3DCNN-DQN-RNN:

A Deep Reinforcement Learning Framework for Semantic Parsing of Large-scale 3D Point Clouds

Fangyu Liu*, Shuaipeng Li*, Liqiang Zhang, Chenghu Zhou, Rongtian Ye, Yuebin Wang, Jiwen Lu

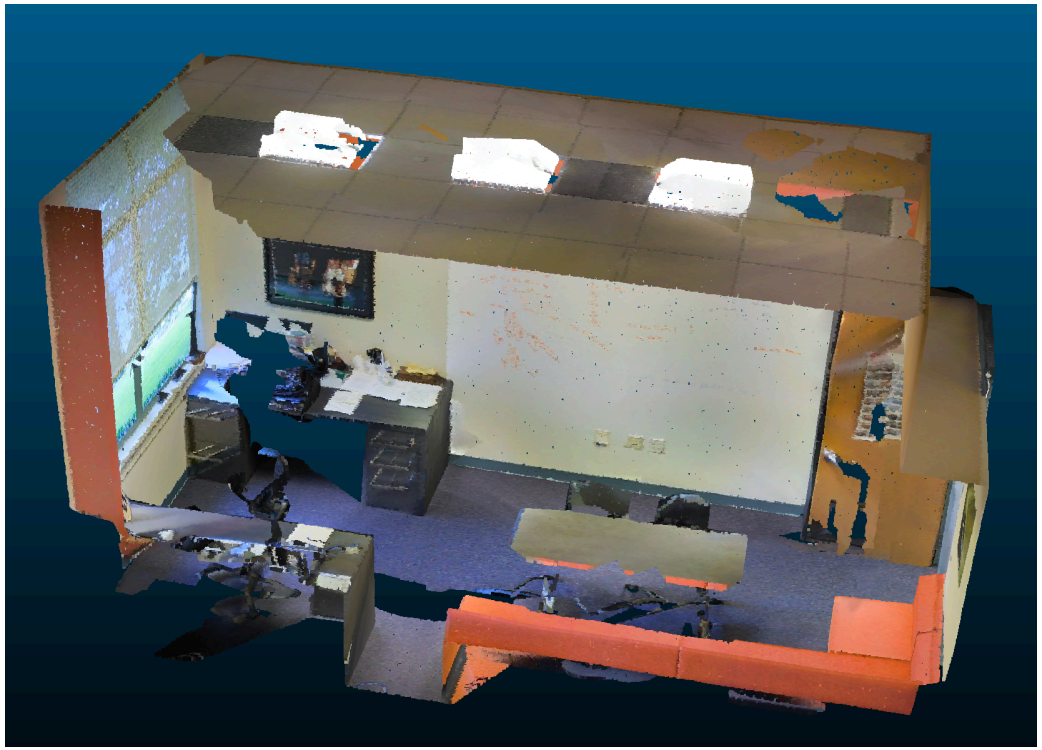
(*indicates equal contribution)

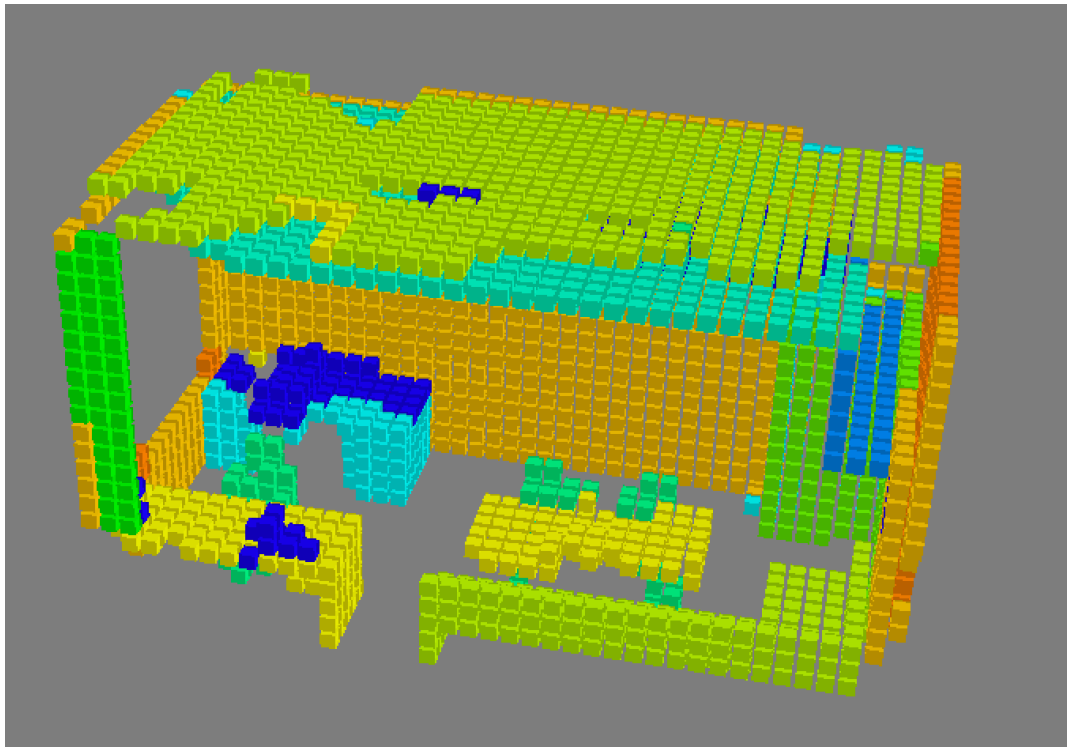
Overview

- Motivation: How we come up with RL+Vision
- Our framework
- Rethink

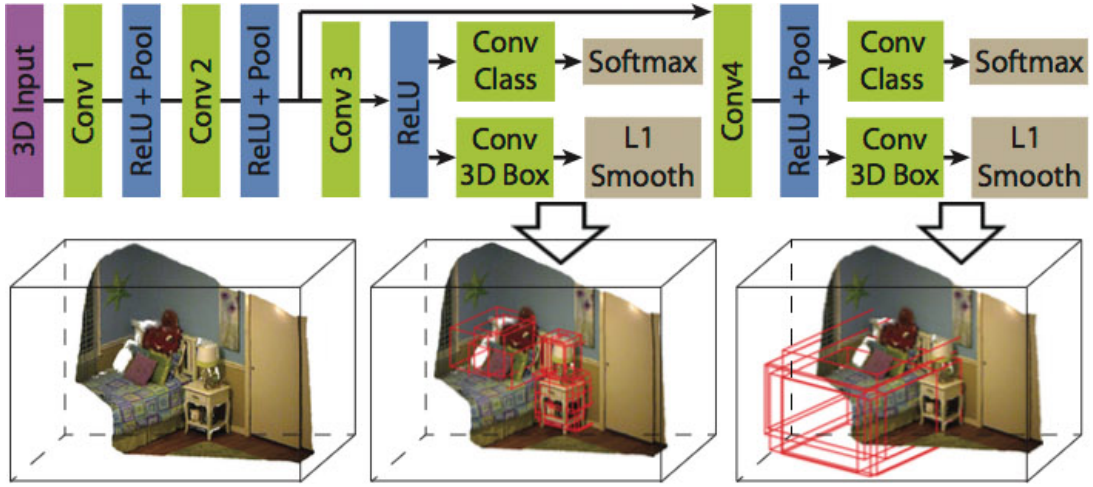
Motivation: why RL+Vision?

