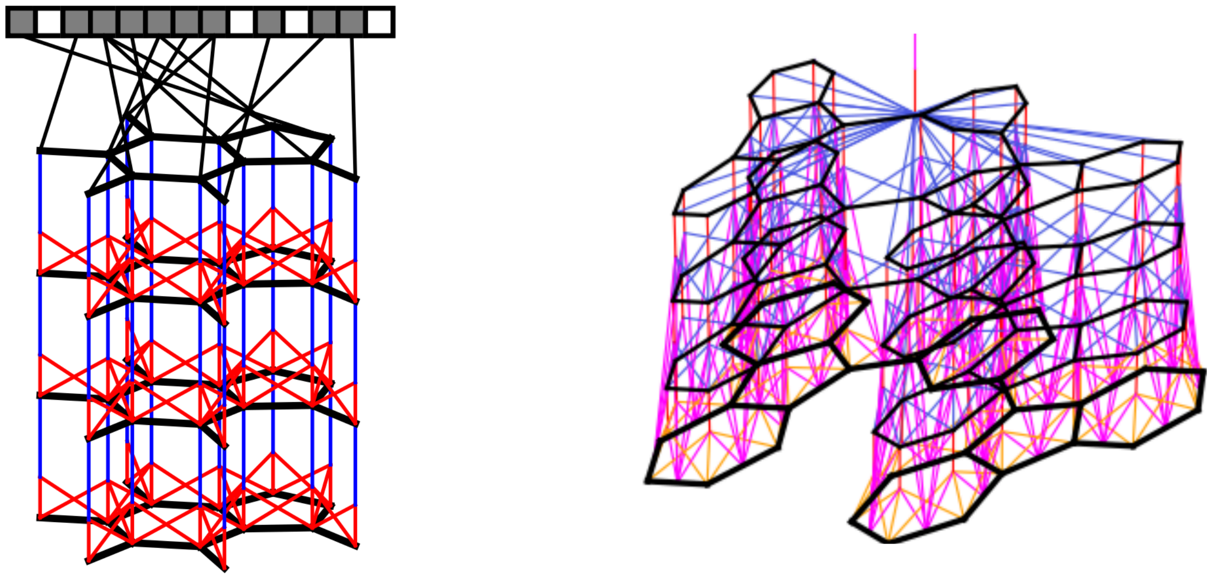


Figure 1: *Left*: A visual representation of the computational graph of both standard circular fingerprints and neural graph fingerprints. First, a graph is constructed matching the topology of the molecule being fingerprinted, in which nodes represent atoms, and edges represent bonds. At each layer, information flows between neighbors in the graph. Finally, each node in the graph turns on one bit in the fixed-length fingerprint vector. *Right*: A more detailed sketch including the bond information used in each operation.

(D. Duvenaud et. al., 2015)

# GNN on link prediction

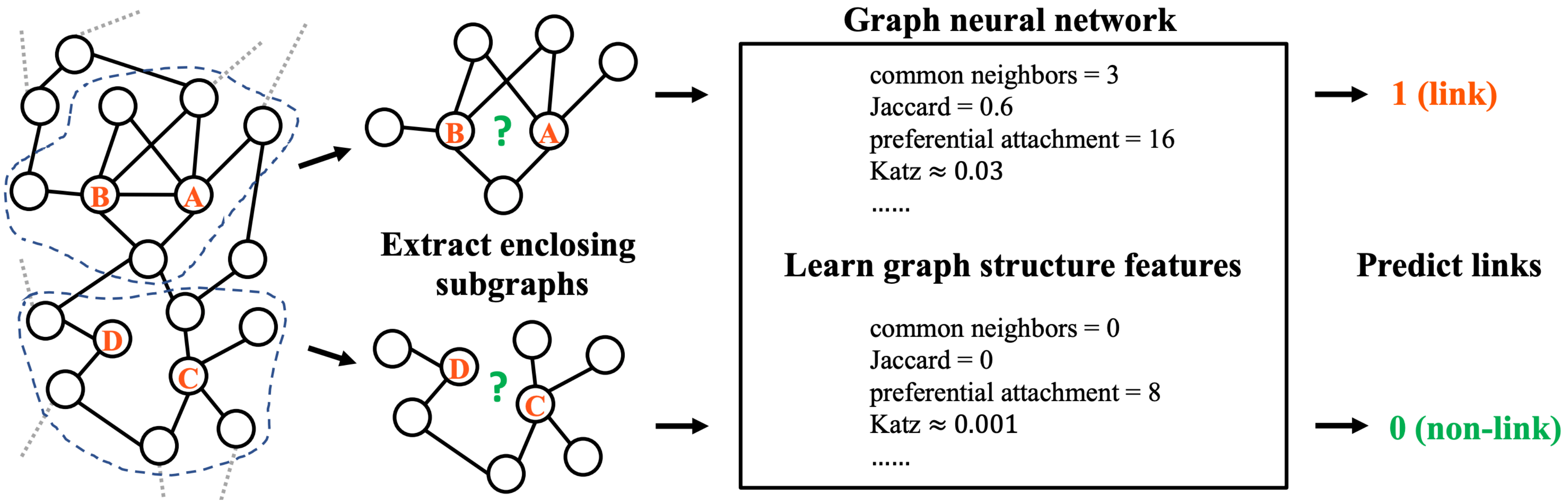
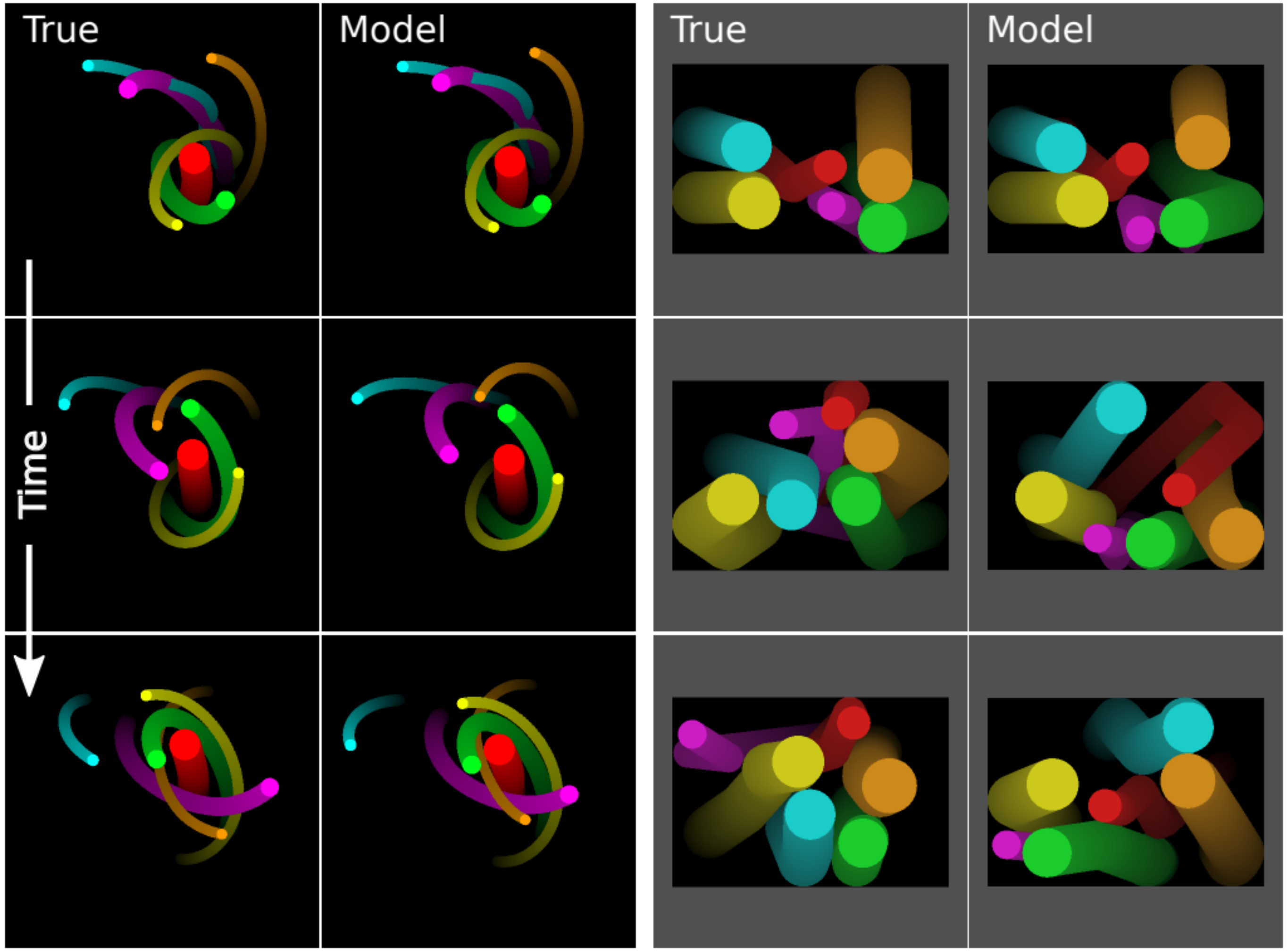



