Tsinghua University

Deep Reinforcement Learning for Computer Vision

Tutors: Jiwen Lu, Liangliang Ren, and Yongming Rao



http://ivg.au.tsinghua.edu.cn/DRLCV/

Outline

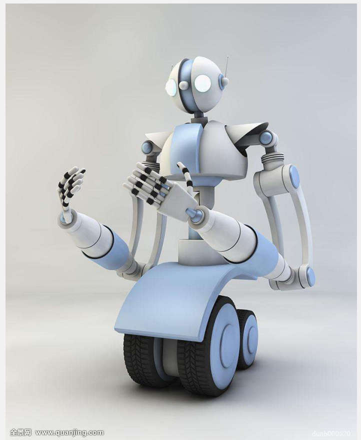
- Part 1: Introduction
- Part 2: DRL for Video Analysis

---------------------Short Break: 30 minutes----------------------

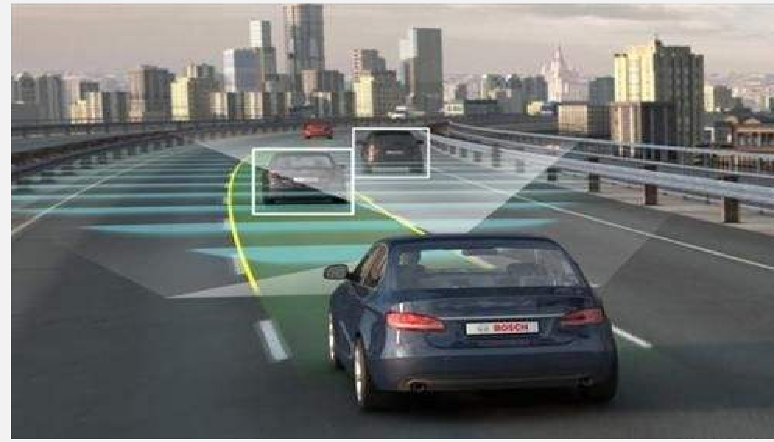
- Part 3: DRL for Network Structure Learning
- Part 4: DRL for Image Editing and Understanding
- Part 5: Conclusion and Future Directions

Part 1: Introduction

Applications for Computer Vision

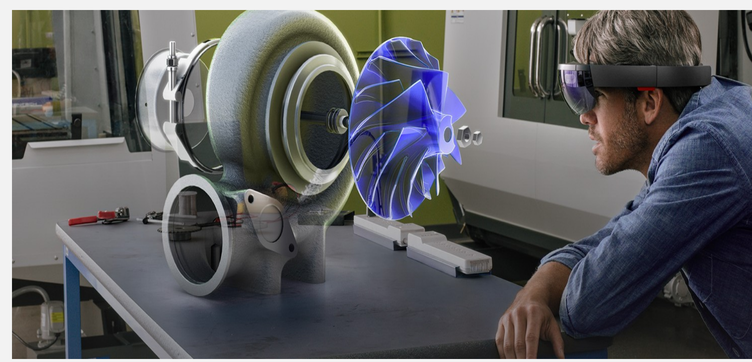


Robotics



Autonomous Vehicles

Social Media



Virtual Reality

Goals for Computer Vision

What is it about?

What are in the picture?

Where are they?

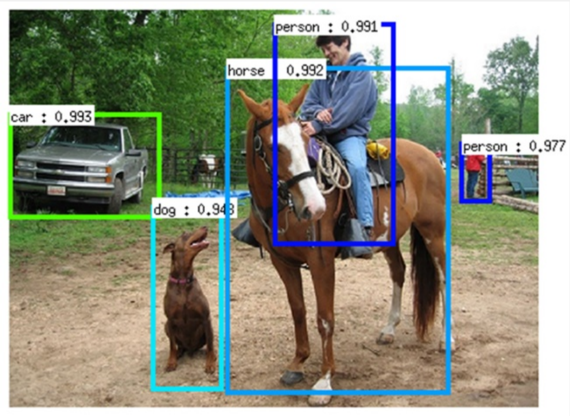
What are the relationships?

What are their spatial dependency?