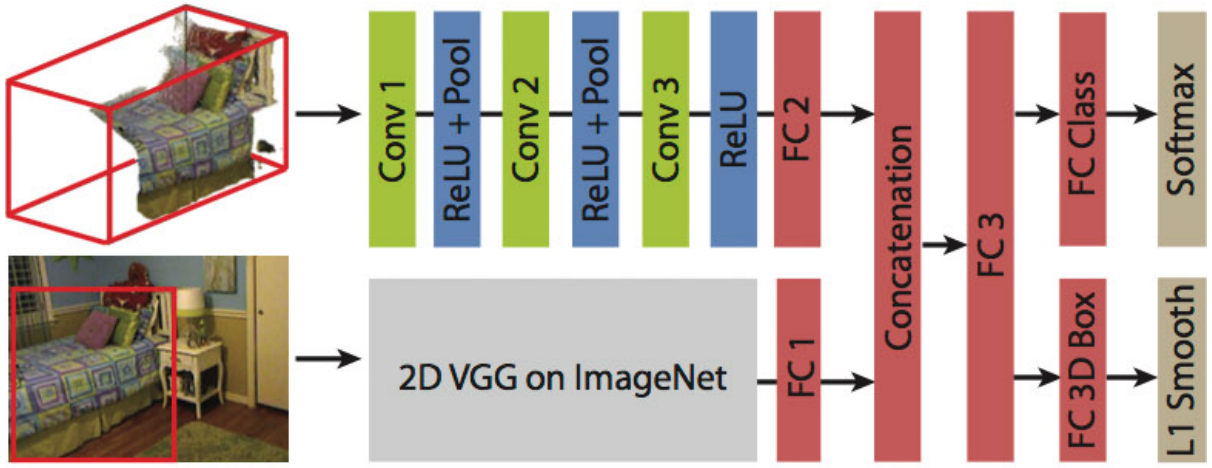
Voxelization





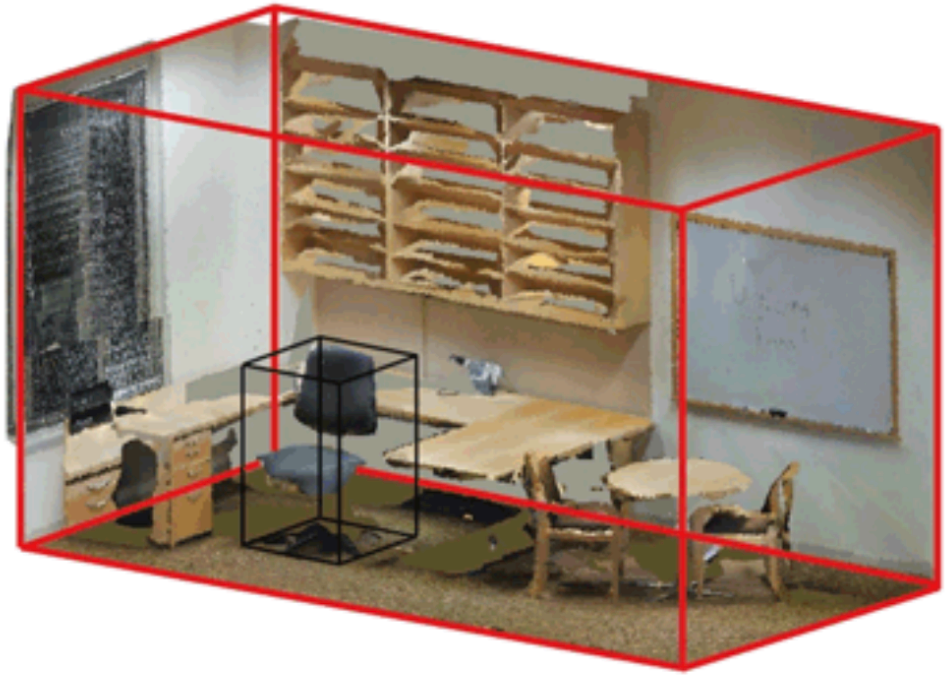
Sliding Window

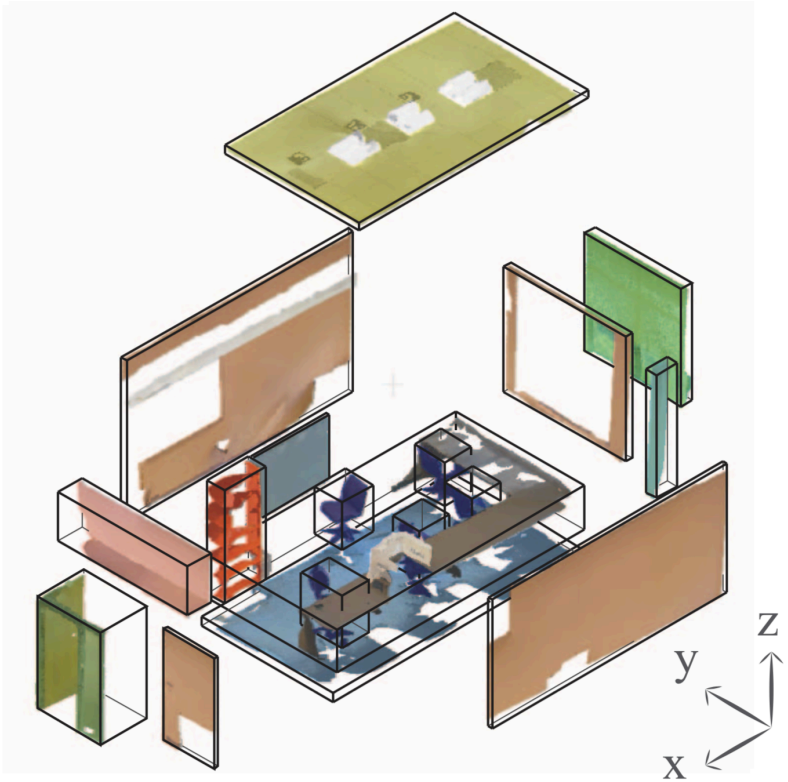




[Song et al. 2016]

Sliding Window





[Armeni et al. 2016]

DQN: Deep Q-Learning Network

Reward

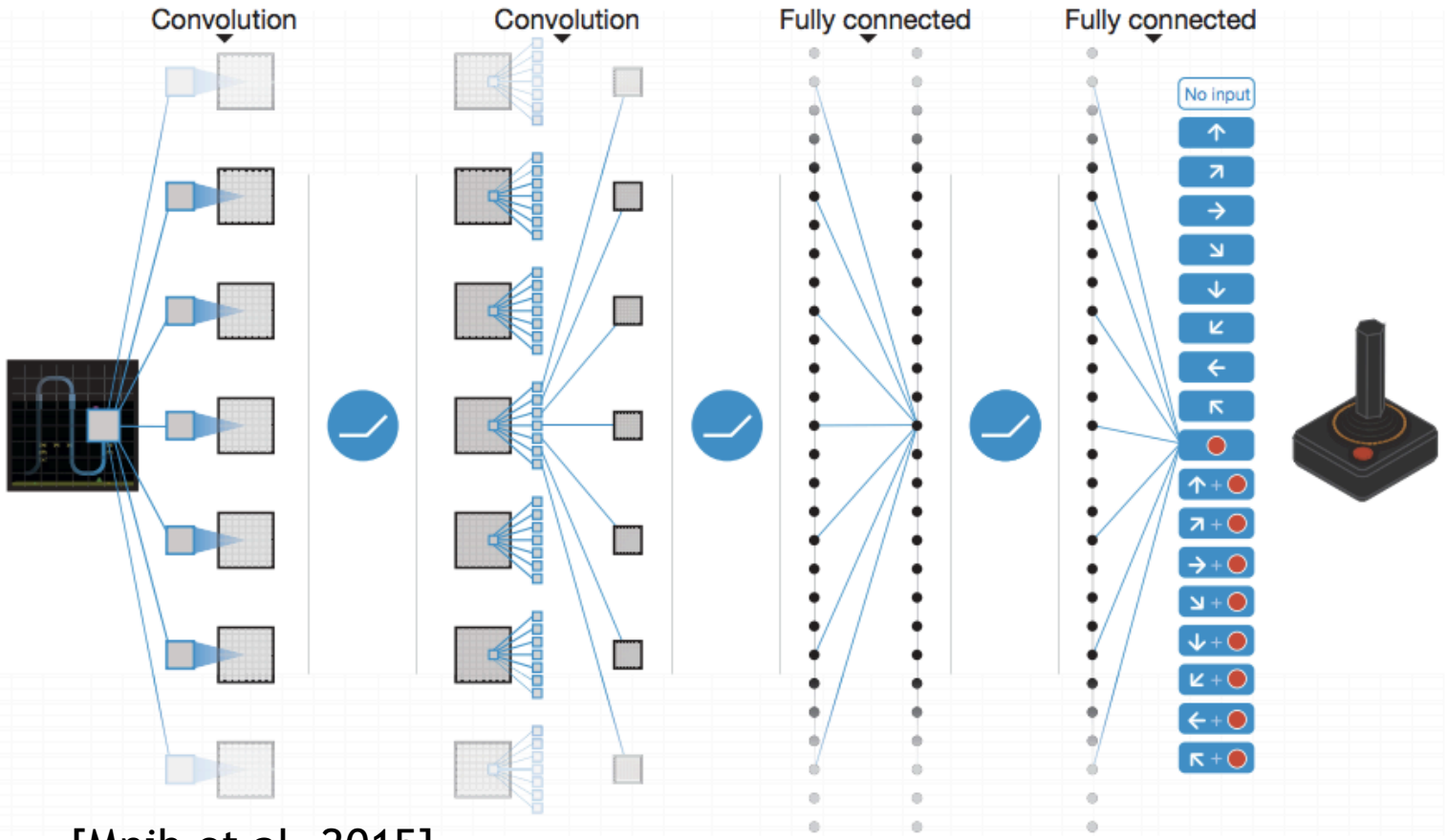
State

Action

[Mnih et al. 2015]

Based on current **state** and potential **reward**, we choose an **action** that may maximize the future winning chance.