**Figure 1:** The SEAL framework. For each target link, SEAL extracts a local enclosing subgraph around it, and uses a GNN to learn general graph structure features for link prediction. Note that the heuristics listed inside the box are just for illustration – the learned features may be completely different from existing heuristics.

(M. Zhang et. al., 2018)

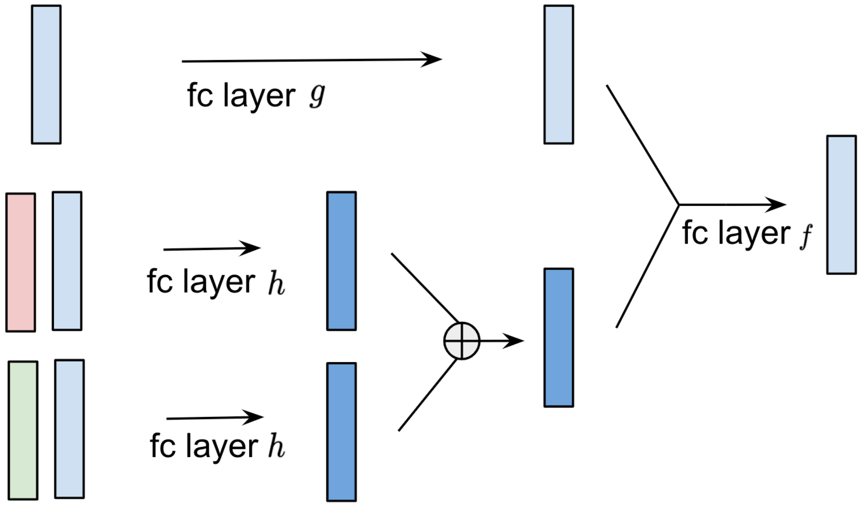
# Interaction Systems (Object – Object)

