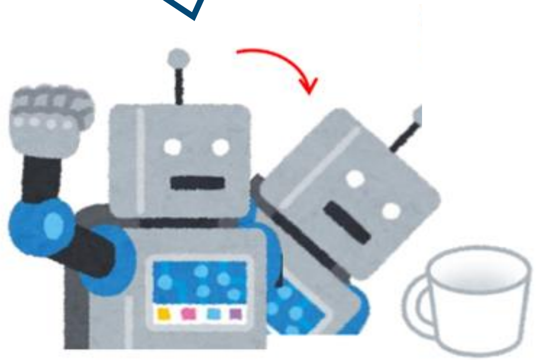
Research Progress

2022.04.01

Guan Yunyi

Make agent learn how to **select next views actively** to increase recognition accuracy

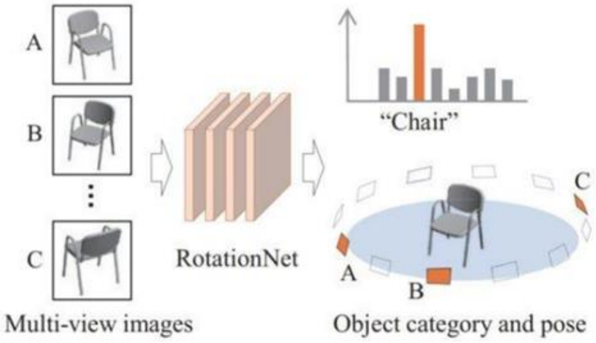
Active Object Recognition with Reinforcement Learning



Key Words: 3D recognition, Next-Best-View (NBV), RL

Main Aim

design a **viewpoint selecting policy** for multi-view based 3D recognition



Definition about “active” in AOR

- **viewpoint selection** around a considered object

  -> reliable classification results with reduced number of views

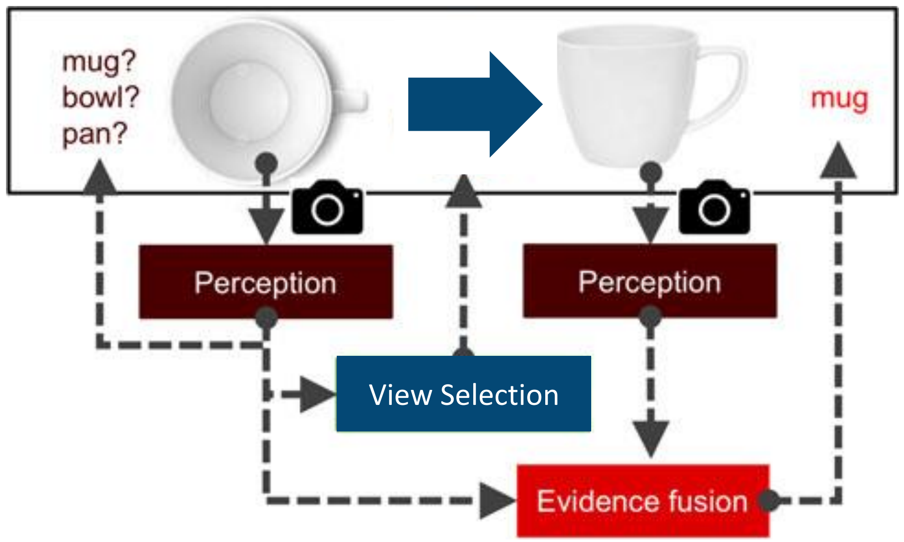
- No need to differentiate between moving the object and camera, only consider about the **relative movement**