DQN: Deep Q-Learning Network

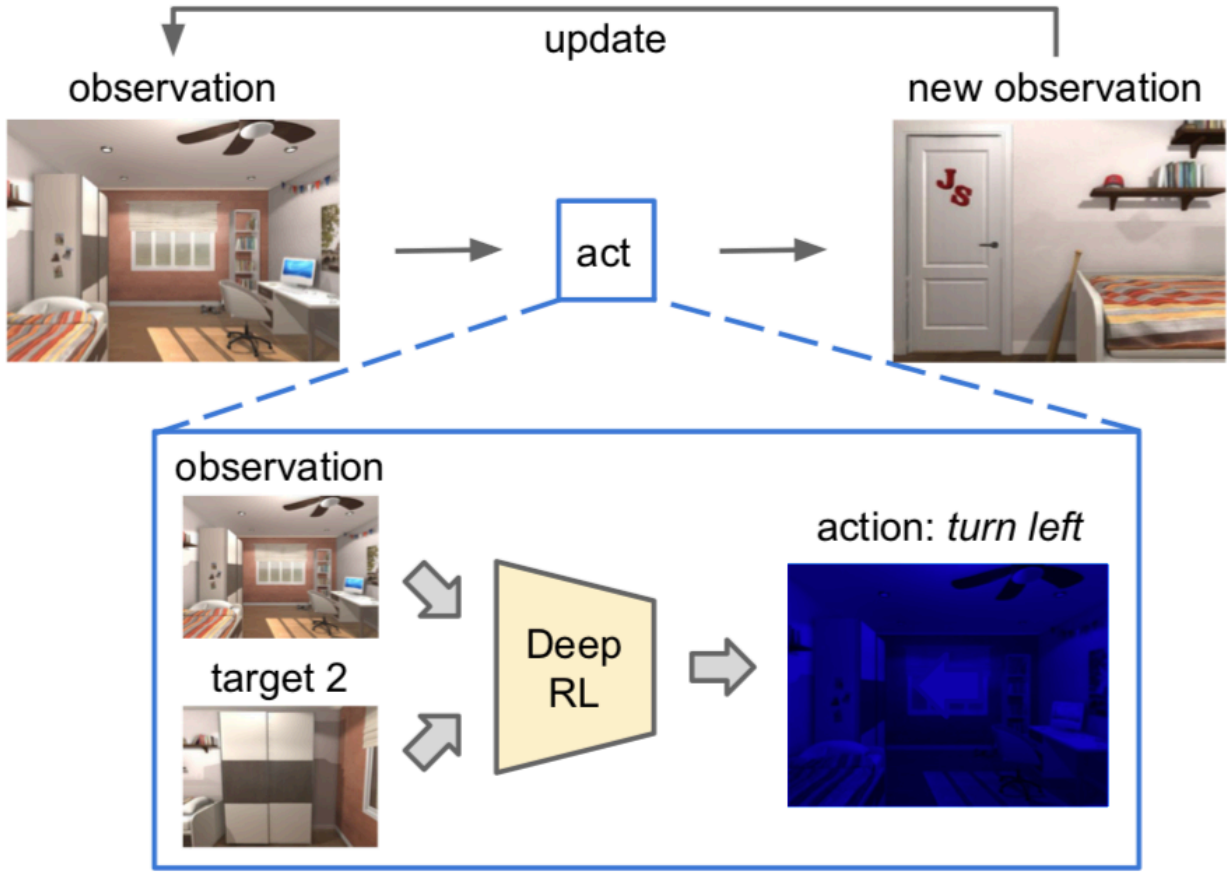


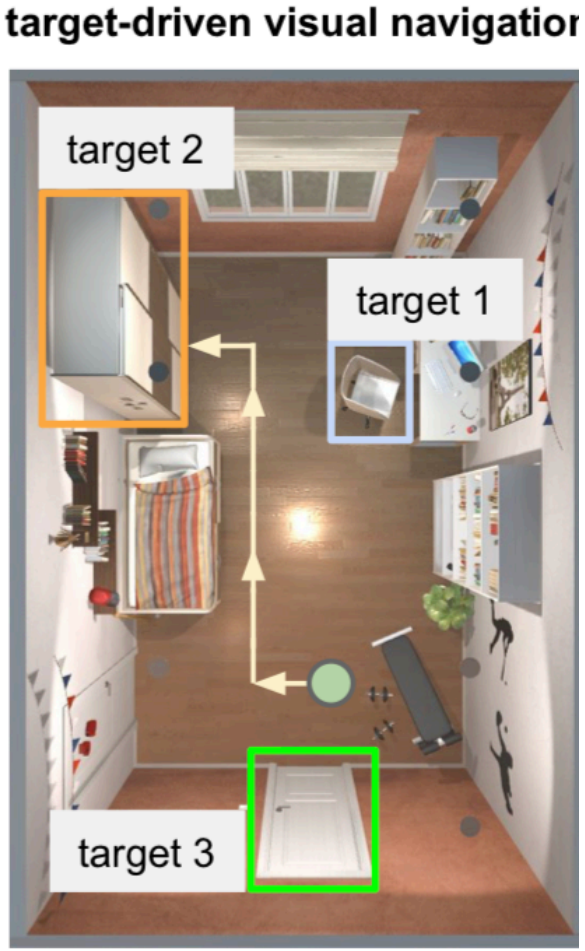
[Mnih et al. 2015]

- Eye (CNN) -> Brain (layers in the middle) -> Action (Output)

“Dueling Network" [Schaul et al. 2015]
“Prioritized Replay” [Wang et al. 2015]

RL for Target-driven Navigation





[Zhu et al. 2017]

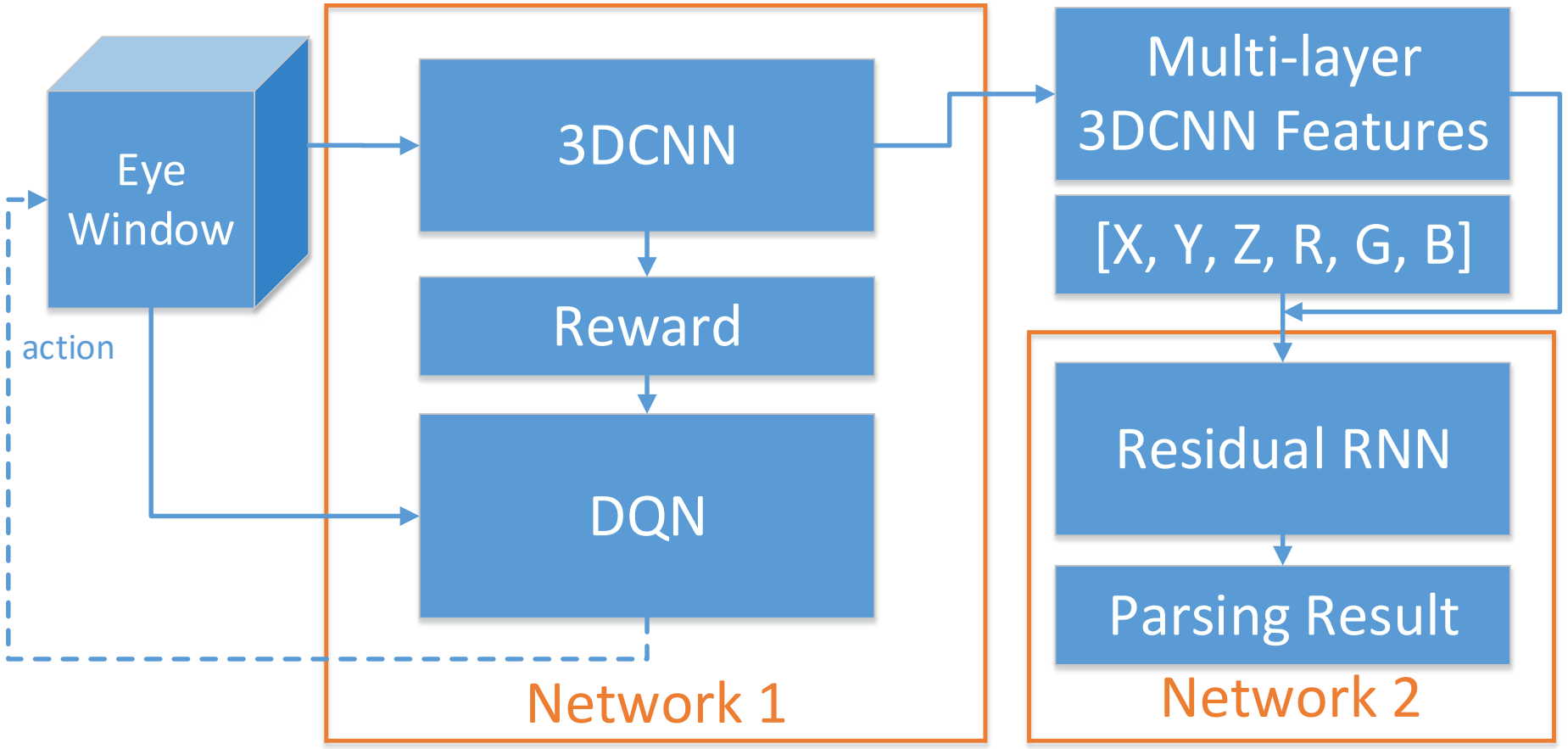
Vision + Reinforcement Learning

- We expect much of the future progress in vision to come from systems that are **trained end-to-end** and combine **ConvNet**s with **RNN**s using **reinforcement learning to decide where to look**.