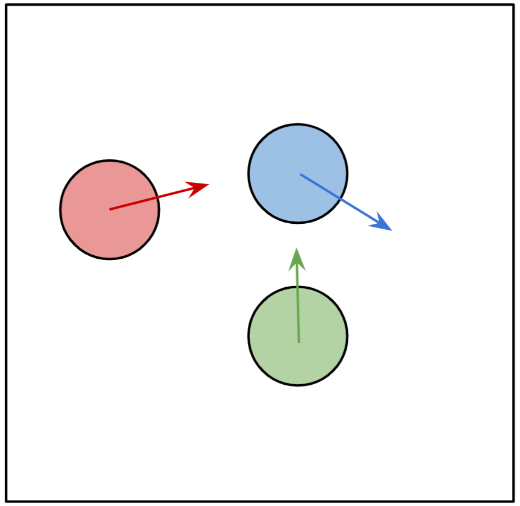


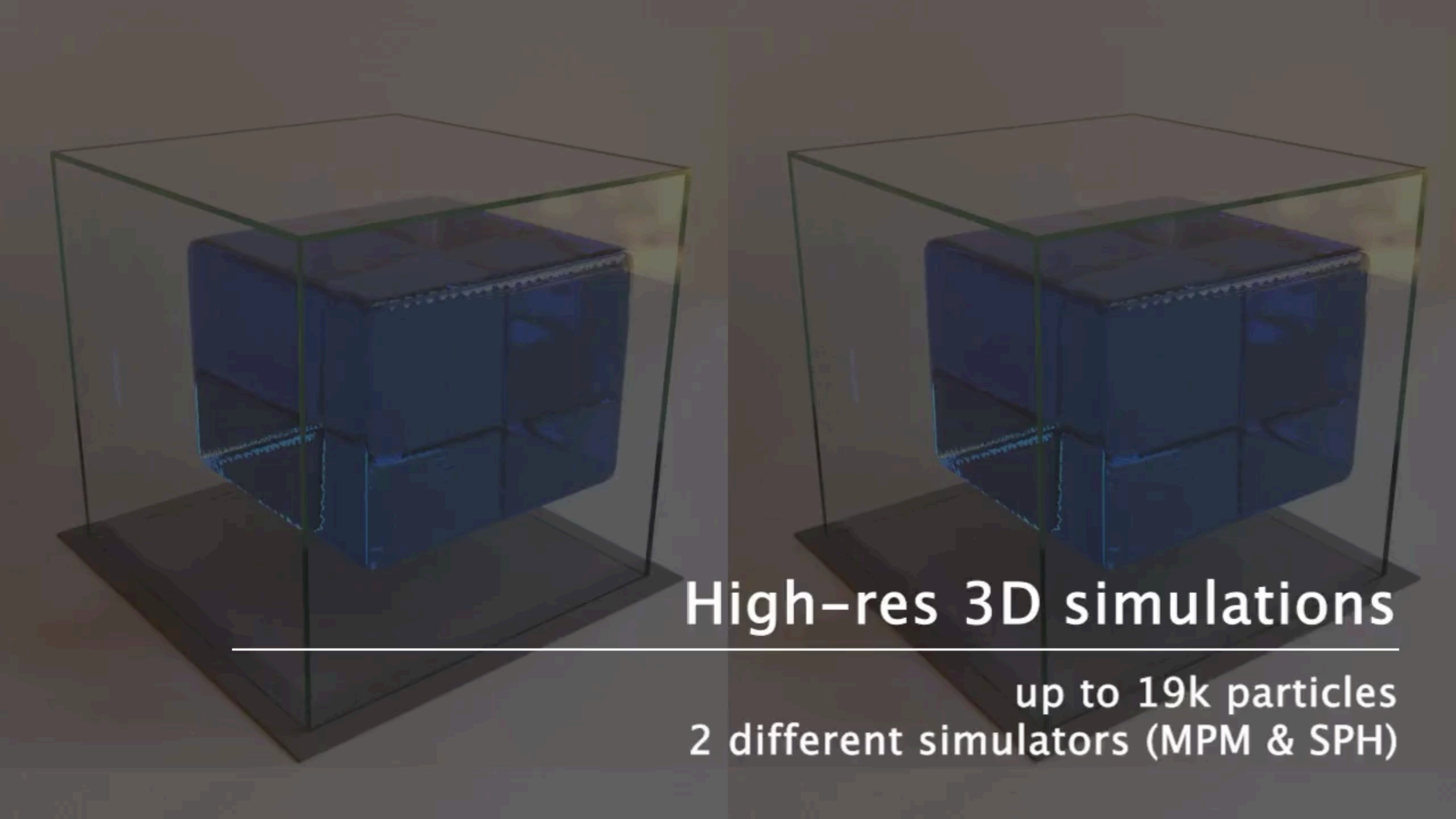

(P. Battaglia, 2016)

# Interaction Systems (Object – Object)

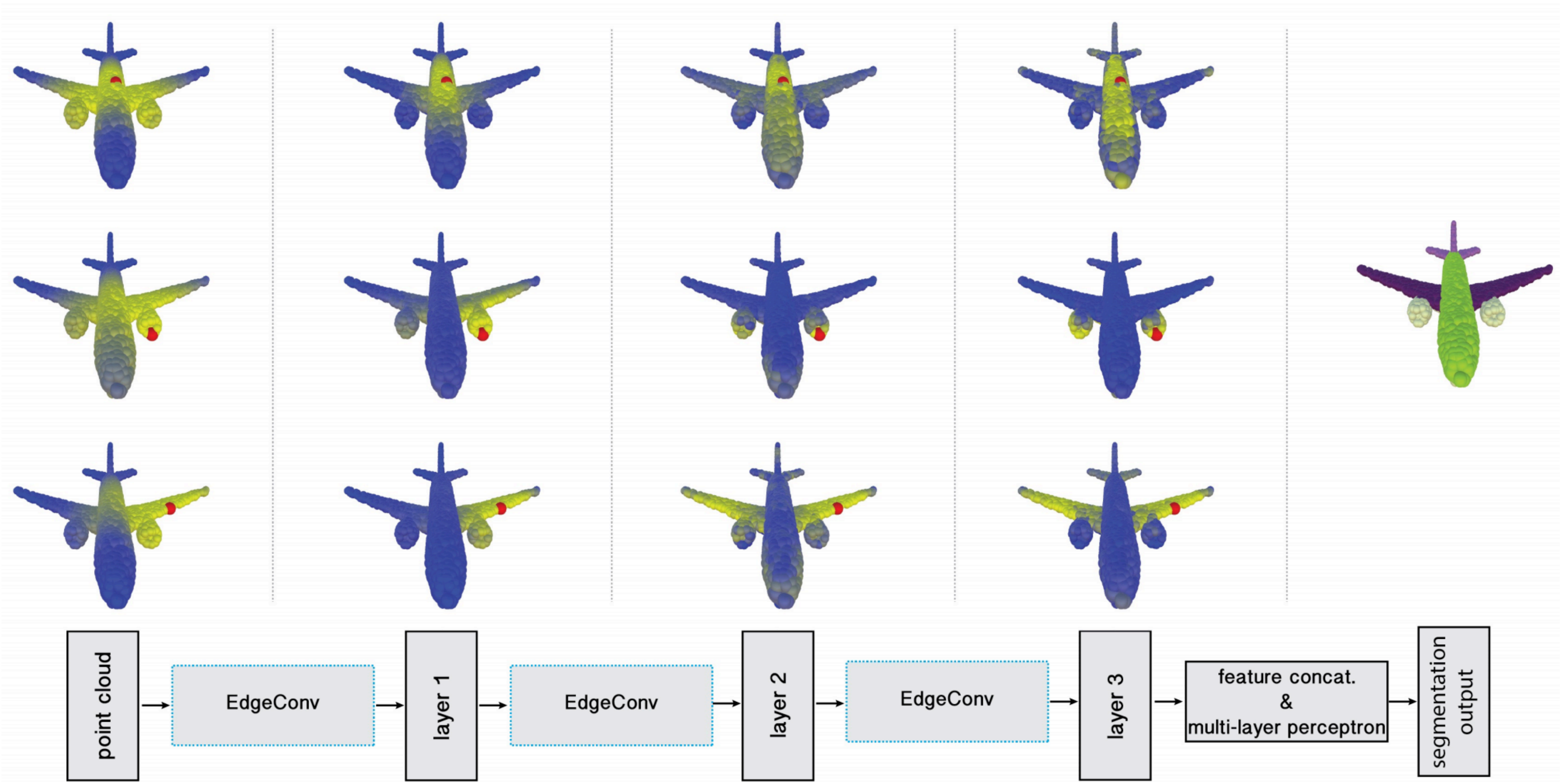
- If we want to predict the future movement of the blue billiard
  - self-dynamics: $g$ (Newton's first law)
  - relation-dynamics: $h$ (Newton's second law)
  - summation relation dynamics (Parallelogram law)
  - Aggregate the above: $f$