  -> assume to there is a perfectly tracked object from the start

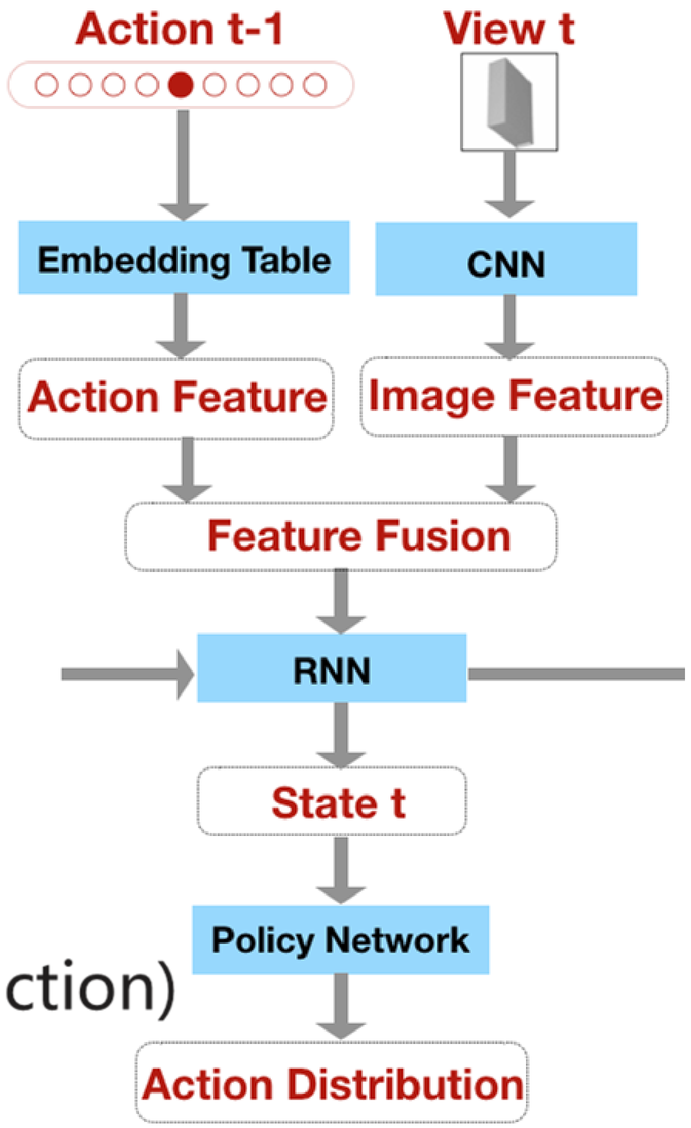
- 2 kinds of viewpoint selection:

- without RL
- with RL

Standard framework of AOR – 3 parts



AOR with action:

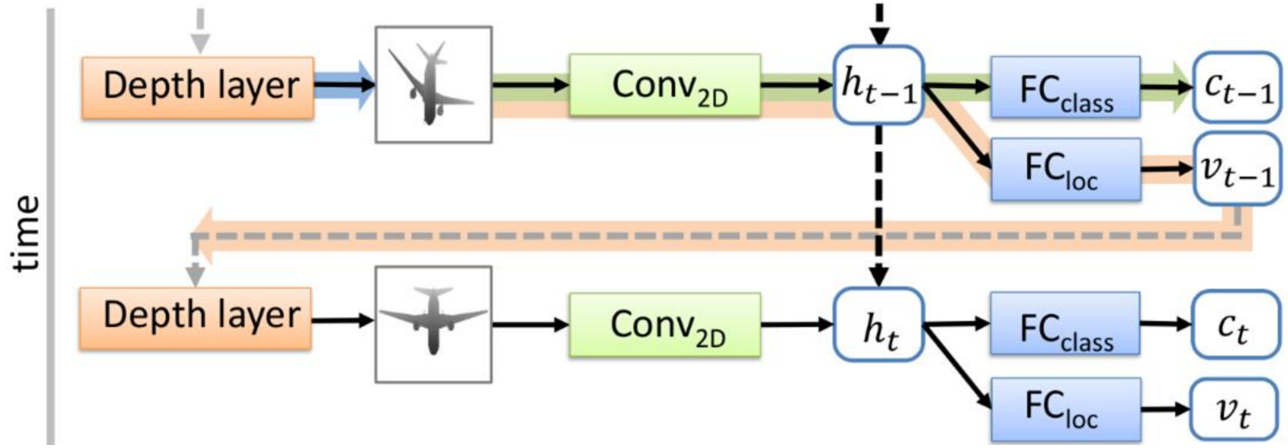


- **Perception**: extract visual features and recognize
- **View selection**: select new view with viewing history
- **Evidence fusion**:

- aggregate visual and action features (only for RL with action)
- aggregate viewing history ($t \ -> \ 1, \ldots, t-1$)

AOR without RL- 3DRAN

Min L., Yifei S., et al. "Recurrent 3D attentional networks for end-to-end active object recognition."