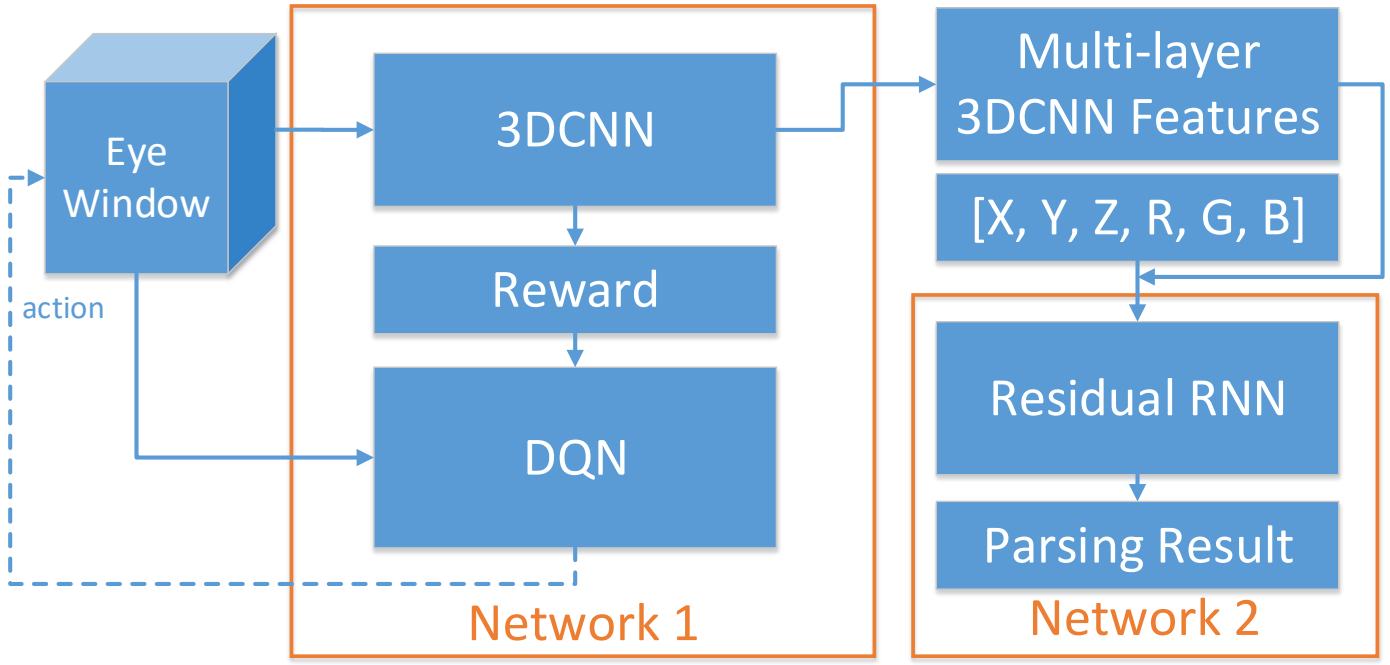
- Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. Nature, 2015.

Our Framework

Pipeline Overview

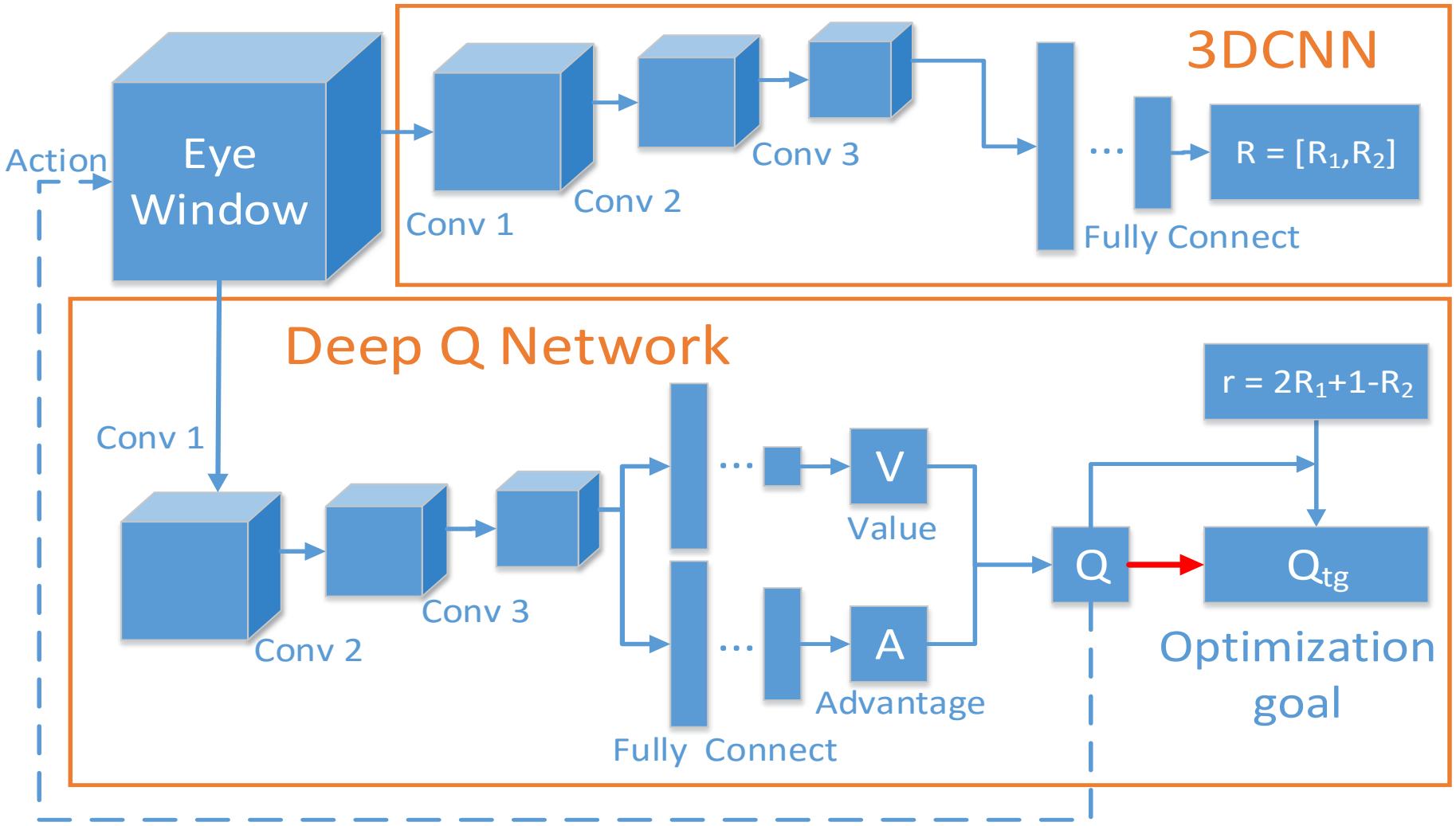


Pipeline Overview



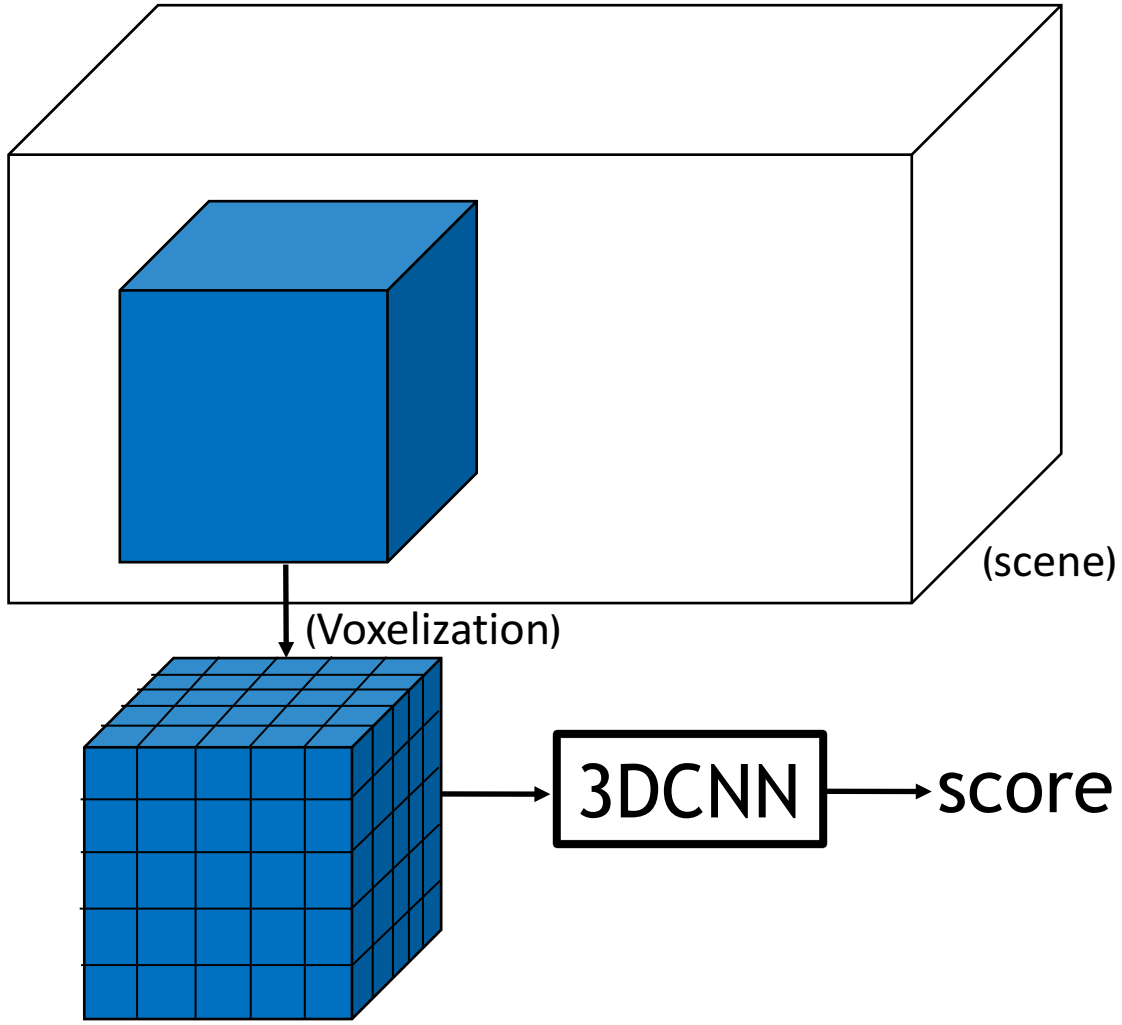
- Eye Window - An agent/robot
- CNN - Evaluation function & Feature Extractor
- DQN - Control System
- RNN - Deep Classifier