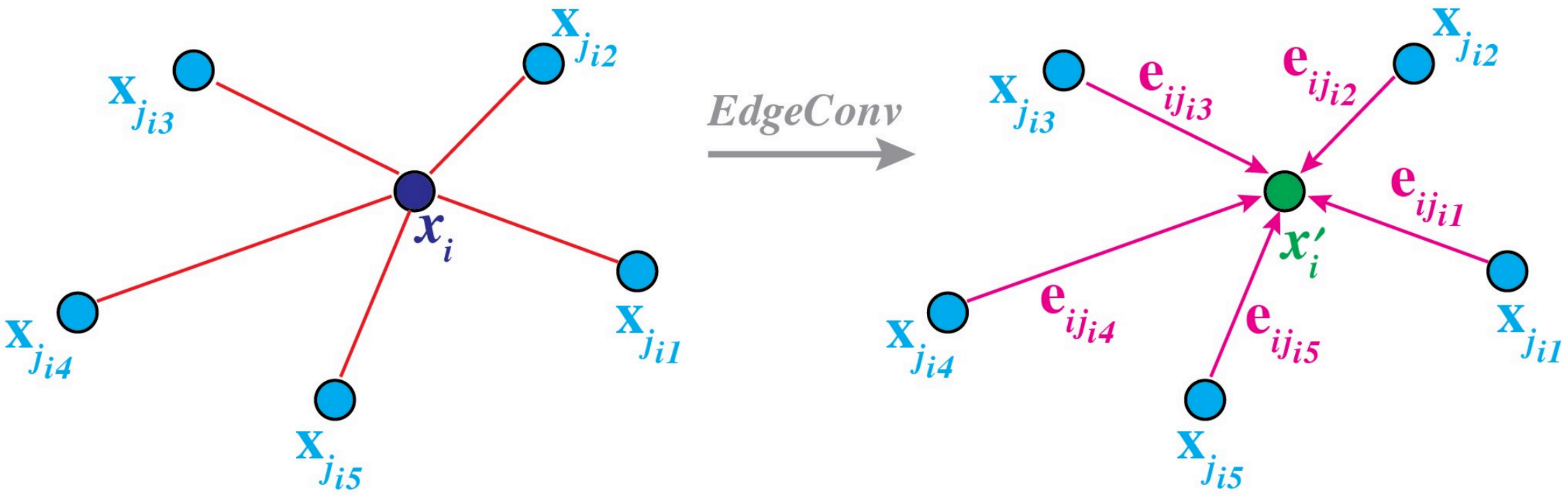

High-res 3D simulations
up to 19k particles
2 different simulators (MPM & SPH)

# GNN on 3D Point Cloud




(Y. Wang et. al., 2018)

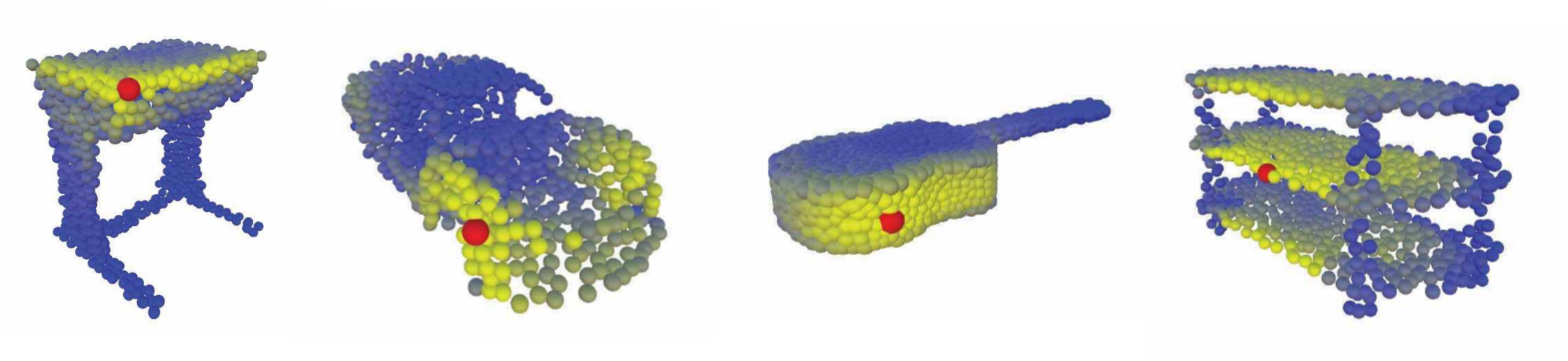
# GNN on 3D Point Cloud




(Y. Wang et. al., 2018)