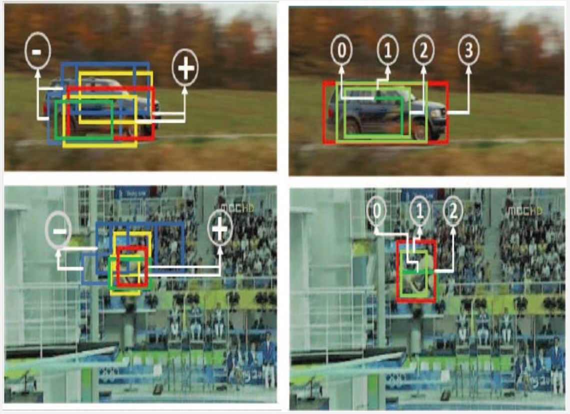
What are the relationships

between the object and the scene?

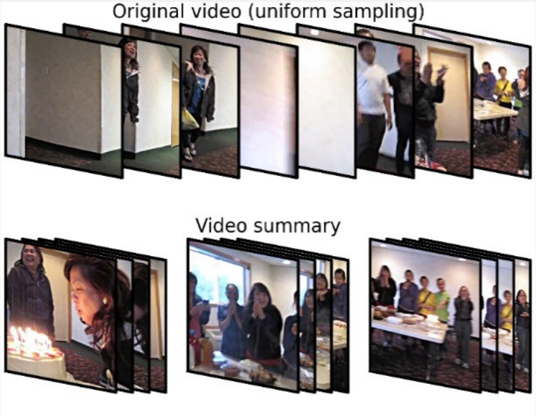
Visual Understanding



Object Detection

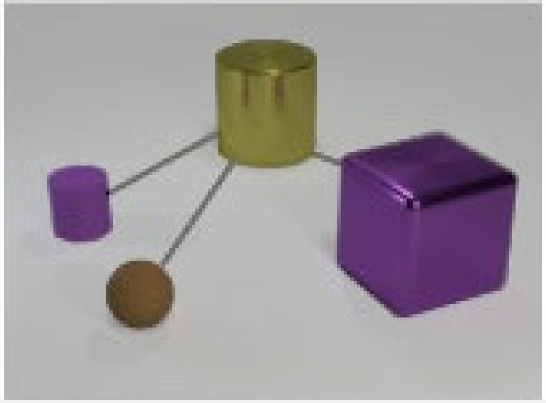


Object Tracking

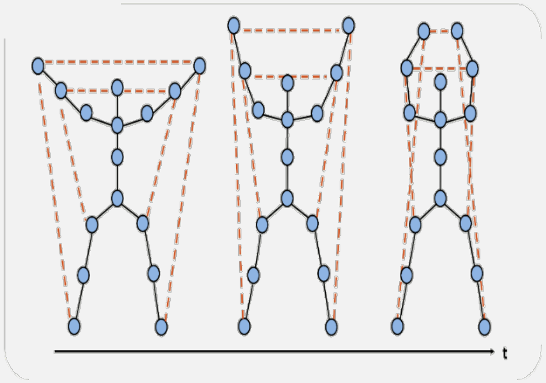


Video Summarization

Face Recognition



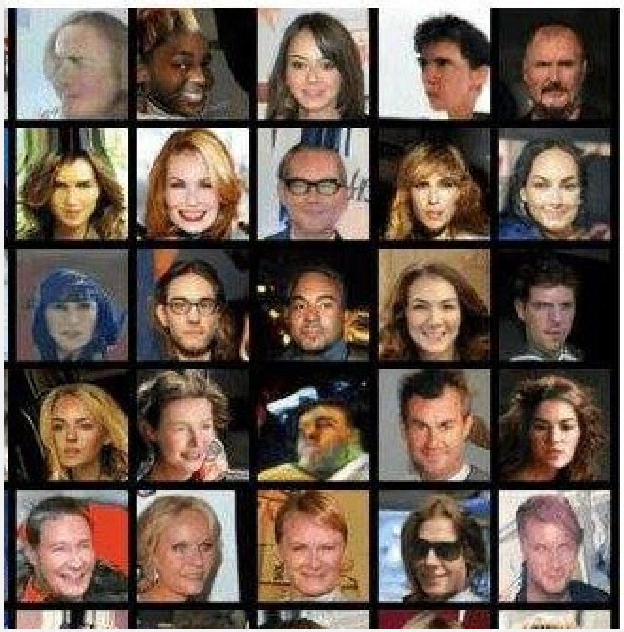
Relationship Reasoning

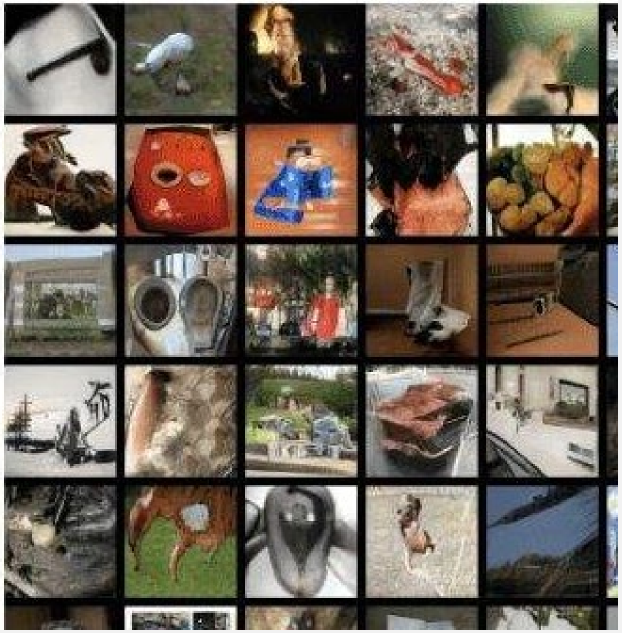


Action Recognition

Progress of Computer Vision

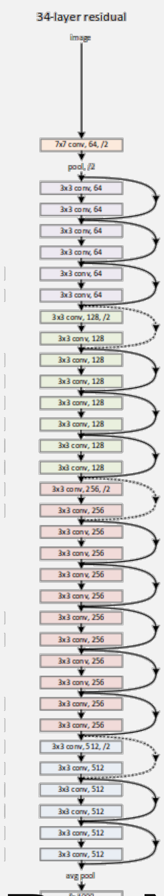
Having outperformed human-level performance on many tasks.