Network 1 for Detection and Localization



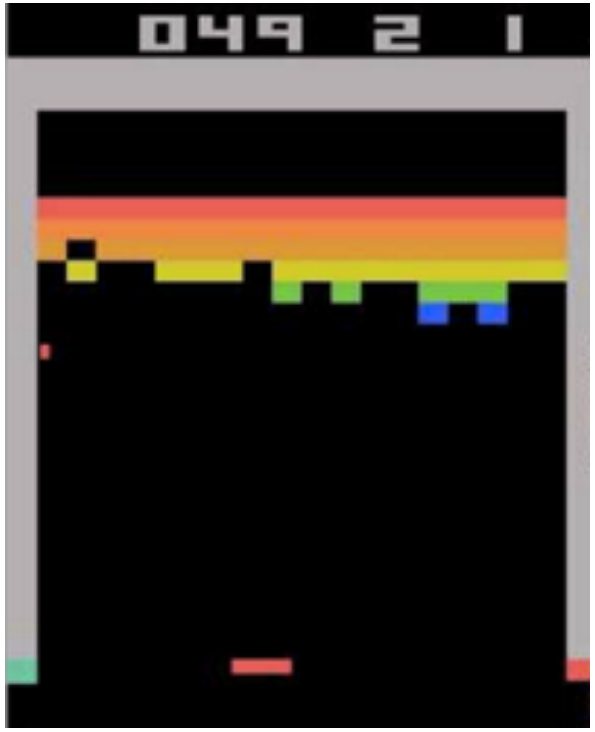
3D CNN

- Details of 3D CNN:
  L1: input(batch size, 40, 40, 40, 3)
  L2: BatchNorm(ReLU(conv3d(8, 5, 3)))
  L3: BatchNorm(ReLU(conv3d(14,4,2)))
  L4: BatchNorm(ReLU(conv3d(32,3,1)))
  L5: conv3d(512, 1, 1)
  L6: Global Average Pooling
  L7: fc(1024)
  L8: softmax(fc(xn))

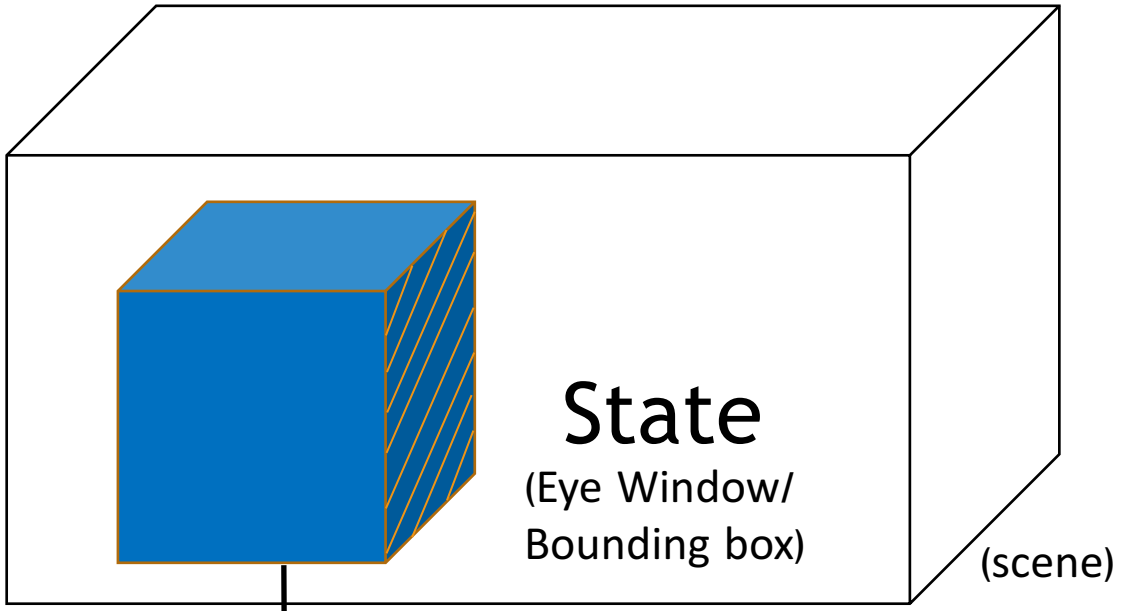


DQN: Deep Q-Learning Network

Reward (score)



State
(screenshot)



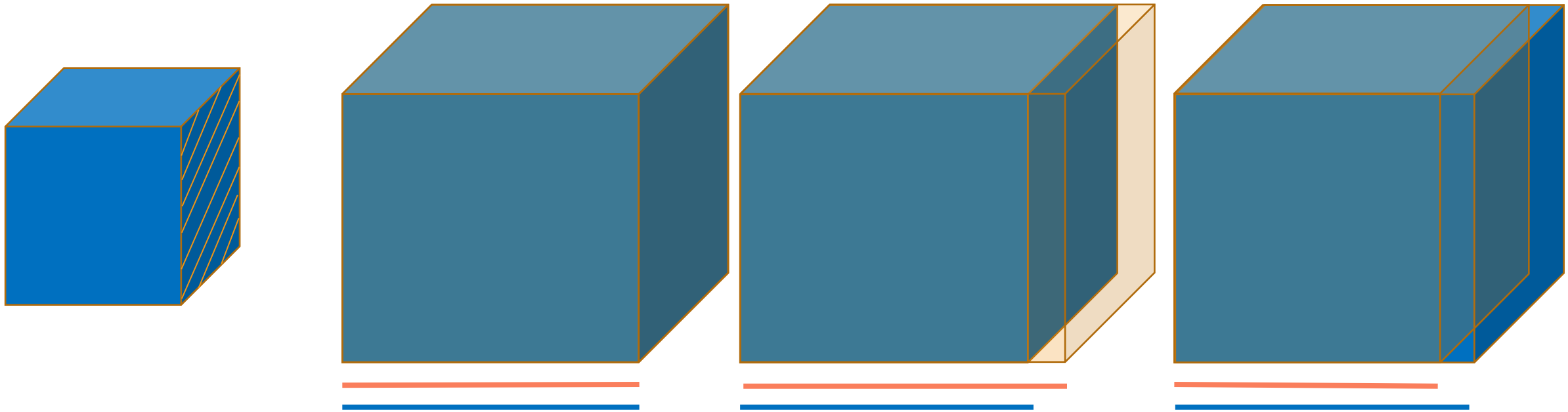
3DCNN (A pre-trained evaluator)

Reward

Action
(left, stay still, right)

DQN: Deep Q-Learning Network

Action (for each side: stay still(0), expand(+1), shrink(-1))



