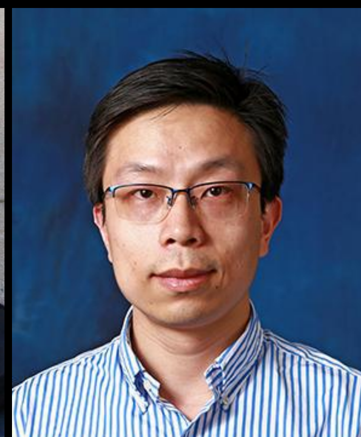
3D Graph Neural Networks for RGBD Semantic Segmentation

Xiaojuan Qi[1], Renjie Liao[2,3], Jiaya Jia[1,4], Sanja Fidler[2], Raquel Urtasun[2,3]

[1]The Chinese University of Hong Kong [2] University of Toronto
[3]Uber Adevanced Technologies Group [4]Youtu Lab, Tencent

Depth Sensors

Microsoft Kinect

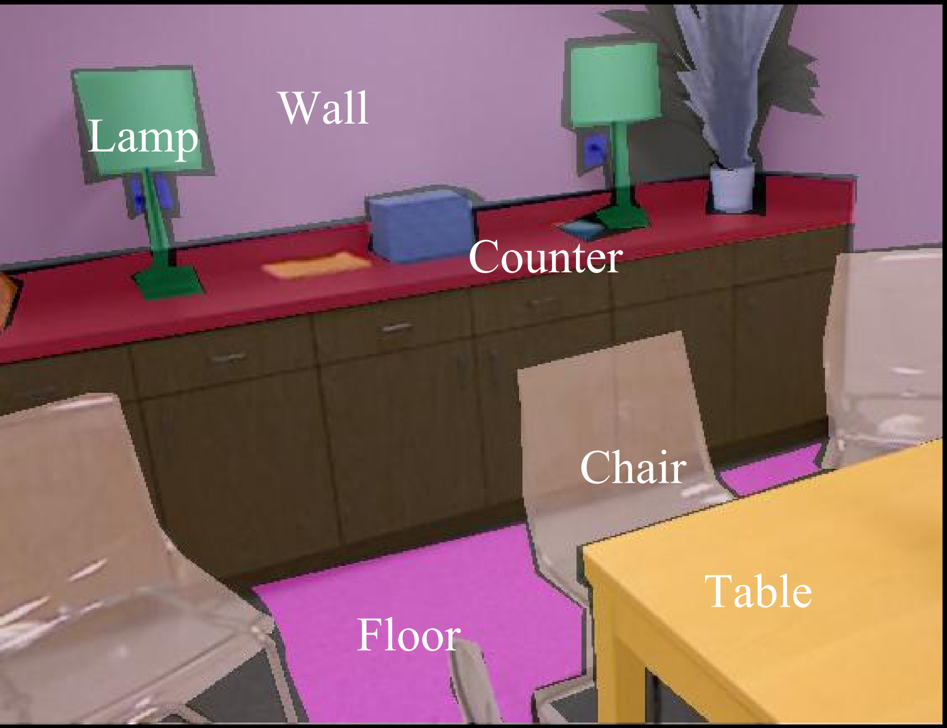
Intel RealSense

Depth information available

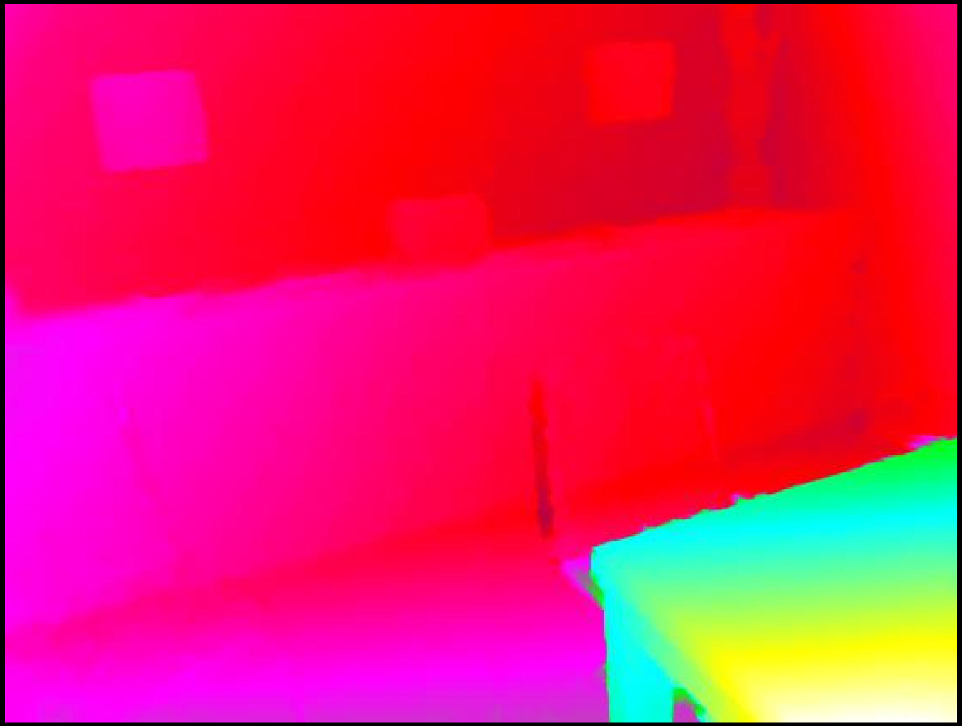
Dual Camera Smartphone

Dual Camera UAV

Problem