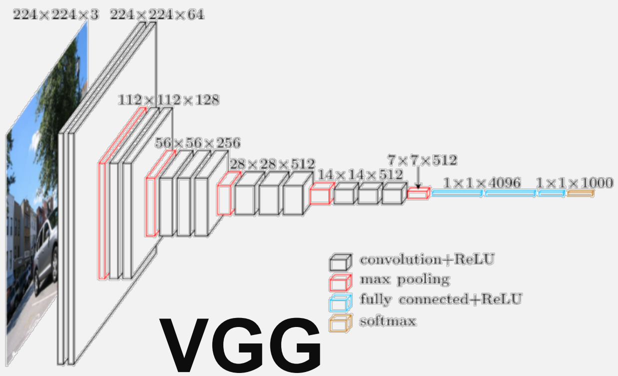


Face Recognition **Object Recognition** **Kinship Verification**

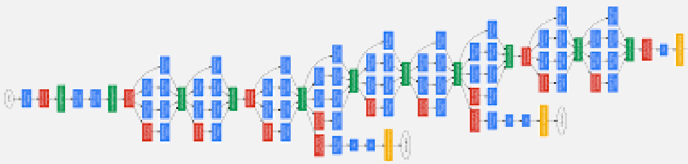
Models of Deep Neural Networks