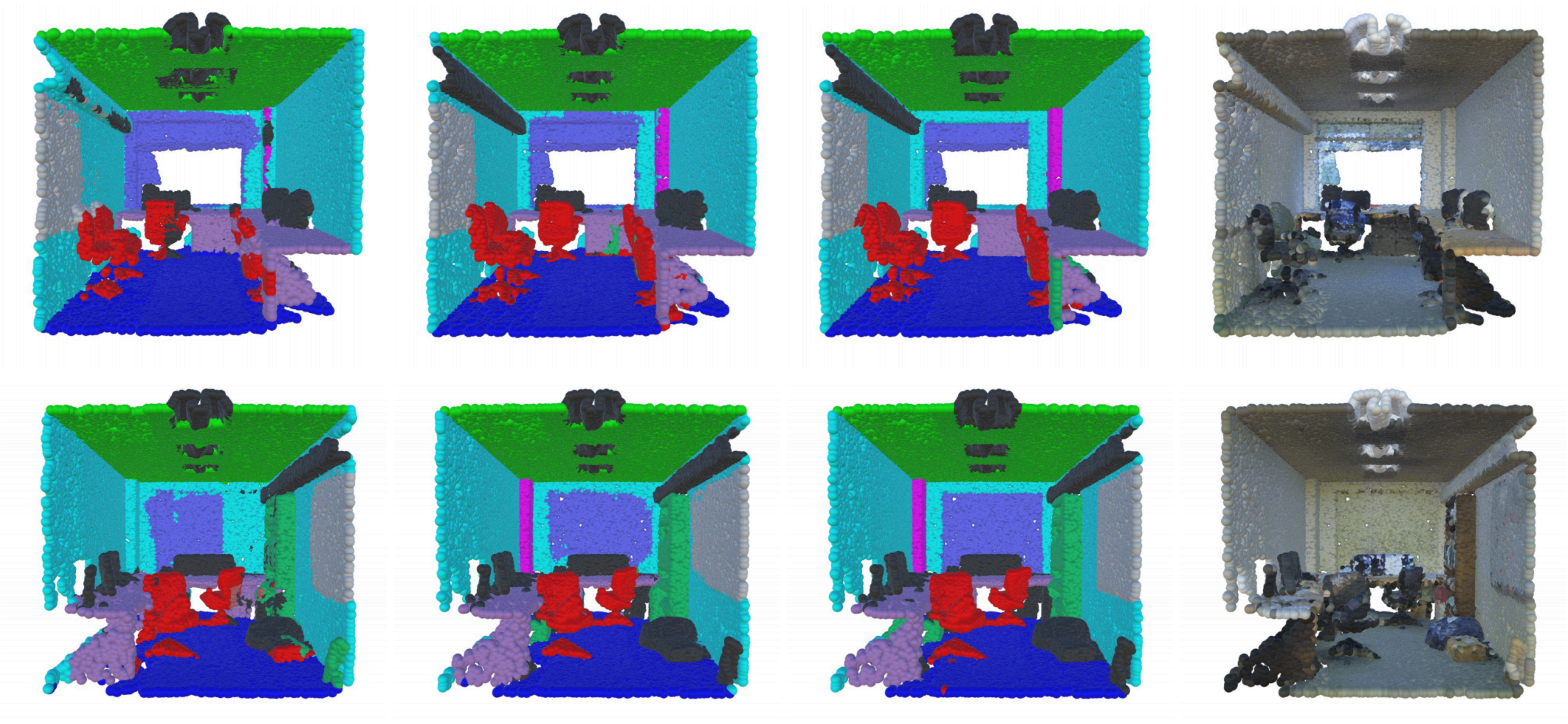
# GNN on 3D Point Cloud

- Aggregated neighborhood



(Y. Wang et. al., 2018)

# GNN on 3D Point Cloud



PointNet Ours Ground truth Real color

(Y. Wang et. al., 2018)

# Video Recognition

(X. Wang et. al., 2017)

# Video Recognition

- Something like a transformer

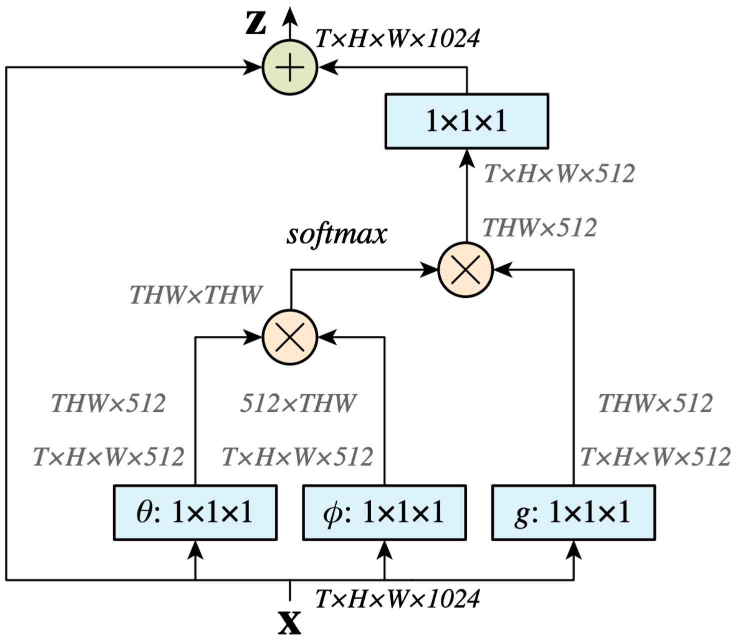
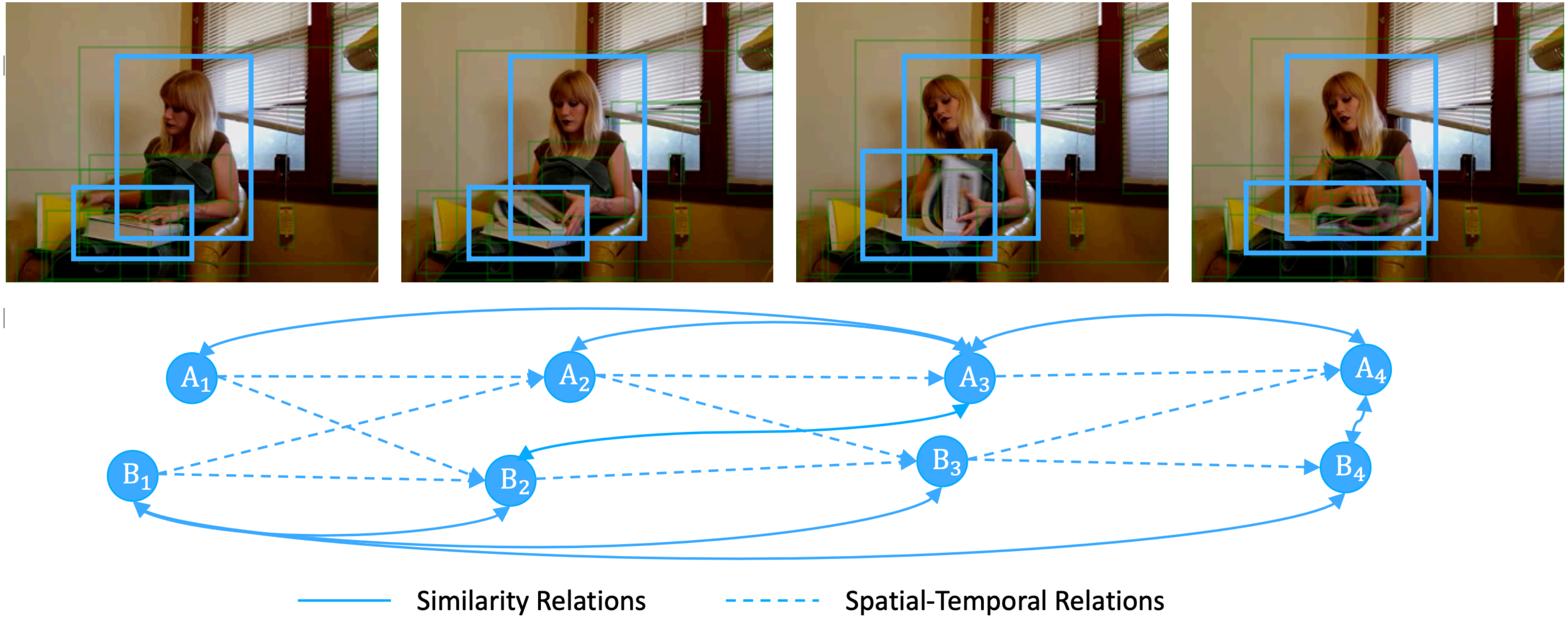