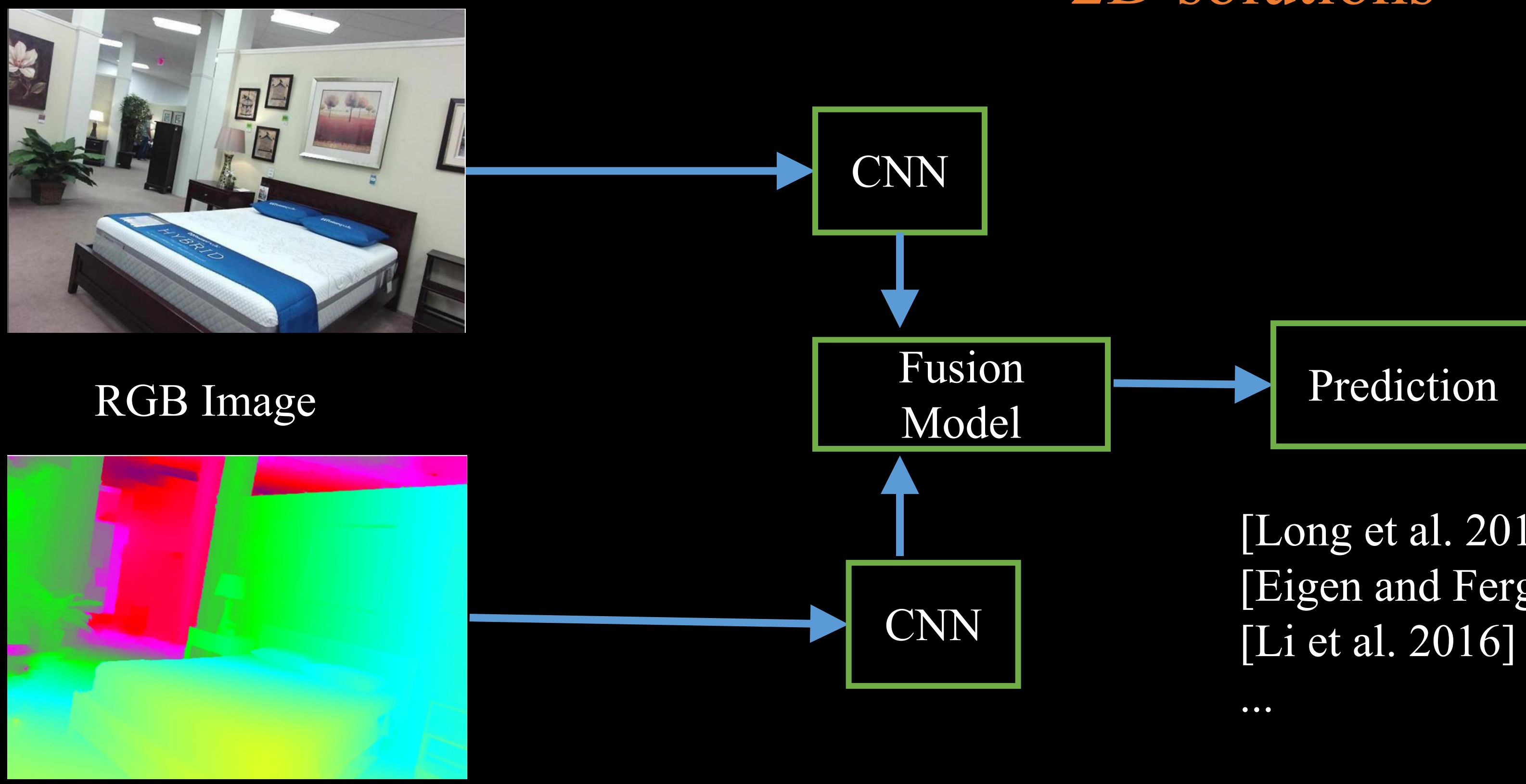
Segmentation map



Depth Image

Previous Approaches

2D solutions

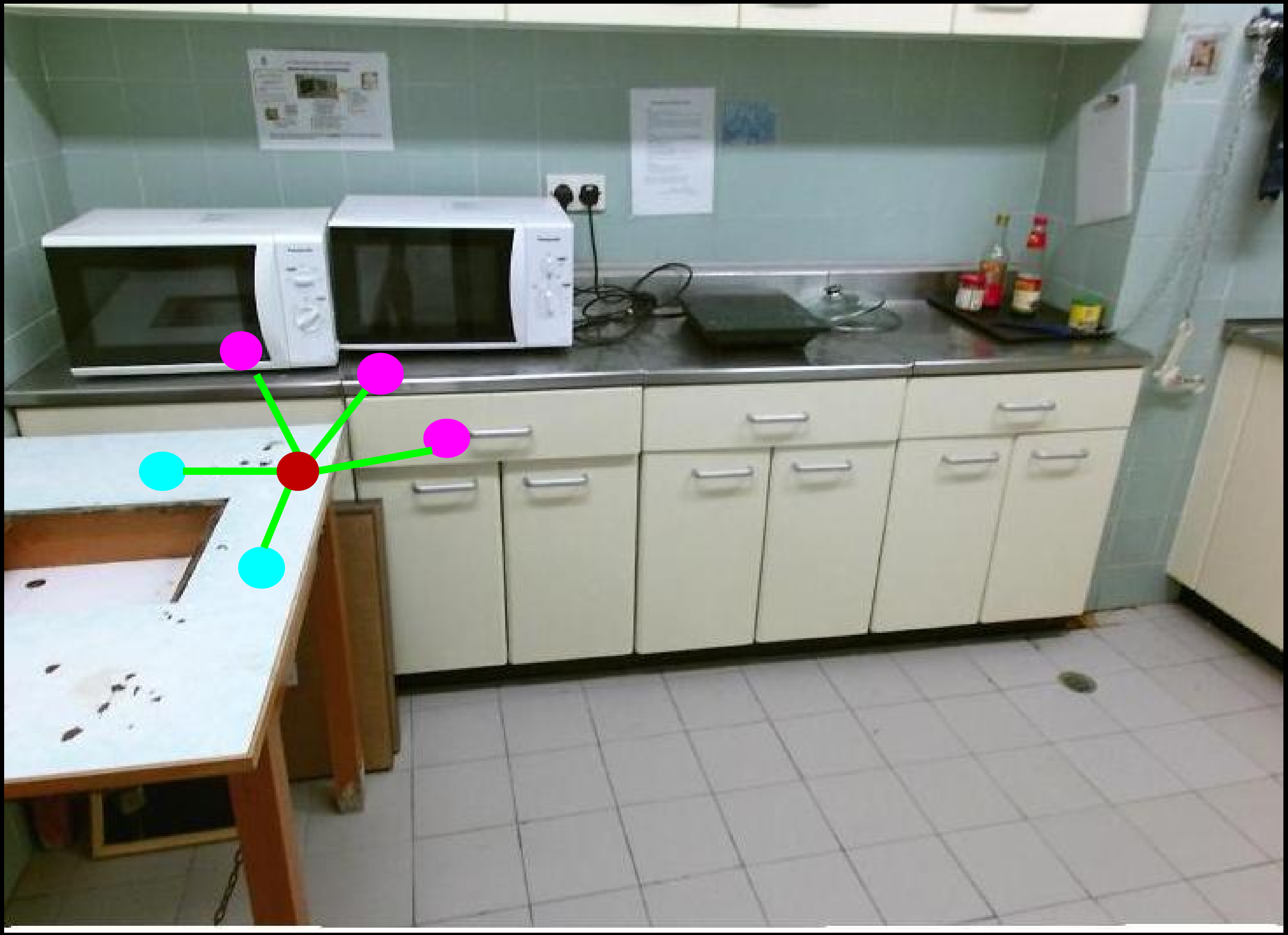


RGB Image

Depth Image

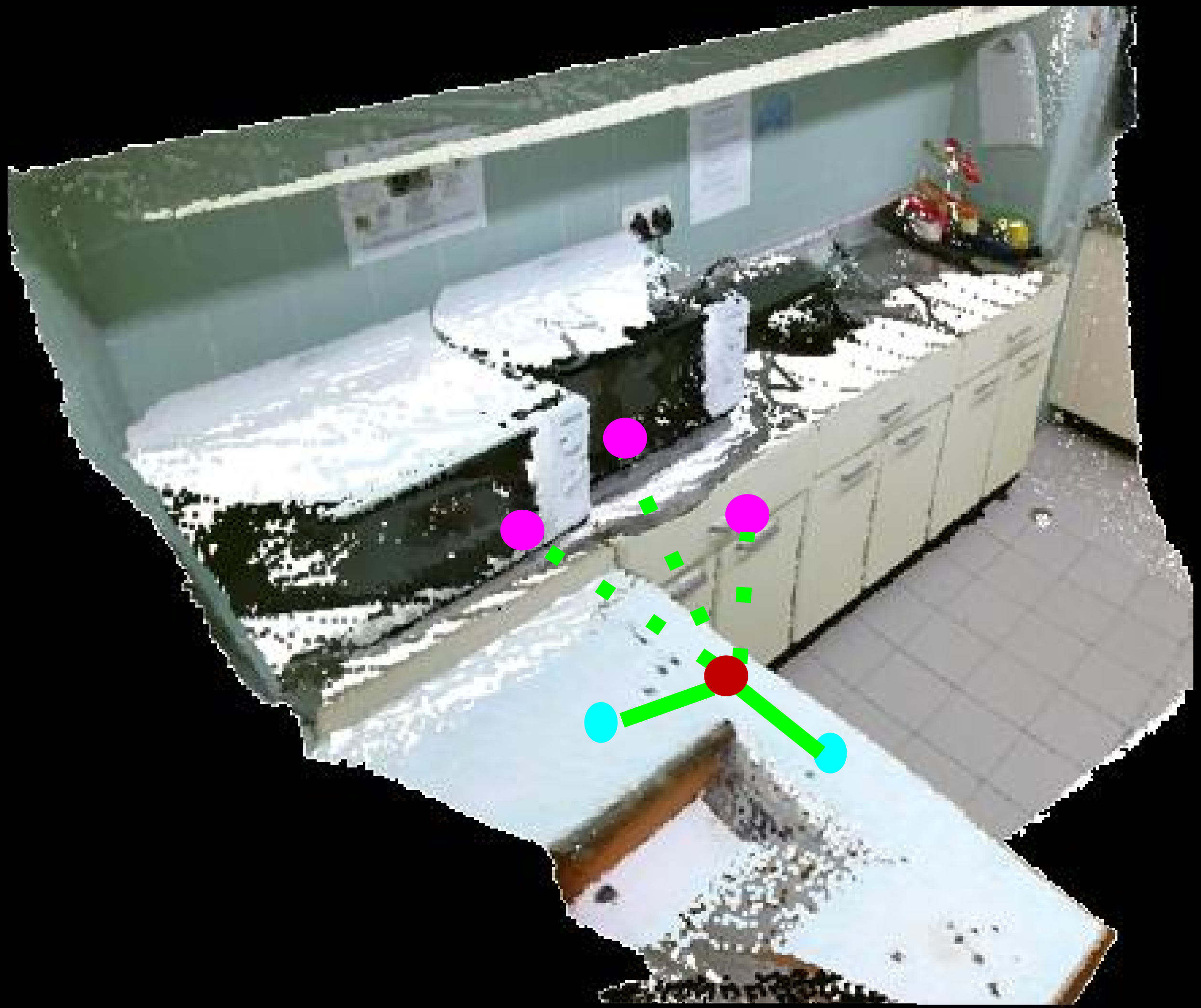
[Long et al. 2015]
[Eigen and Fergus 2015]
[Li et al. 2016]
...