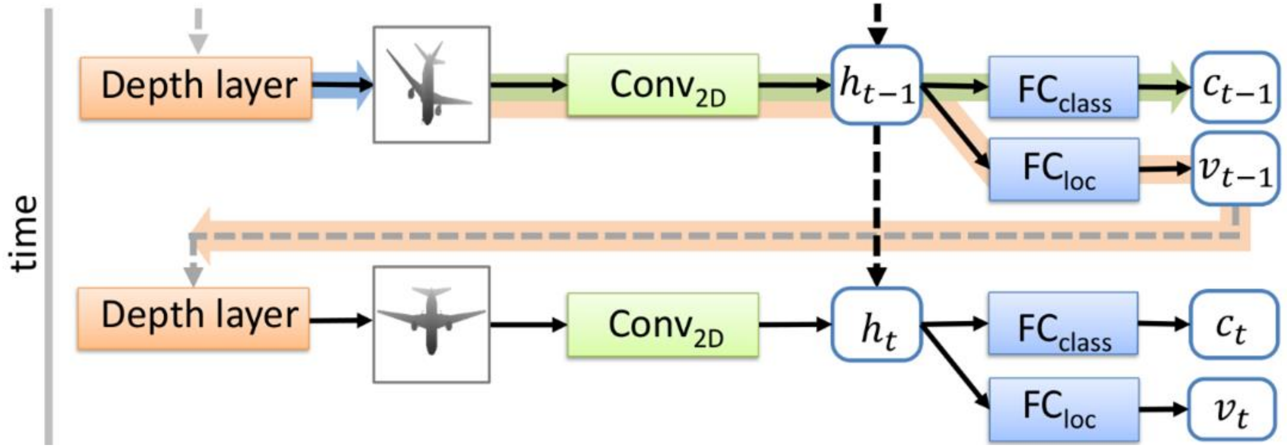


- **Depth layer**: generate 2D images with ray casting algorithm

  -> make the **whole pipeline differentiable** (no need of sampling in RL)
- **Conv2D**: extract image features
- **RNN:** aggregate past view features and store in hidden layer h_{t-1}
- **FCclass:** classify 3D shape c_t
- **FCloc:** regress new view parameters v_t

AOR without RL- 3DRAN

Min Liu, Yifei Shi, et al. "Recurrent 3D attentional networks for end-to-end active object recognition."



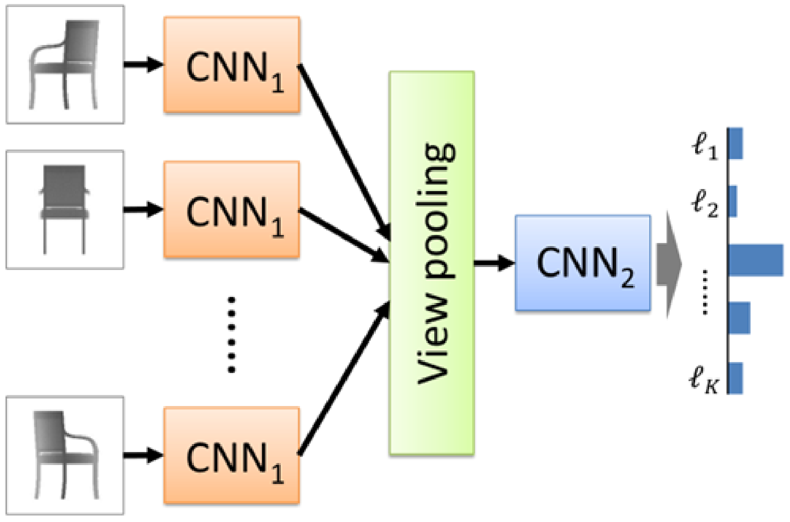
loss is only related to classification result