Figure 2. A spacetime **non-local block**. The feature maps are shown as the shape of their tensors, *e.g.*, $T\times H\times W\times 1024$ for 1024 channels (proper reshaping is performed when noted). "$\otimes$" denotes matrix multiplication, and "$\oplus$" denotes element-wise sum. The softmax operation is performed on each row. The blue boxes denote $1\times1\times1$ convolutions. Here we show the embedded Gaussian version, with a bottleneck of 512 channels. The vanilla Gaussian version can be done by removing $\theta$ and $\phi$, and the dot-product version can be done by replacing softmax with scaling by $1/N$.

(X. Wang et. al., 2017)

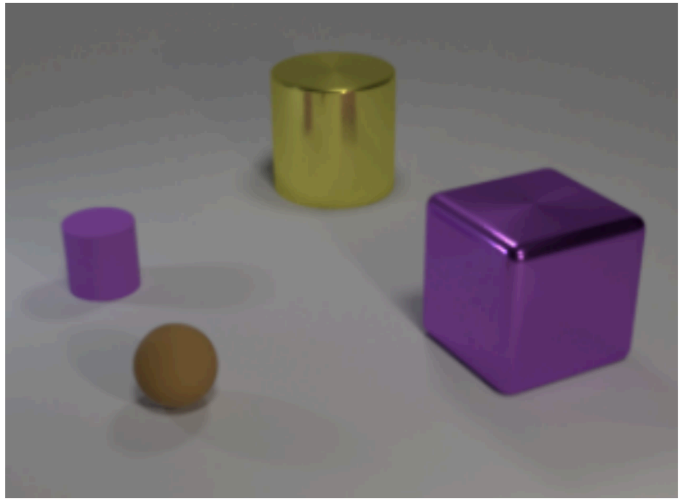
# Video Recognition




(X. Wang et. al., 2018)