$$a = [p_1, p_2, p_3, p_4, p_5, p_6], p_k \in \{-1,0,1\}, k = 1,2, \dots 6$$

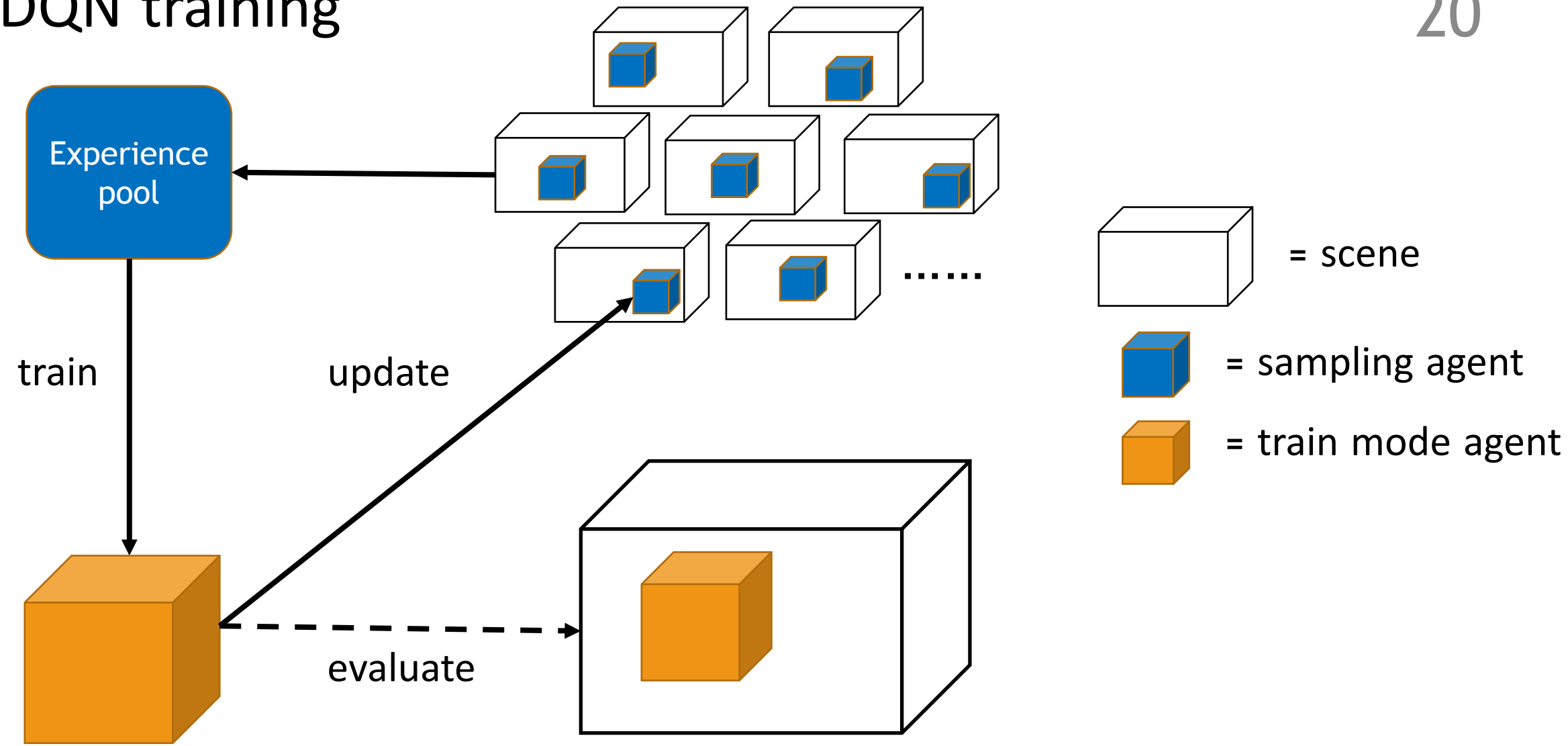
DQN: Deep Q-Learning Network

- n-step simulation
- Simulate k times

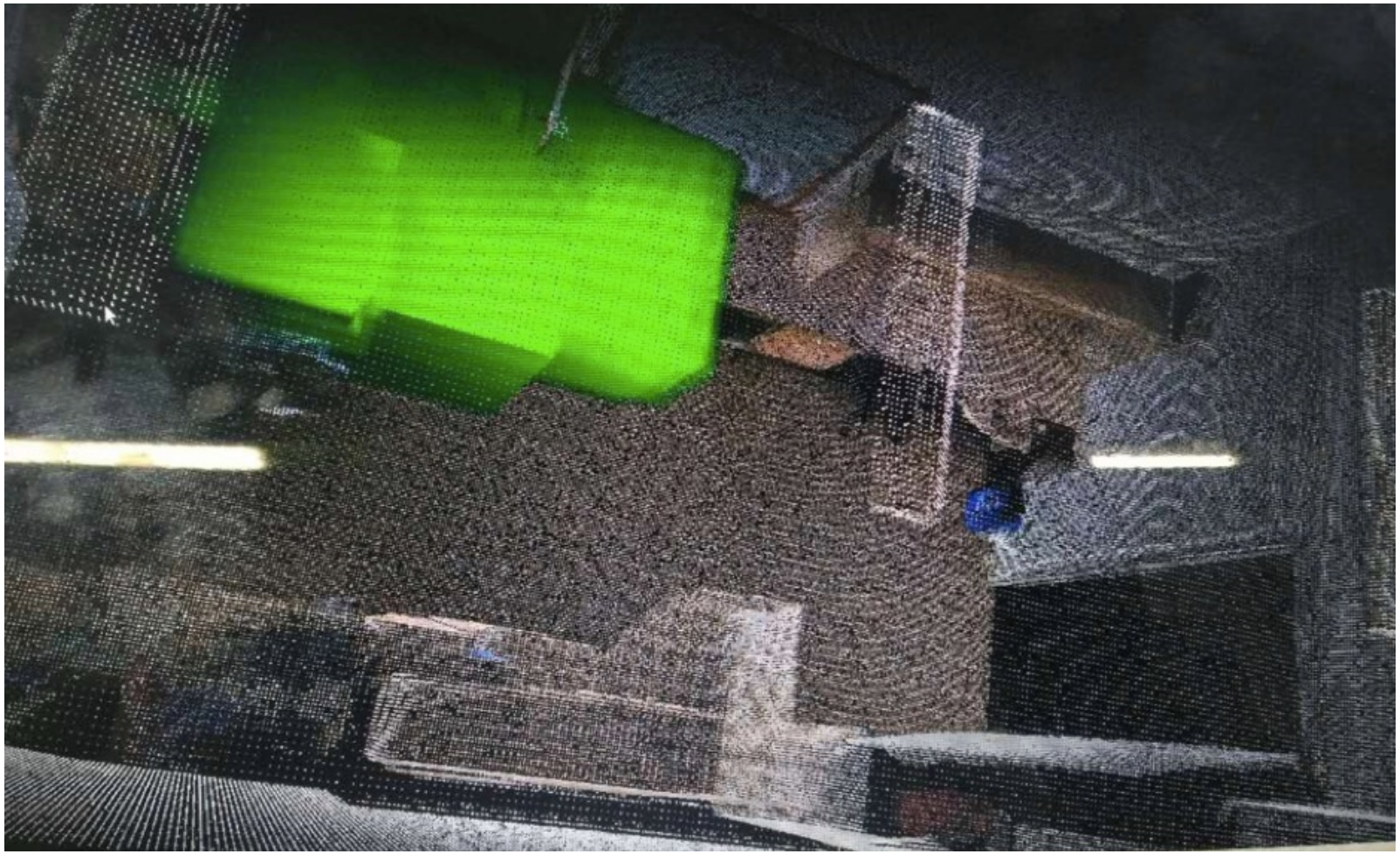
$$Q_{tg} = \tanh(\sum_{t=0}^{N-1} \lambda^t r_t + \lambda^N Q')$$

$$\theta_{T+1} = \theta_T + \lambda(Q_{tg} - Q(s, a; \theta_T))\nabla_{\theta_T} Q(s, a; \theta)$$