Motivation



2D Image

Motivation

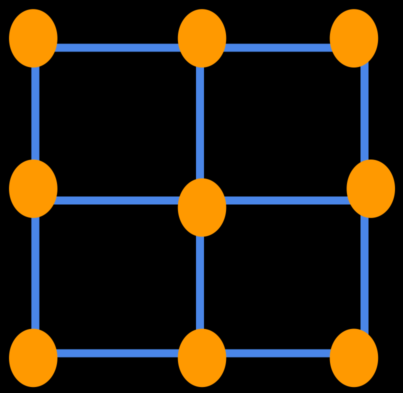


3D Point Cloud

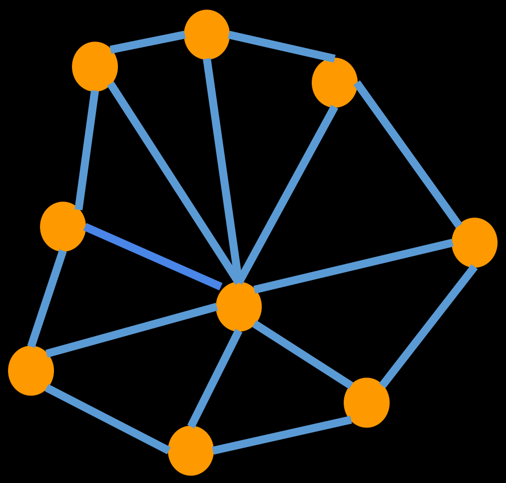
- Accurate Context
- 3D Geometry

Challenge

3D Point Cloud is non-uniformly structured data



Grid structured data



Non-uniform structured data

Solutions

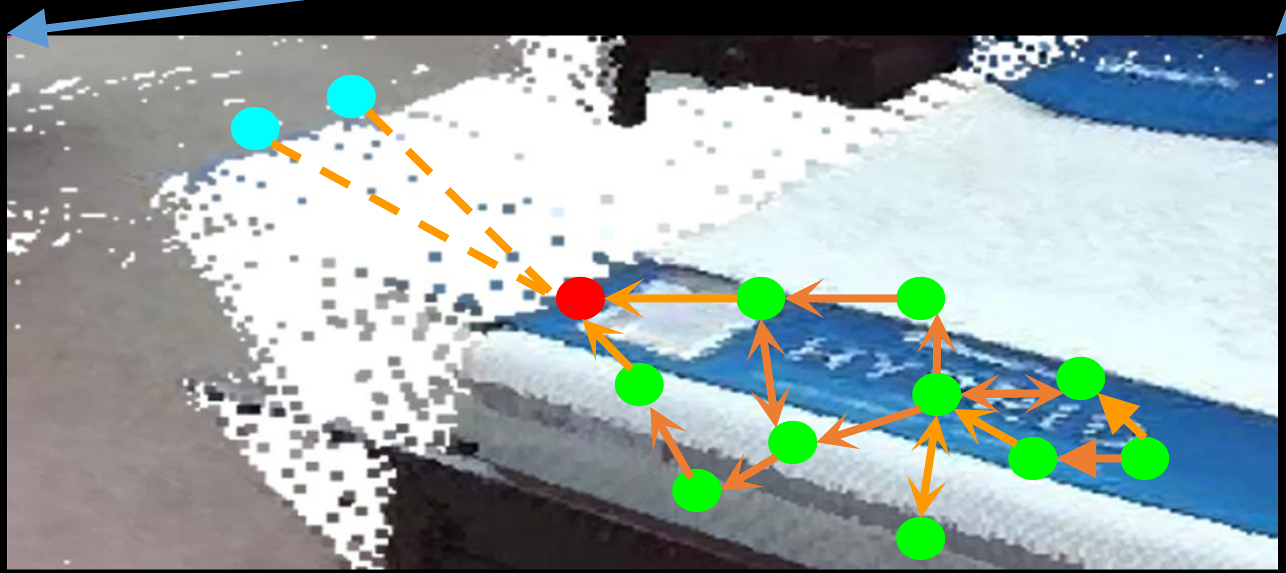
3D Graph Neural Networks

3D Graph Neural Networks (3DGNN)