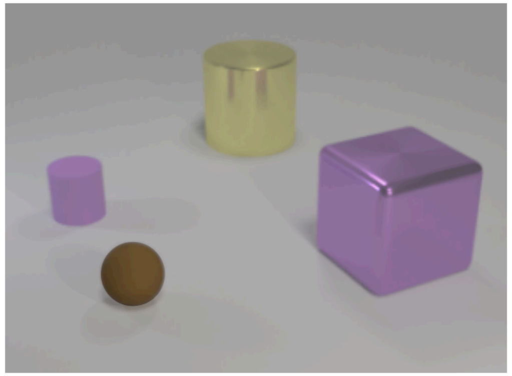
# Visual Reasoning

**Original Image:**



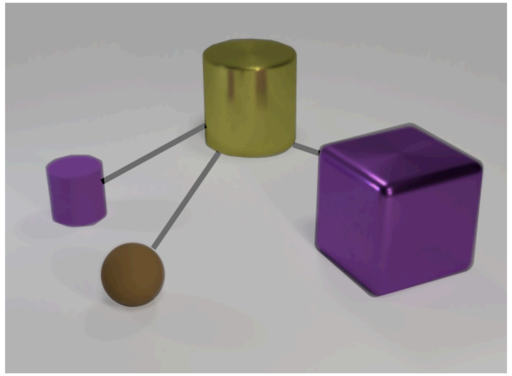
**Non-relational question:**

`What is the size of the brown sphere?`

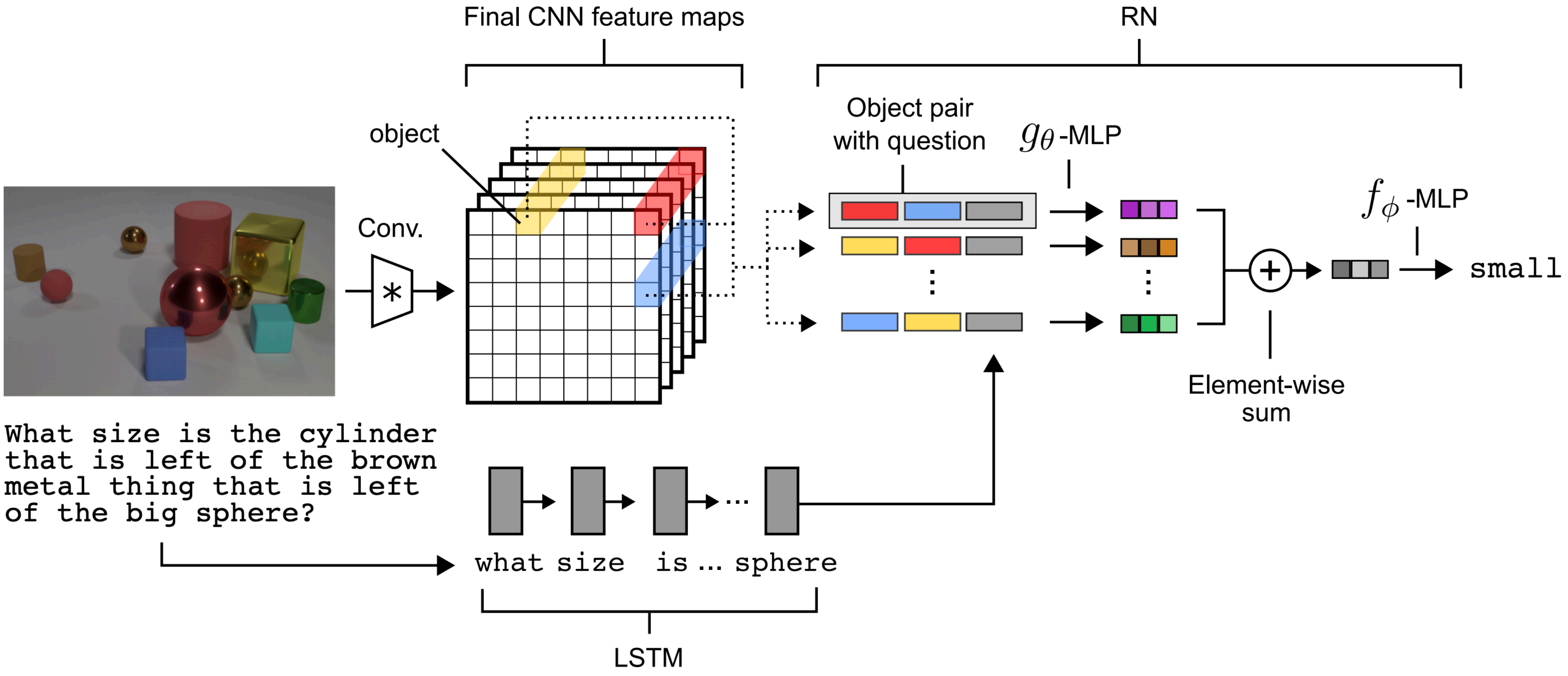


**Relational question:**

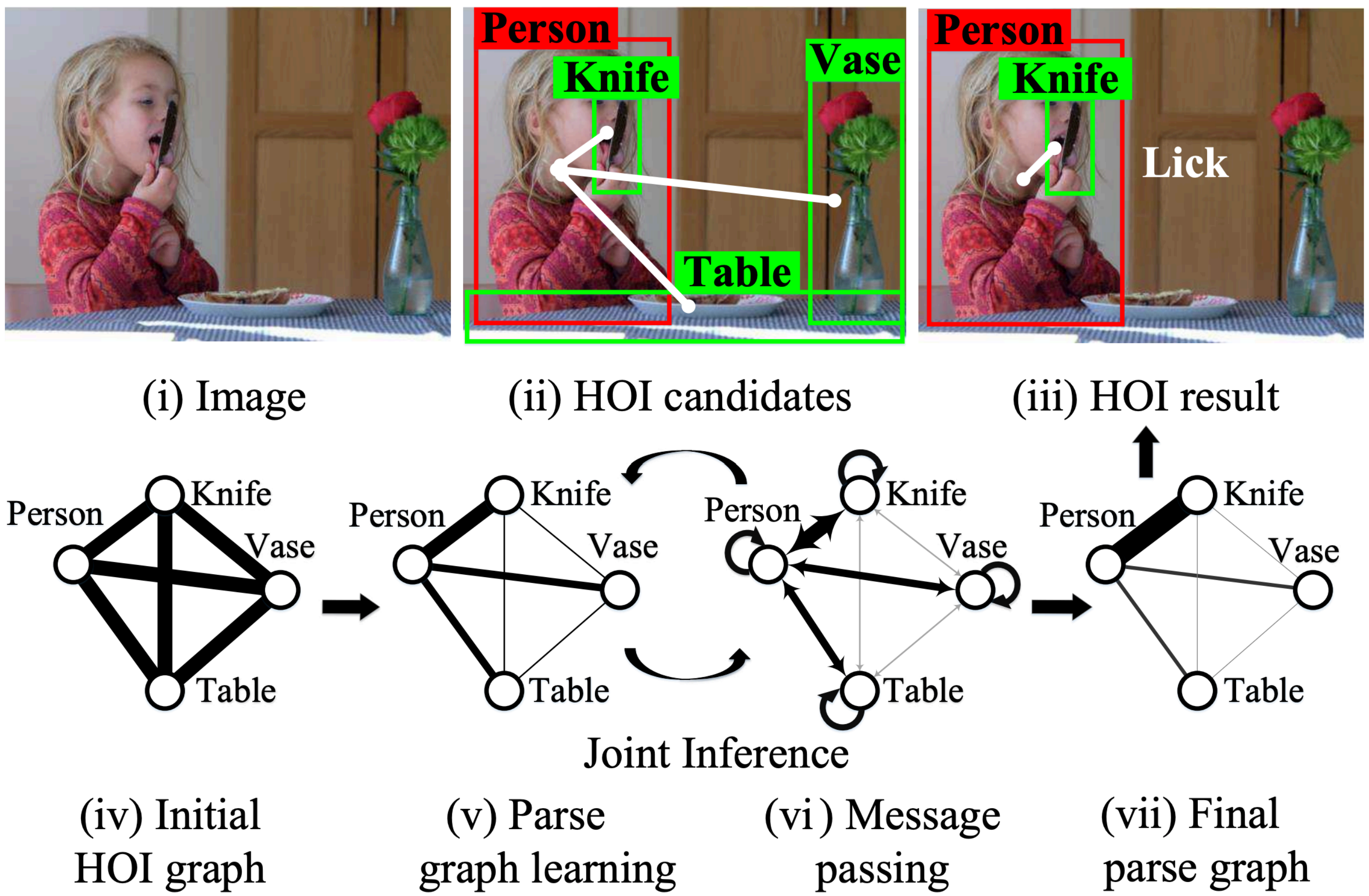
`Are there any rubber things that have the same size as the yellow metallic cylinder?`