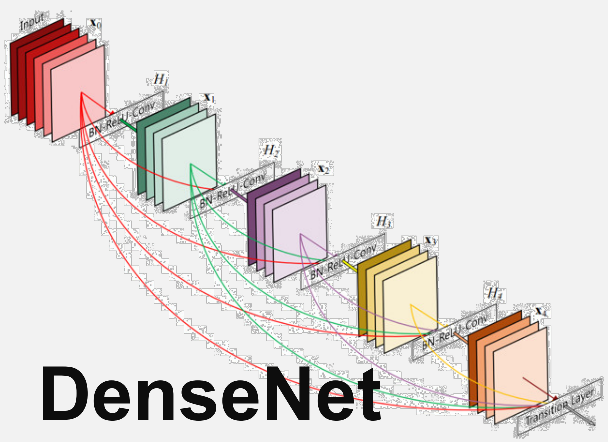
ResNet



VGG

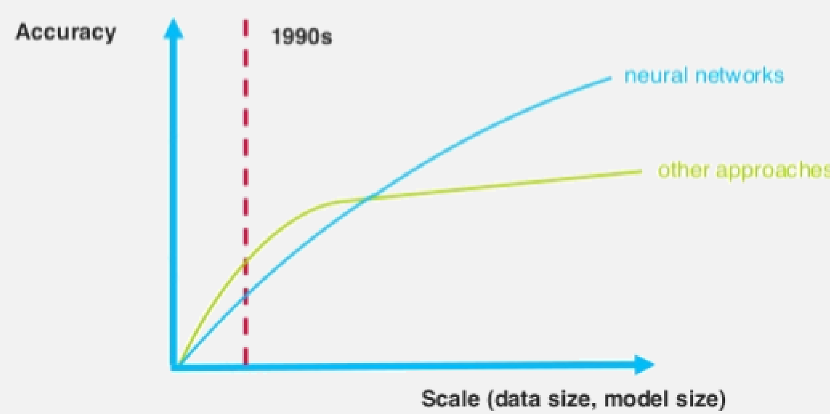


GoogLeNet



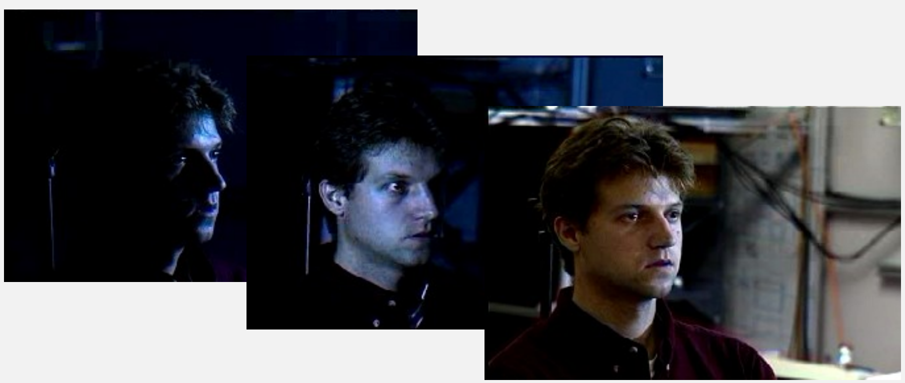
DenseNet

More Data + Bigger Models