- Graph Construction
- Propagation Model

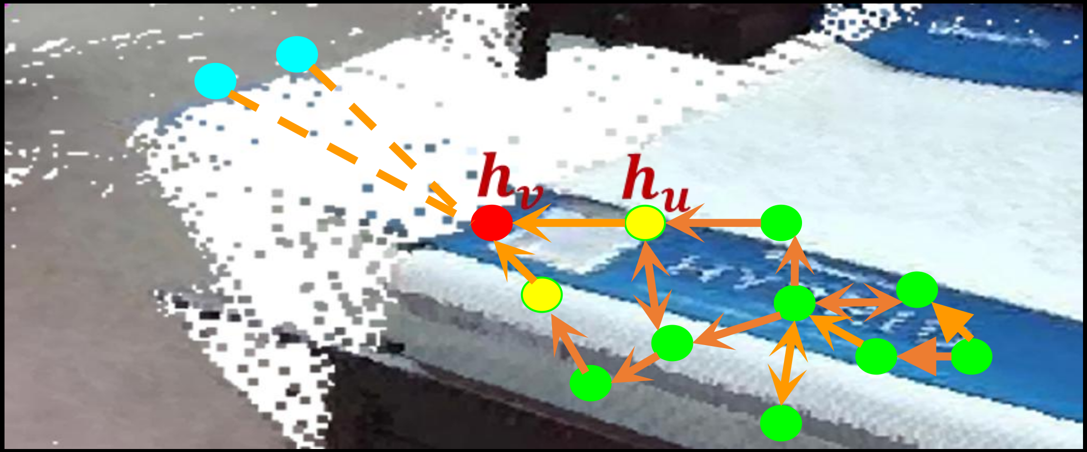
3DGNN: Graph Construction

- Graph Construction
  - ➢ Node: each point in the cloud
  - ➢ Edge: directed edge



3DGNN: Propagation Model

- Node v is associated with a state vector: h_v.
- The state vector is recurrently updated based on its history state and messages from its neighbor .



3DGNN: Propagation Model

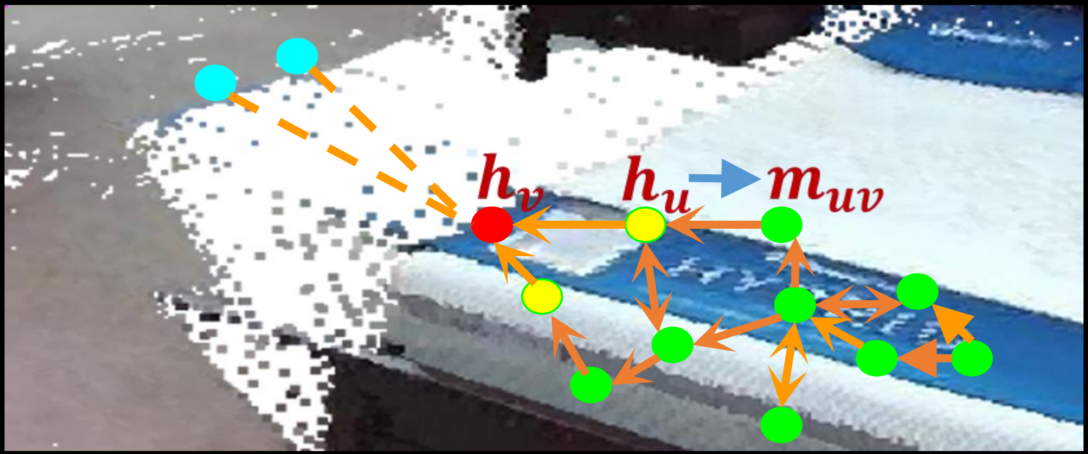
- In each block, we perform:
  - Message computation
  - Message aggregation
  - Node state update

3DGNN: Message Computation

- Step 1: For each directed edge (u, v), the message is:

$$m_{uv} = g_{uv}(h_u)$$

g_{uv}: a message function (an identity mapping in our case).

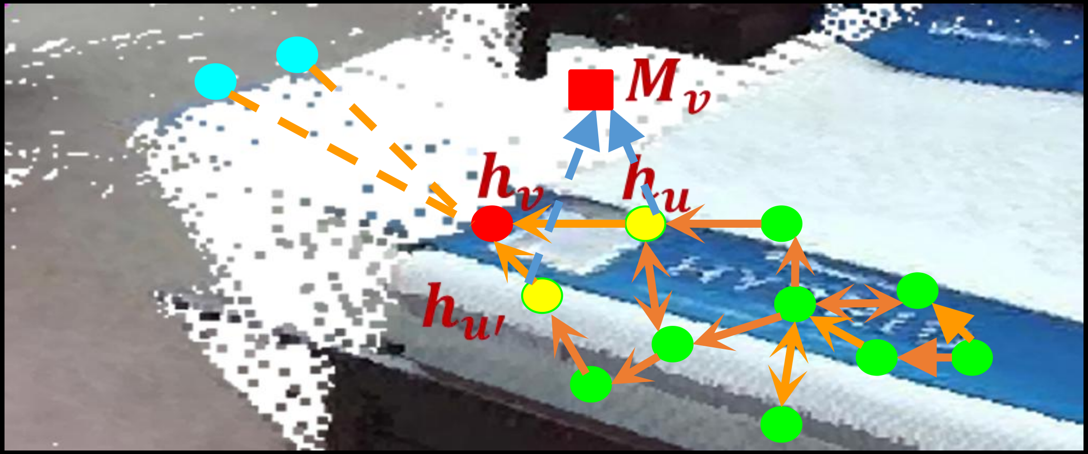


3DGNN: Message Aggregation

- Step 2: Each node v aggregates messages from its neighbors Ω_v:

$$M_v = q\{m_{uv} | u \in \Omega_v\}$$

q: aggregation function (average in our case)