# Visual Reasoning

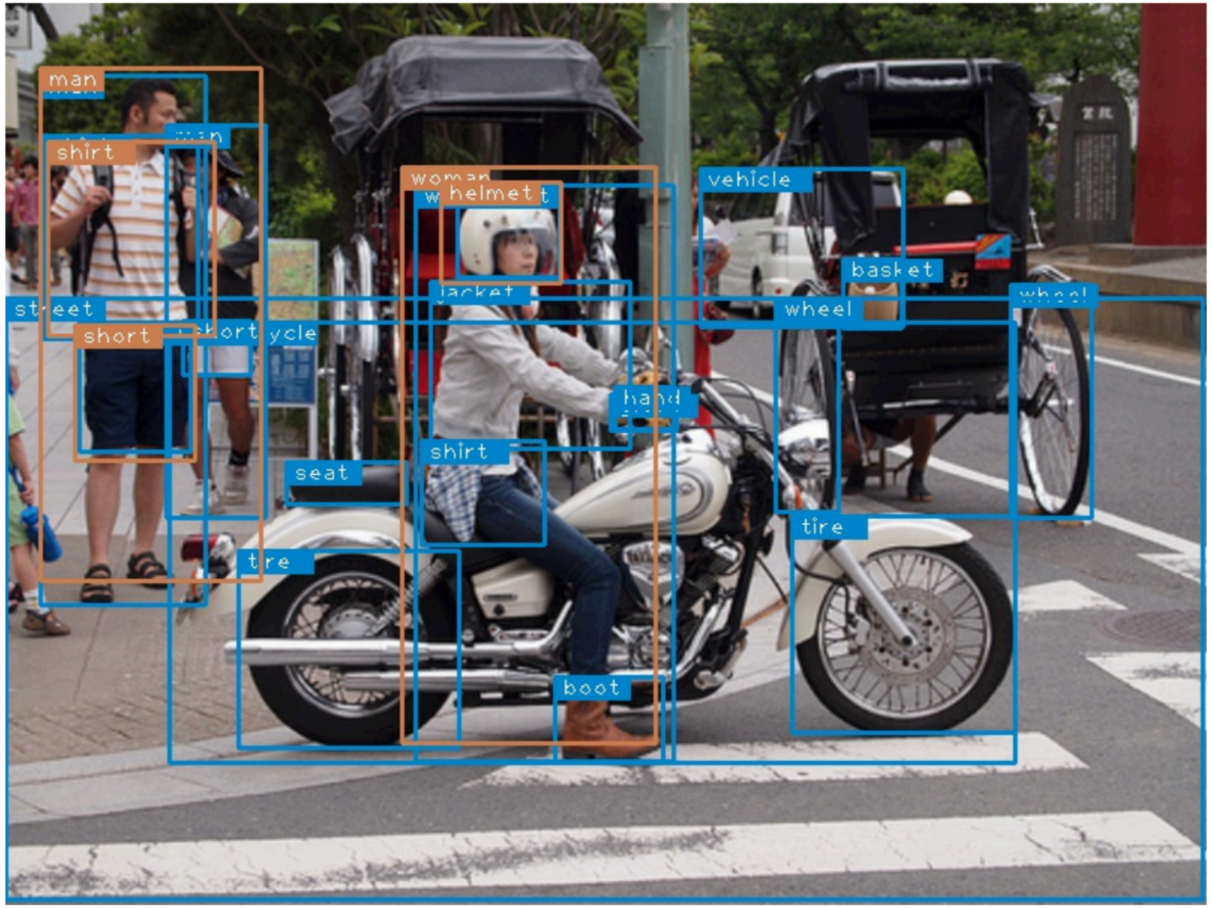


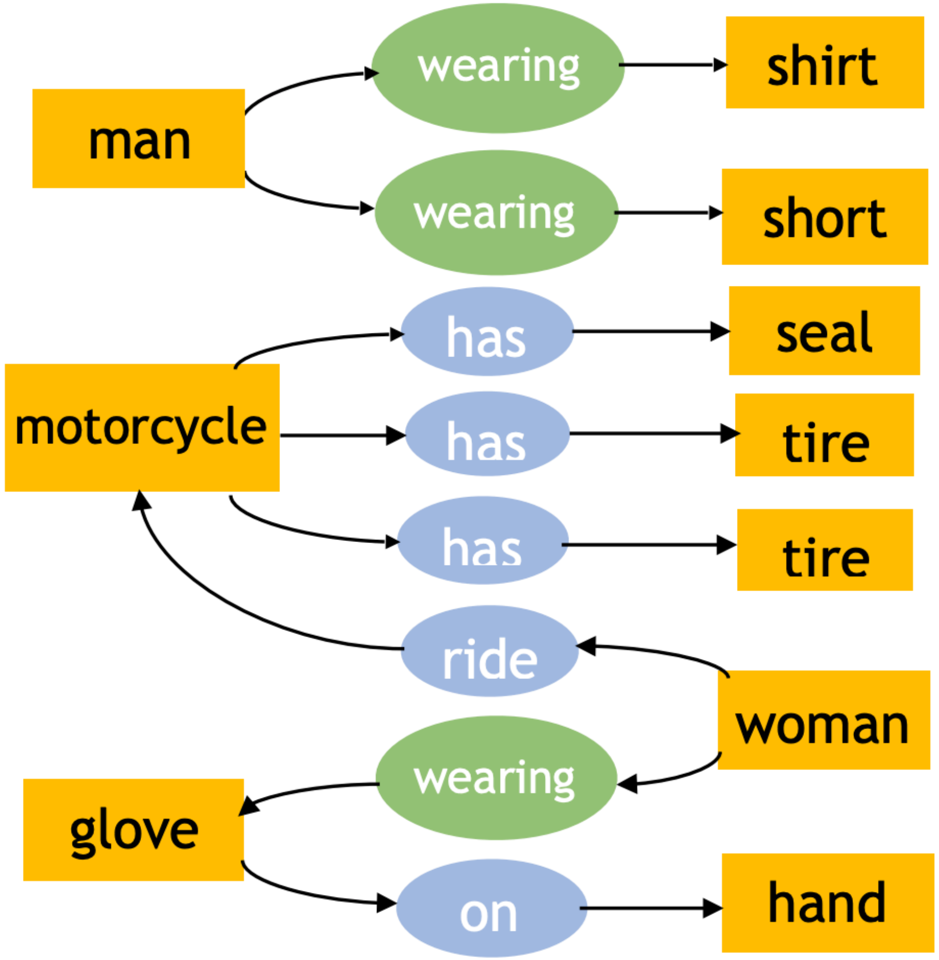
# Interaction Systems (Agent – Object)




**(a) Human-Object Interaction Detection in Still Images**

# Scene Graph Generation



# Scene Graph Generation