$$L = -\sum_{c=1}^{k} y_{o,c} \log(p_{o,c})$$

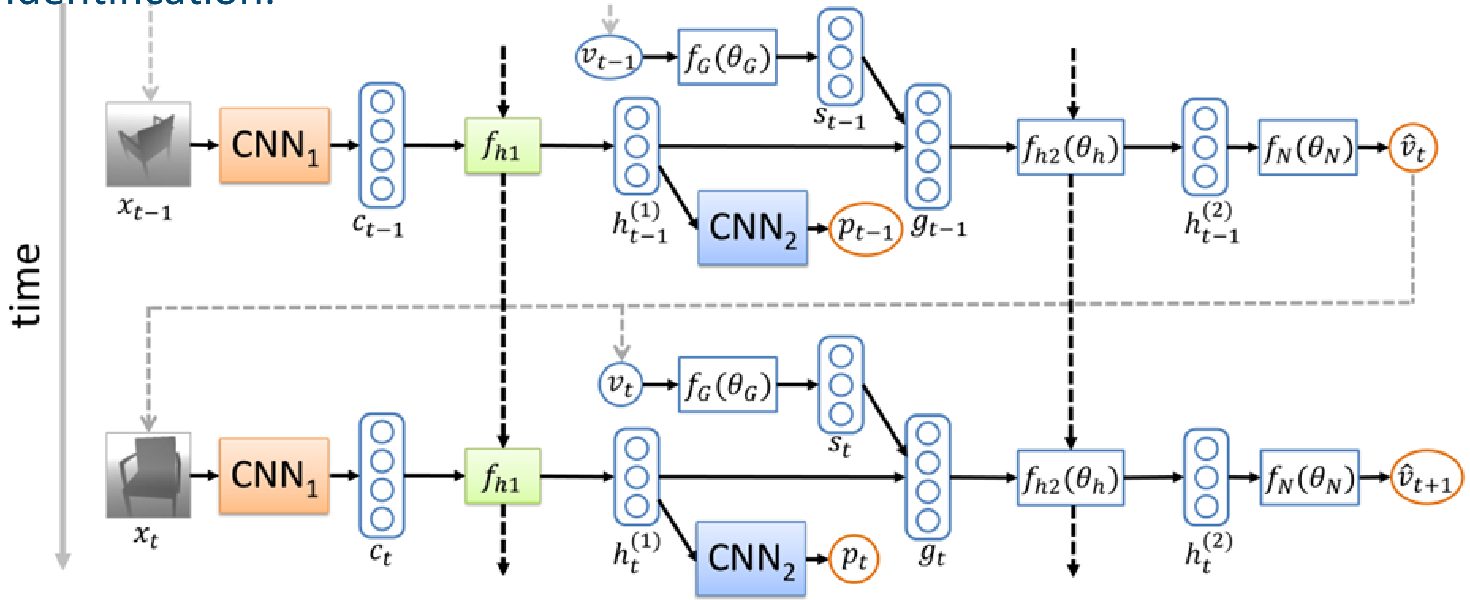
- **Training:** T=10 for each initial view from 50 evenly selected views
- First pre-train classifier (Conv2D+FCclass)
- Then tune Conv2D, FCclass, FCloc and RNN jointly
- **2 termination conditions:**
- Entropy of the classification probability <0.1
- Maximum number of timestep (10)

AOR with RL- MV-RNN

Kai Xu, et al. "3D Attention-Driven Depth Acquisition for Object Identification.“



(a) MV-CNN.



(b) MV-RNN.

- **CNN1**: extract visual features; **CNN2**: classify
- f_{h_1}: view pooling, aggregate all past visual features to $h_t^{(1)}$
- f_G: non-linear function, encode view parameters features s_t

element-wise multiplication

$$g_t = h_t^{(1)} \odot s_t$$

- f_{h_2} : RNN, aggregate all past fusion features to $h_t^{(2)}$
- f_N: <u>fully connected layer</u>, predict NBV parameters v_t