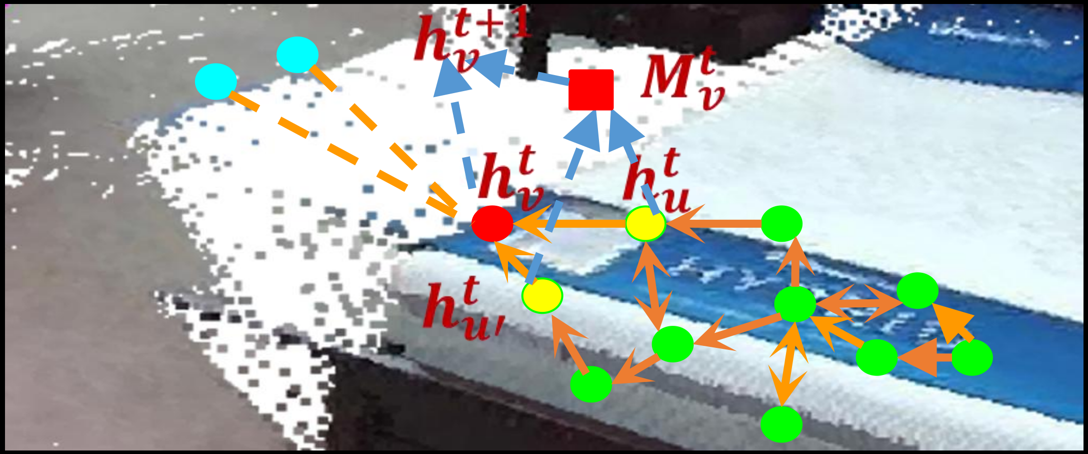


3DGNN: Node State Update

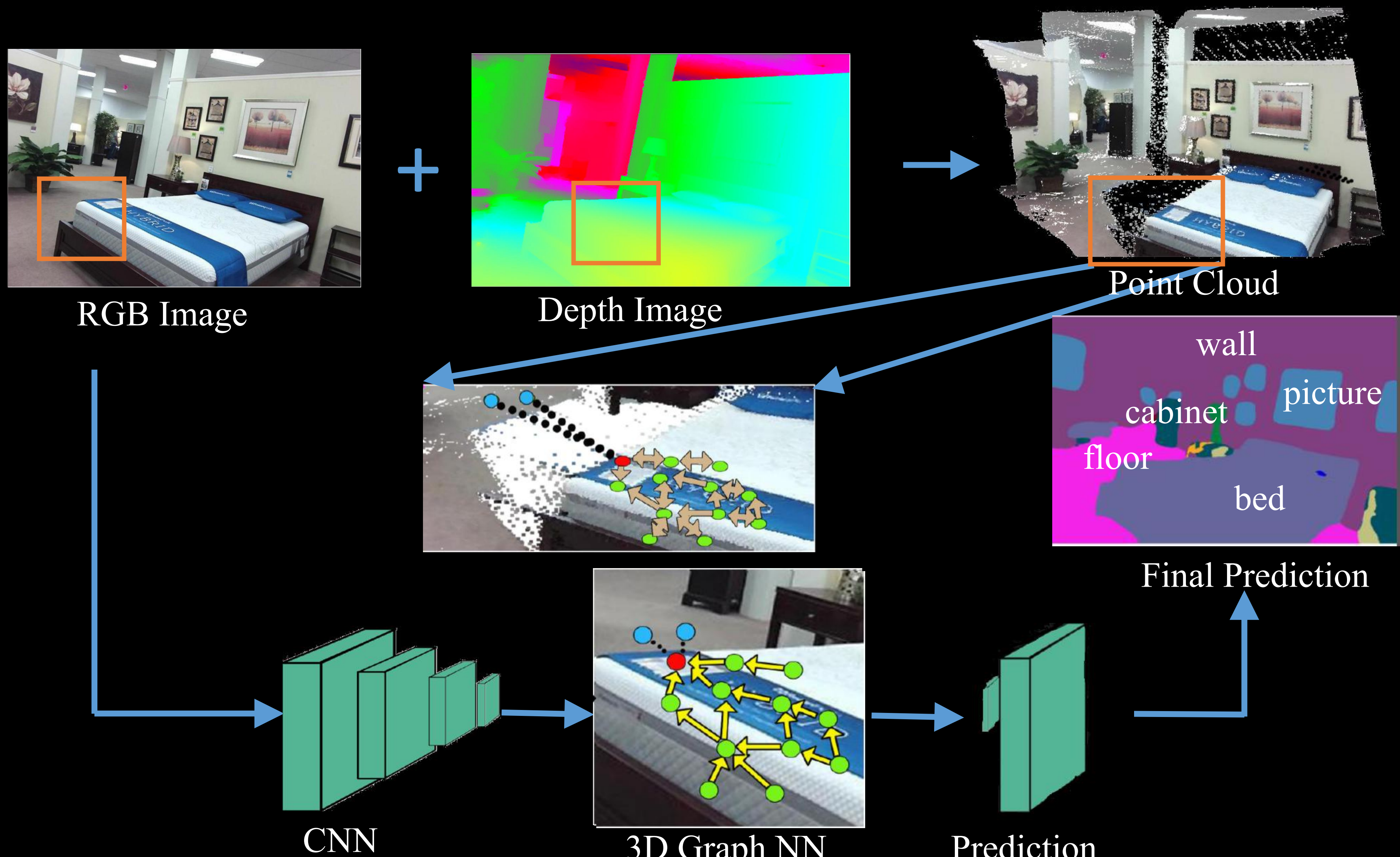
- Step 3: Update state of node v:

$$h_v^{t+1} = f(M_v^t, h_v^t)$$

f: update function (MLP with ReLu in our case)



Model Overview



Generalization of Existing Models

❖ 3D GNN to PointNet

- Graph structure → fully connected with self loop
- Message function → identity function
- Aggregation function → max pooling

❖ 3D GNN to RNN/LSTM

- Graph structure → chain structure

Generalization of Existing Models

❖ MRF Inference

$$Q(y_i) = \frac{1}{Z_i}\{-\phi_u(y_i) - \sum_{j\in\Omega_i} E_{Q(y_i)}[\phi_p(y_j, y_i)]\}$$

❖ Graphical Neural Network Interpretation

- Message function → $E_{Q(y_i)}[\phi_p(y_j, y_i)]$
- Aggregation function → Negation summation
- Update function → Softmax

Quantitative Experimental Results

Evaluation

[Gupta et al. 2014]	28.6	35.1
[Eigen an Fergus 2015]	34.1	45.1
3DGNN (VGG)	**41.7**	**55.4**

NYUD2 test set under 40 classes setting

Evaluation

[Song et al. 2014]	-	36.3
[Li et al. 2016]	-	48.1
	42.3	54.6
3DGNN (ResNet101)	**45.9**	**57.0**

SUN-RGBD test set

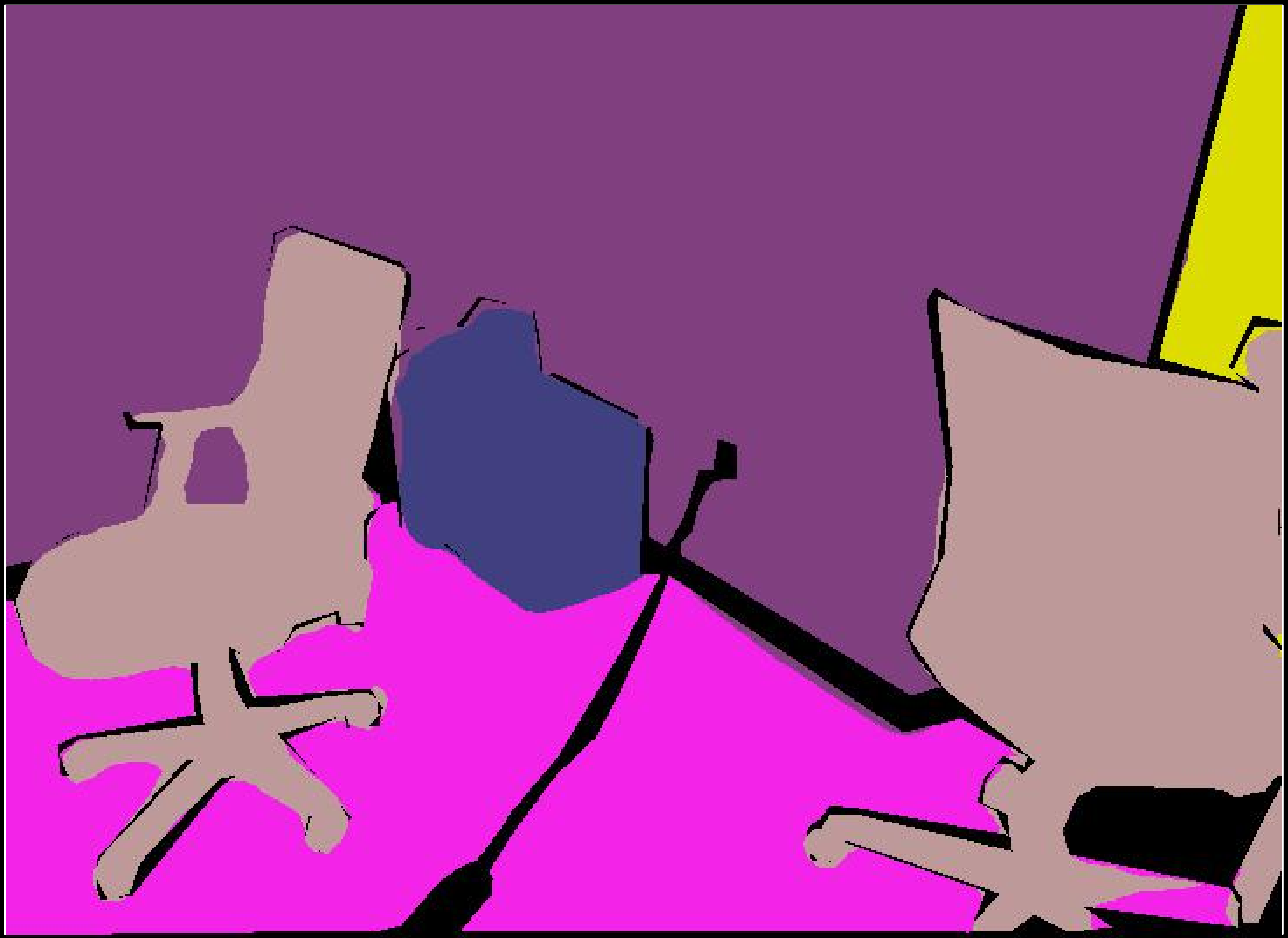
Ablation Study

- Effectiveness of 3D Graph Neural Network

NYUD2-40	Unary CNN	37.1	51.0
	3D GNN	**39.9**	**54.0**
	2D GNN	38.9	50.3
	3D GNN	**40.2**	**52.5**

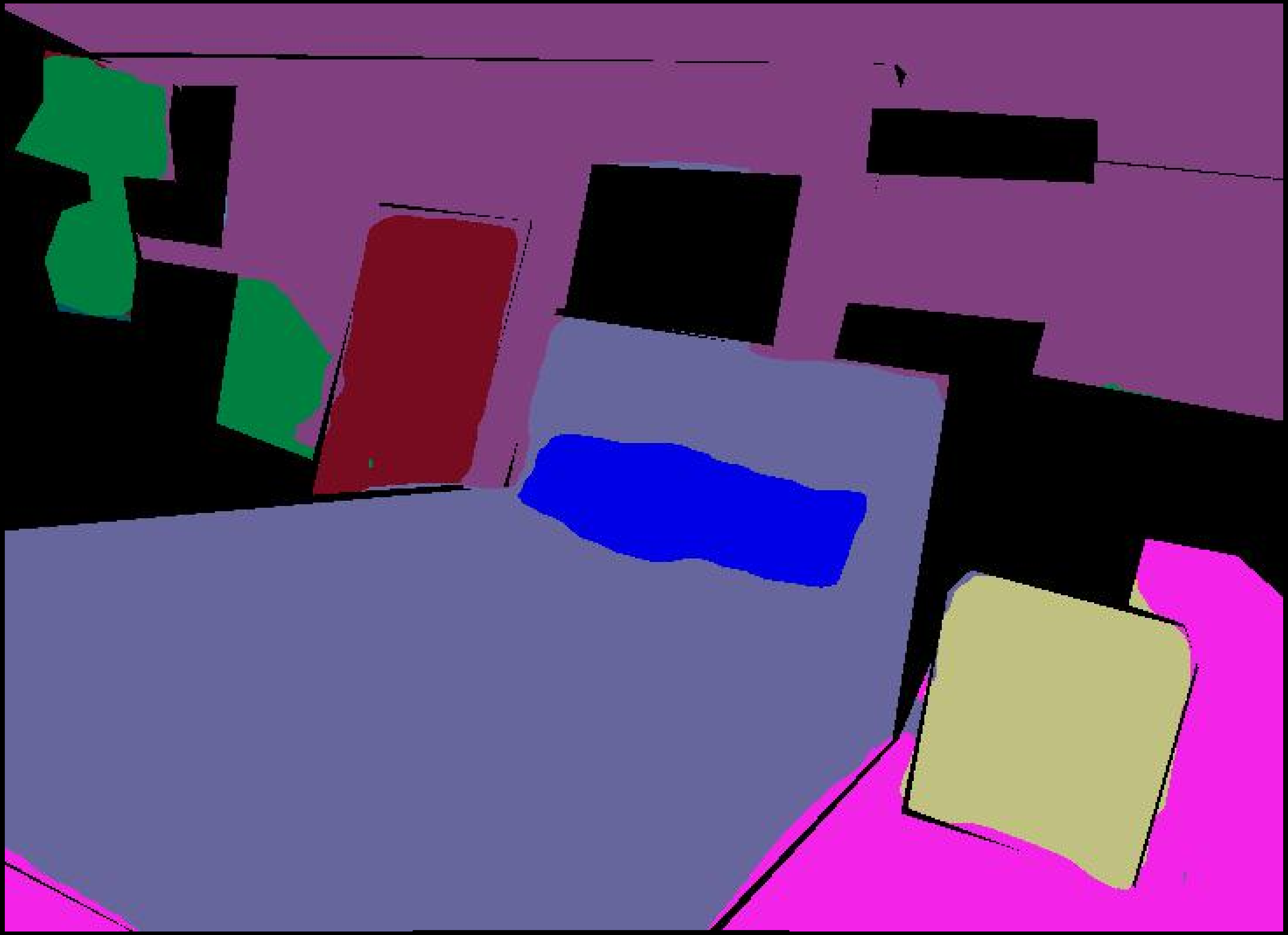
Qualitative Experimental Results

Visual Results



3DGNN

Visual Results



3DGNN

Visual Results

3DGNN

Conclusion

- **3DGNN** is a general framework for modeling RGBD data.

- **3DGNN** achieves state-of-the-art performance on RGBD semantic segmentation.

Thank You!

Image 2D (Resnet-101) 3DGNN