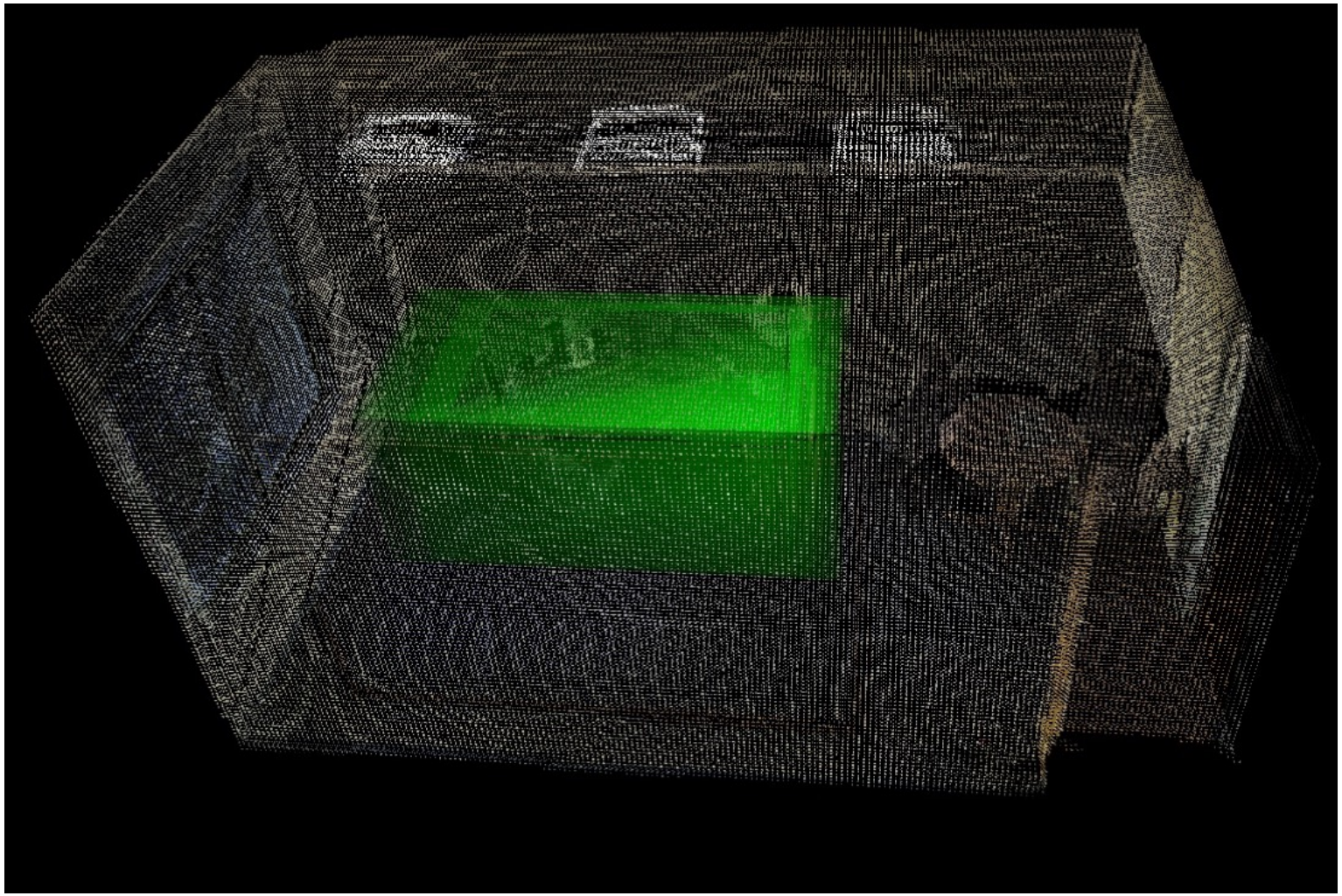
DQN training

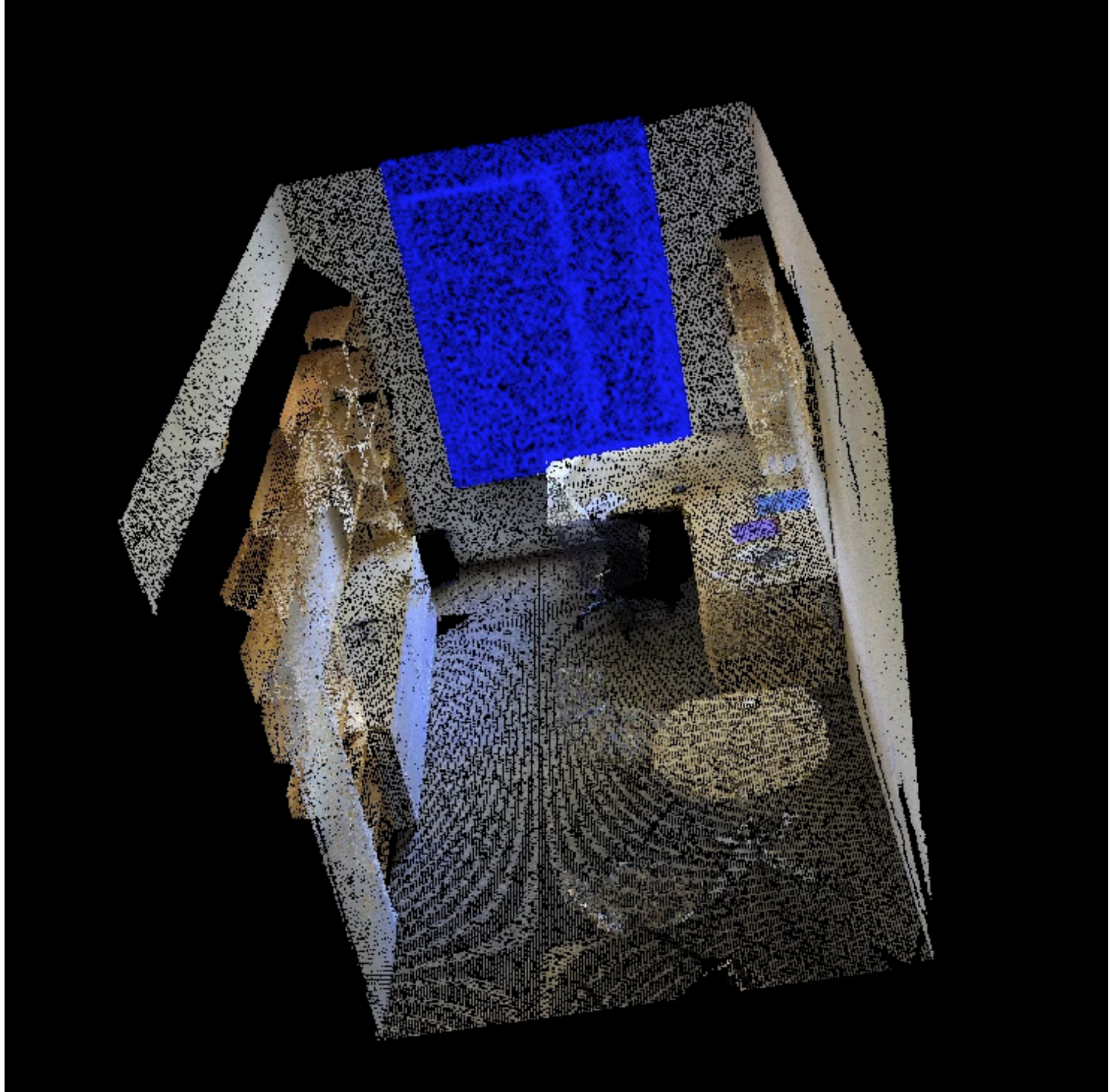


Searching tables…

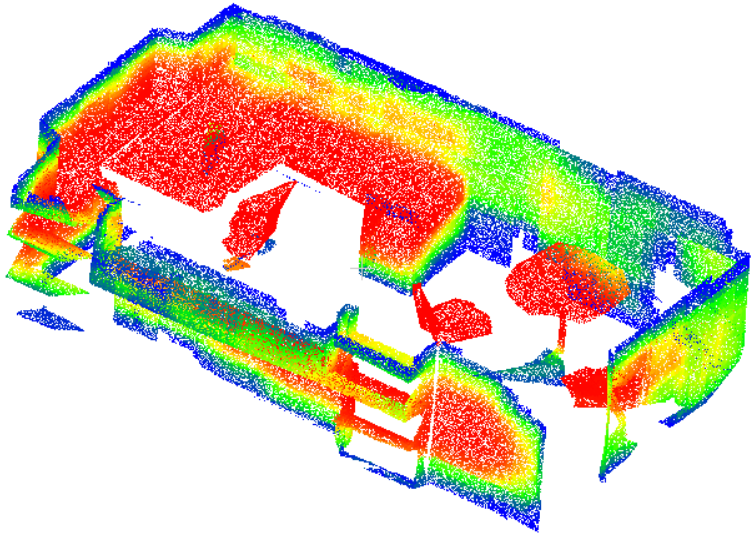


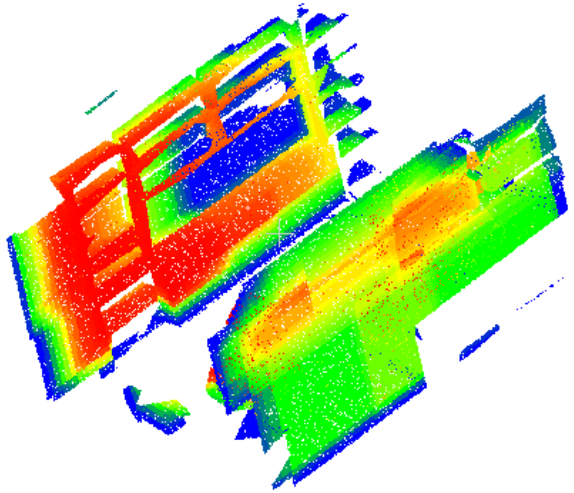
Searching tables…



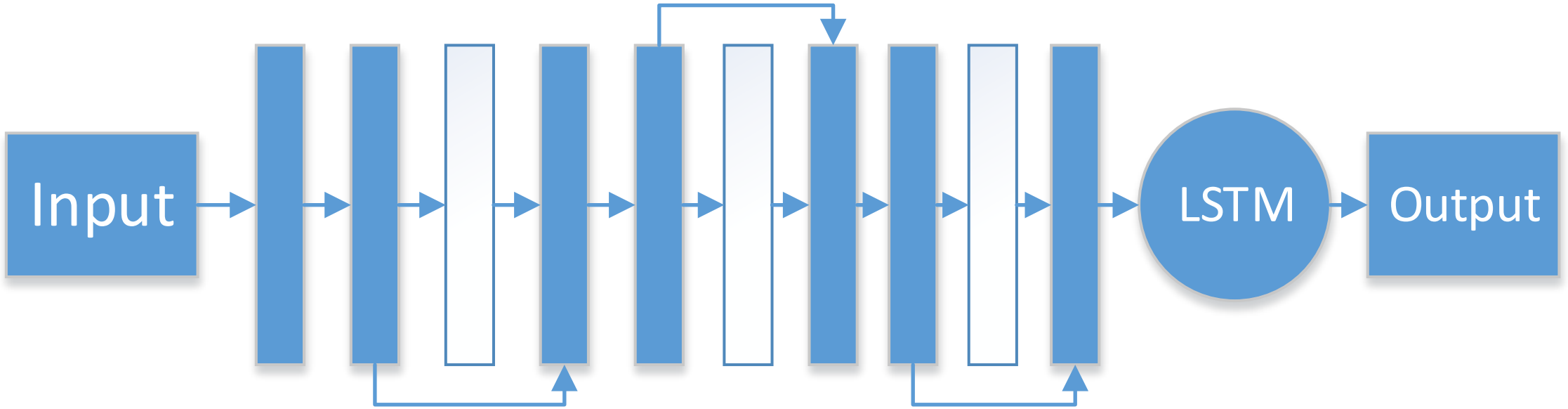


Non-Maximum Suppression



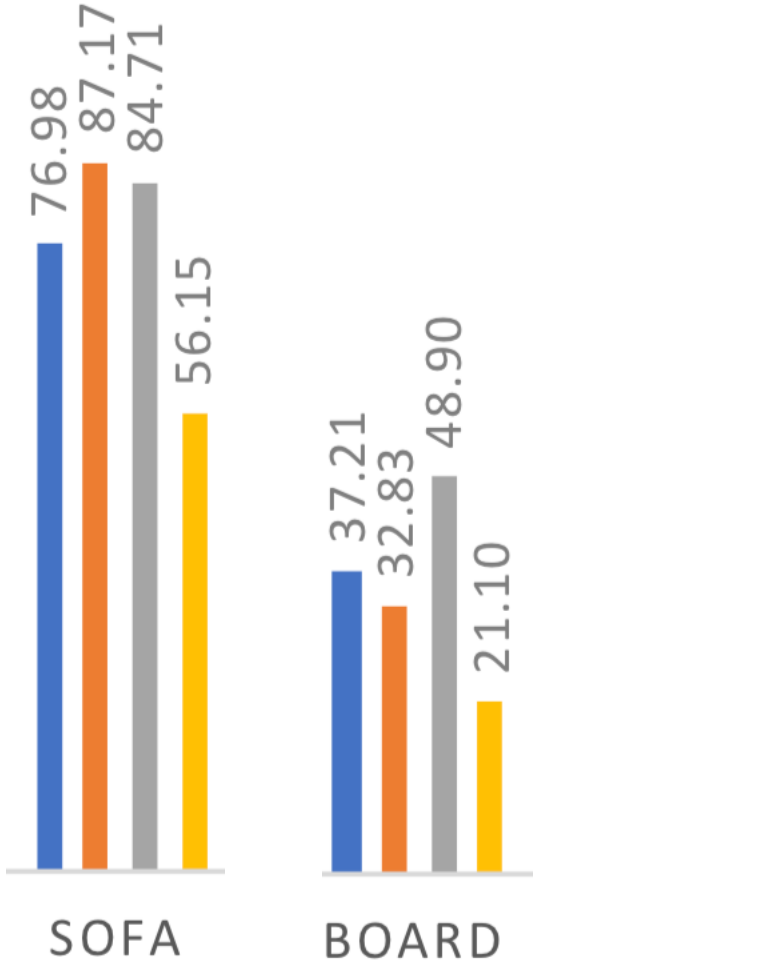


Network 2: Residual RNN for Classification



Network 2: Residual RNN for Classification

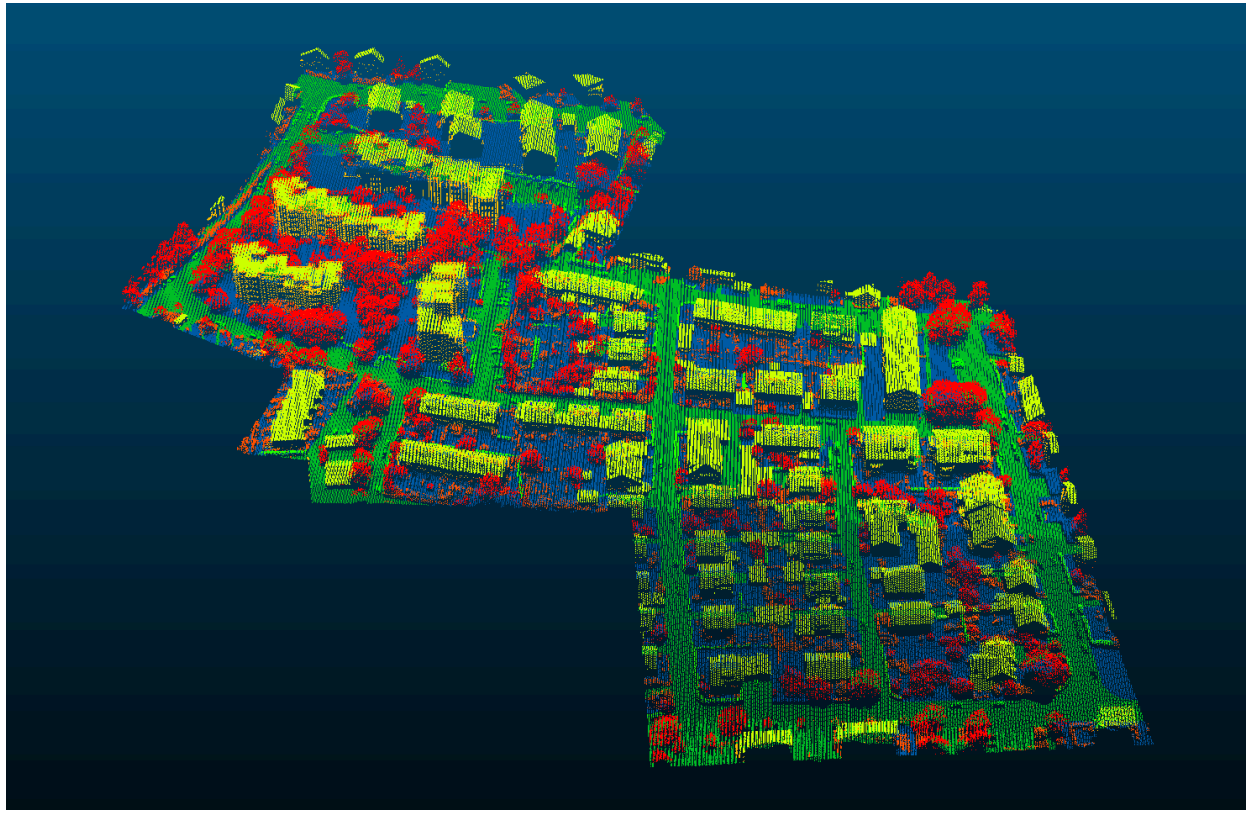