AOR with RL- MV-RNN

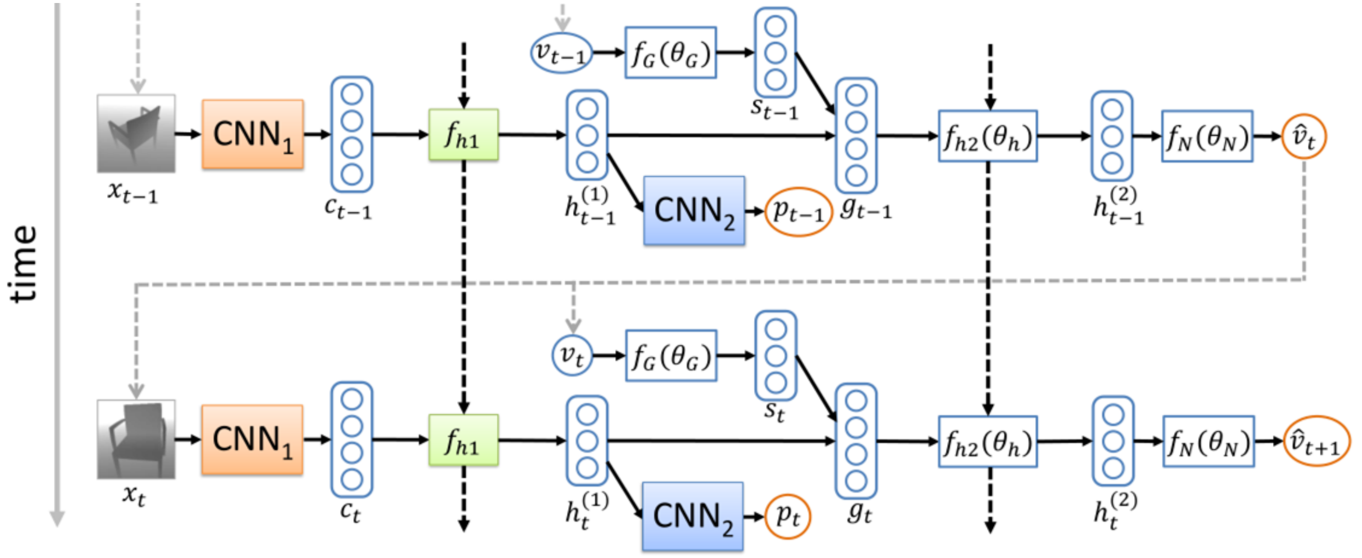
Kai Xu, et al. "3D Attention-Driven Depth Acquisition for Object Identification.“

- Pretrain feature encoding and classification networks <u>outside MV-RNN</u>
- **Training in the NBV regression network f_N :** **REINFORCE**

- starting from a random view
- To avoid examining too many view combinations, sample the views at each time step using Monte Carlo method