- Improve Precision

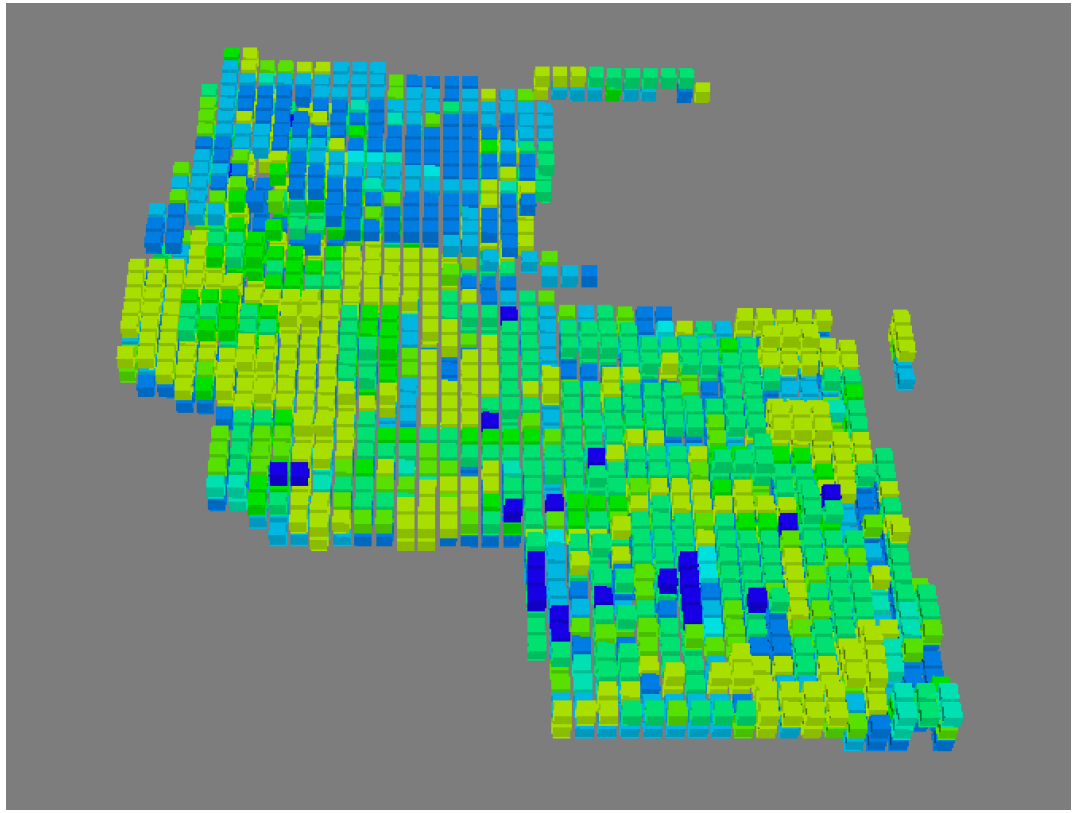


(N1+N2 accuracy,N1 accuracy,N1+N2 precision,N1 precision)

Rethink

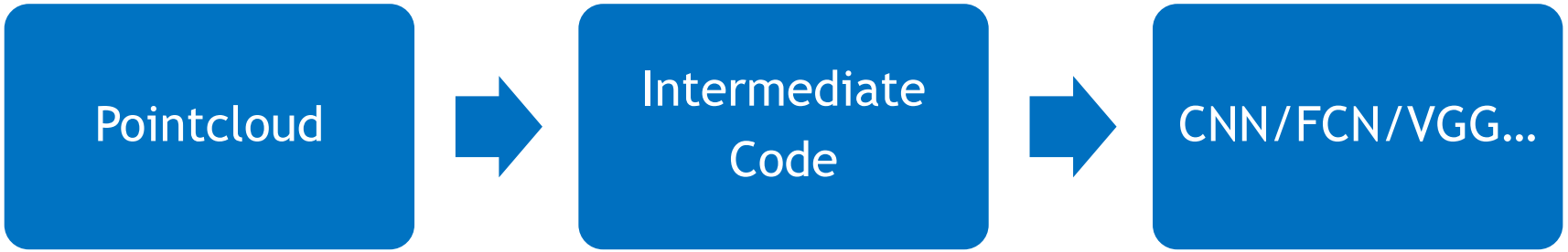
Downside of Voxelization and Using bbox in Large-scale Scenes





- bbox only works on regular objects
- Scale varies in large-scale scenes

Intermediate Code



- Voxelization and Bounding box (not that good)
- Multi-view, PointNet, etc.

Thank you!

fangyu.liu@uwaterloo.ca