- **3 parts of Reward:**

- classification accuracy
- information gain
- movement cost

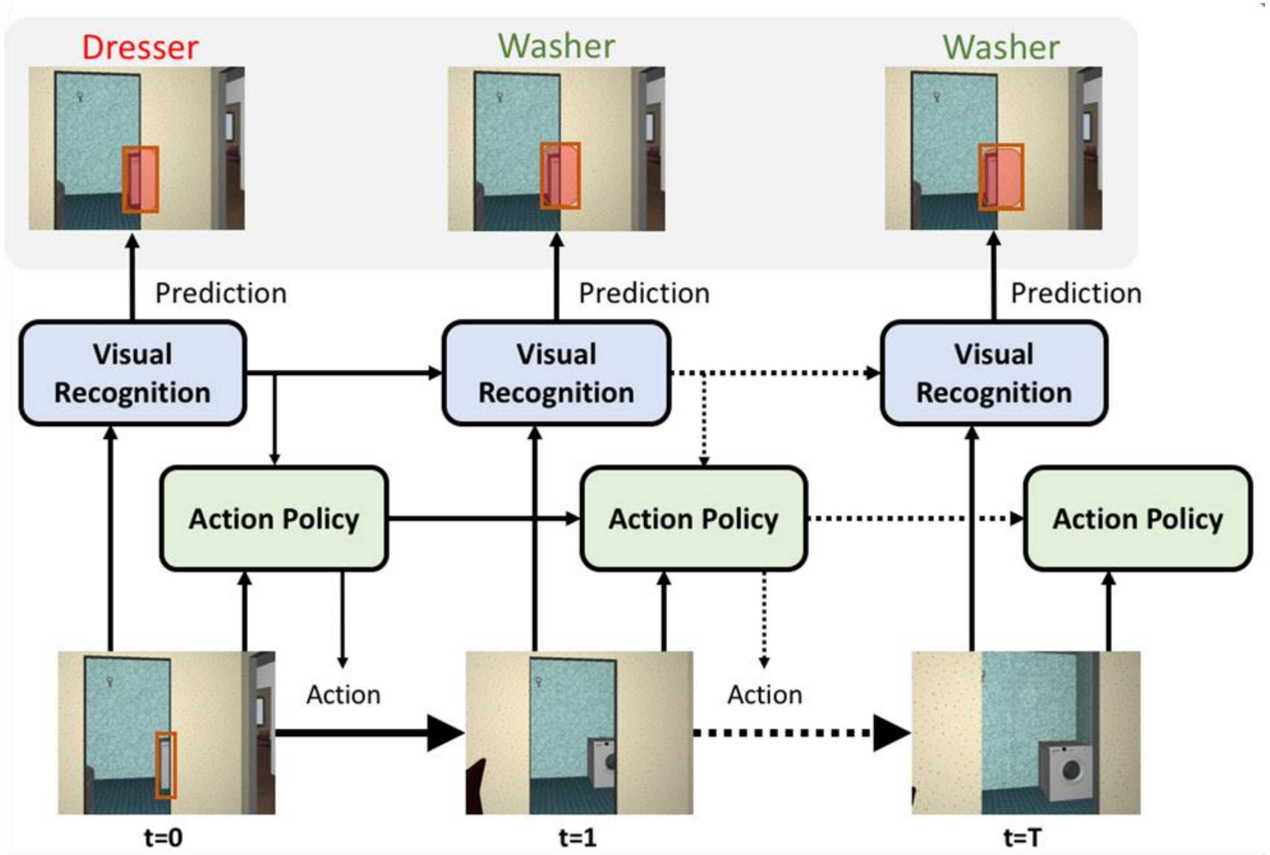


AOR with RL- EVR

Jianwei Yang, Zhile Ren, et al. "Embodied Visual Recognition."(ICCV 2019)

- Actively move in 3D environment to learn to move around to recognize occluded objects (amodal) better -> **recognition and detection**
- 3 sub-tasks:
- Object recognition
- 2D amodal localization
- 2D amodal segmentation
- 2 **separate** networks
- Perception network
- Policy network

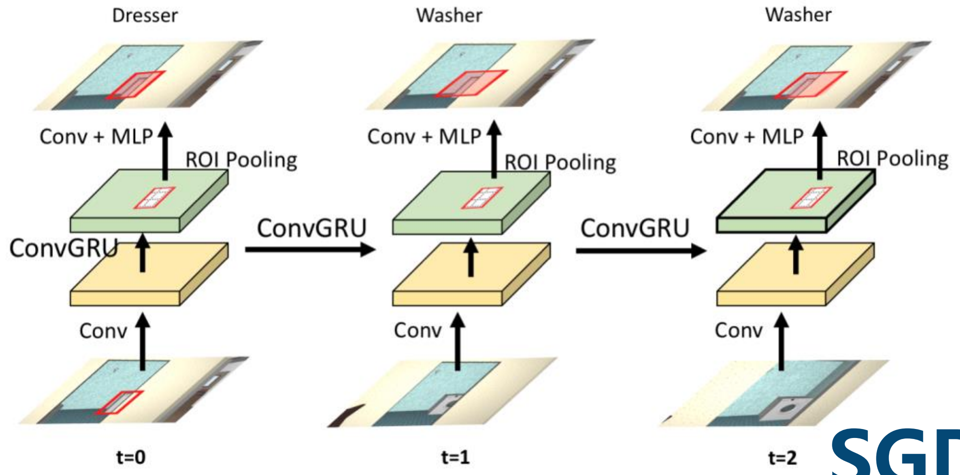
SGD + REINFORCE



AOR with RL- EVR

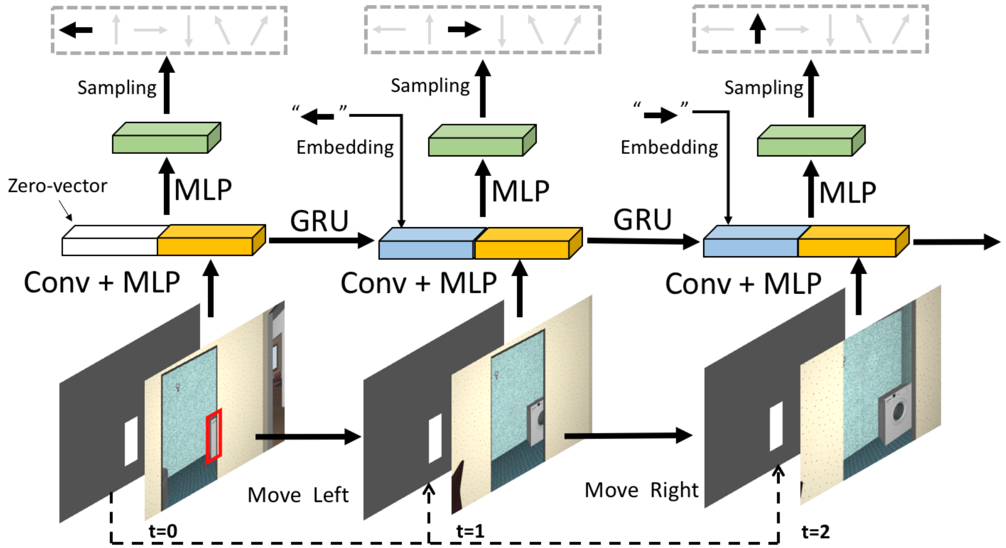
Jianwei Yang, Zhile Ren, et al. "Embodied Visual Recognition."(ICCV 2019)

- **Perception Network**: output $y_t = \{c_t, \boldsymbol{b}_t, \boldsymbol{m}_t\}$
- **CNN**: extract visual features
- **GRU**: aggerate history
- **Region-of-Interest** (RoI)



SGD

- **Policy Network**: output probabilities over discrete action space
- **CNN**: encode image features
- **MLP**: encode action features
- **GRU**: aggerate history fusion features
- **MLP with Softmax**: output probabilities



REINFORCE

AOR with RL- EVR

Jianwei Yang, Zhile Ren, et al. "Embodied Visual Recognition."(ICCV 2019)

- **Staged training** for difficulty in joint training:
- First train Perception Network with images from the shortest path*
- Then, fix the perception part and train the Policy Network
- Finally, retrain Perception Network to adapt to the learned action policy
- No other termination expect T=10

* shortest path: moves along the shortest path for training visual recognition, one of the baselines

final model (active path): shorted path + fine-tuned recognition model

Summary of AOR papers

<table>
<tr><th></th><th colspan="2">Views</th><th>Fusion network</th><th>Training</th><th>Termination</th><th>Classify</th></tr>
<tr><td>EVR (2019)</td><td colspan="2">Continuous 3D environment
+ discrete action space</td><td>GRU</td><td rowspan="3">SGD
+
REINFORCE</td><td rowspan="3">Max T</td><td rowspan="4">at each t</td></tr>
<tr><td>LookAround (CVPR2018)</td><td colspan="2" rowspan="2">Pre-defined discrete view grid
+
Sample from action distribution</td><td>LSTM</td></tr>
<tr><td>LookAhead (ECCV2016)</td><td rowspan="4">RNN
(VERAM also uses LSTM)</td></tr>
<tr><td>3DRAN (2016)</td><td>Viewing parameters in spherical coordinate system</td><td rowspan="3">+
Regress location of NBV</td><td>SGD</td><td>Max T,
Entropy < 0.1</td></tr>
<tr><td>VERAM (2016)</td><td>Pre-defined discrete view grid</td><td rowspan="2">SGD
+
REINFORCE</td><td>Max T</td><td>only at T</td></tr>
<tr><td>MV-RNN (2015)</td><td>Viewing parameters in spherical coordinate system</td><td>Max T,
Entropy < …</td><td>at each t</td></tr>
</table>

Thinking – About differentiable rendering

- Hope: differentiable renderer + RL
- If use a **differentiable renderer** (e.g. Pytorch3D)

- Predict <u>continuous</u> coordinate

  -> no need of discrete action space

- Input need to be 3D models

  -> cannot use image datasets

- **How to combine with RL without action?**

Next to do

- **Coding:**

- Learning RAM in `Pytorch` version

- Develop RL reward with RotationNet scores

- **Paper reading**:

- AOR works with Q-learning

Thinking (2) – How to train with RotationNet?

- Jointly train at the same time

  -> unbalanced training

- **Staged training with policy network (correct?)**

- Pre-pretrain RotationNet outside the pipeline
- Pre-train RotationNet with random policy
- Fix RotationNet and train policy network
- fin-tune RotationNet with trained policy