Challenges for Visual Understanding

View-point



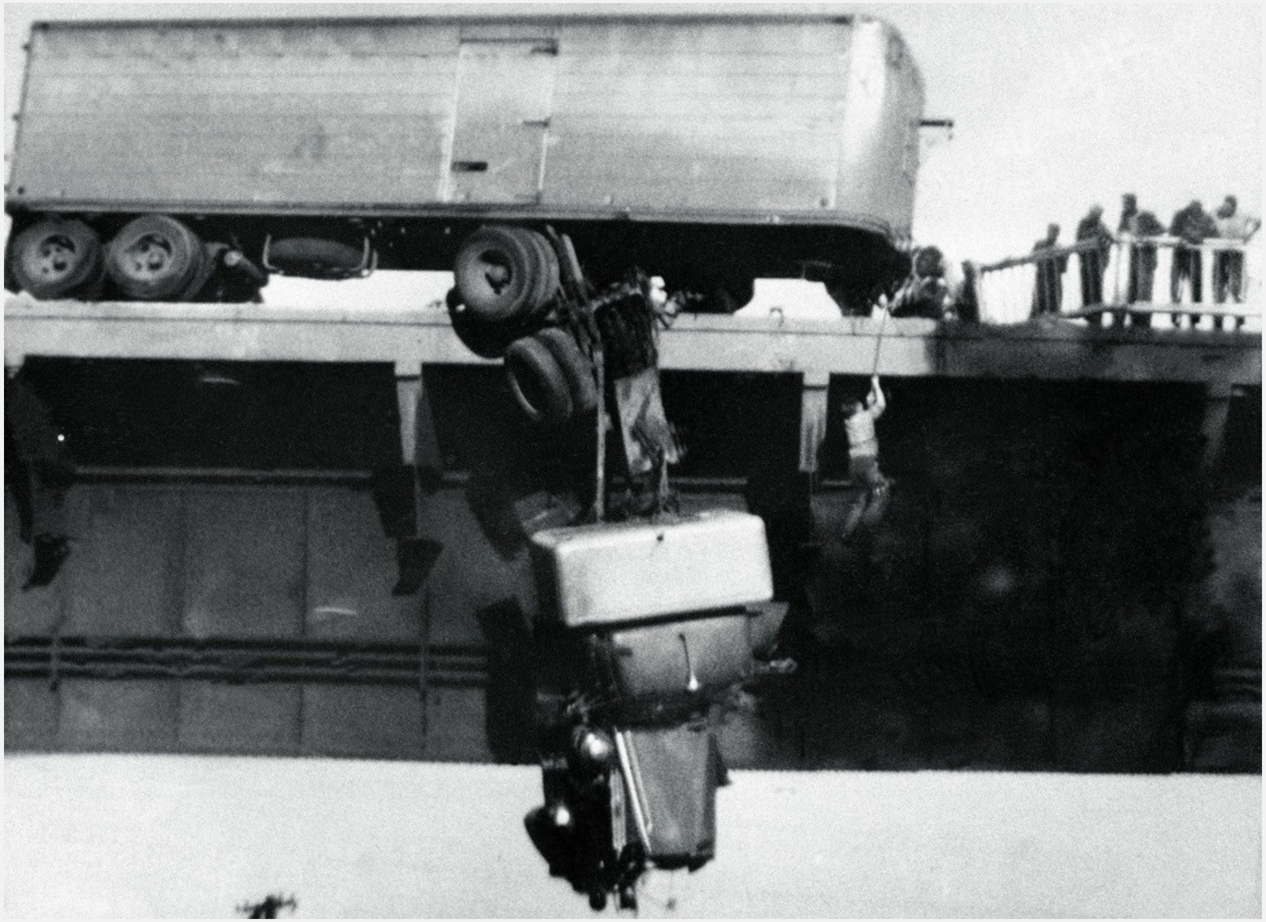
Illumination

Scale

Challenges for Visual Understanding

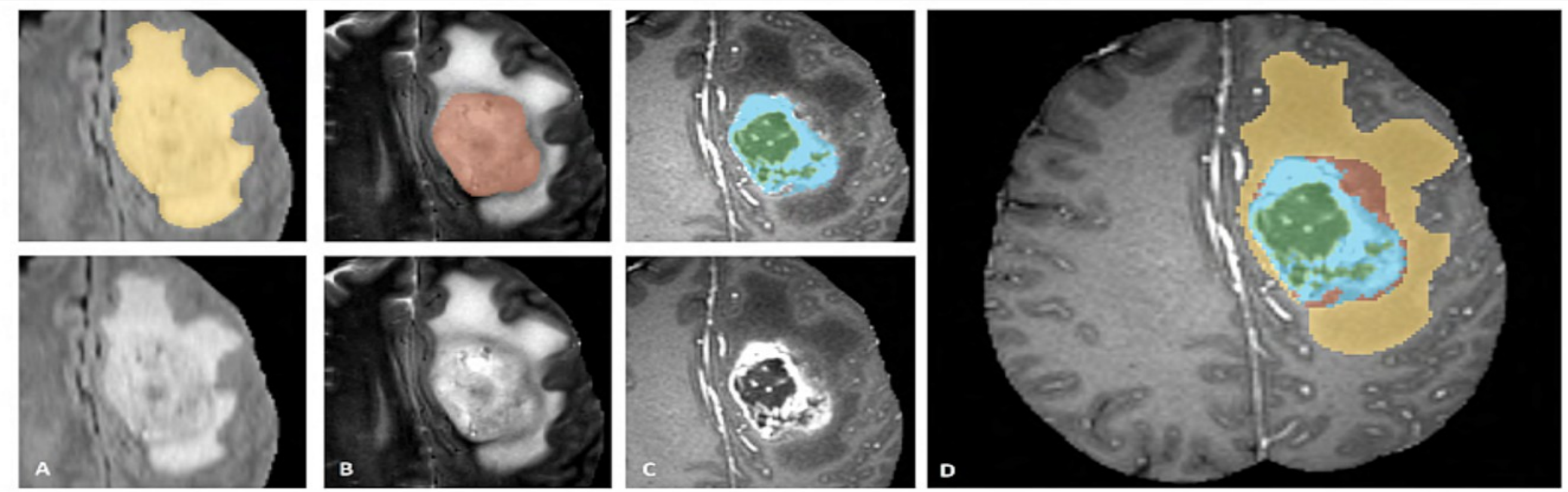
Rare Animals

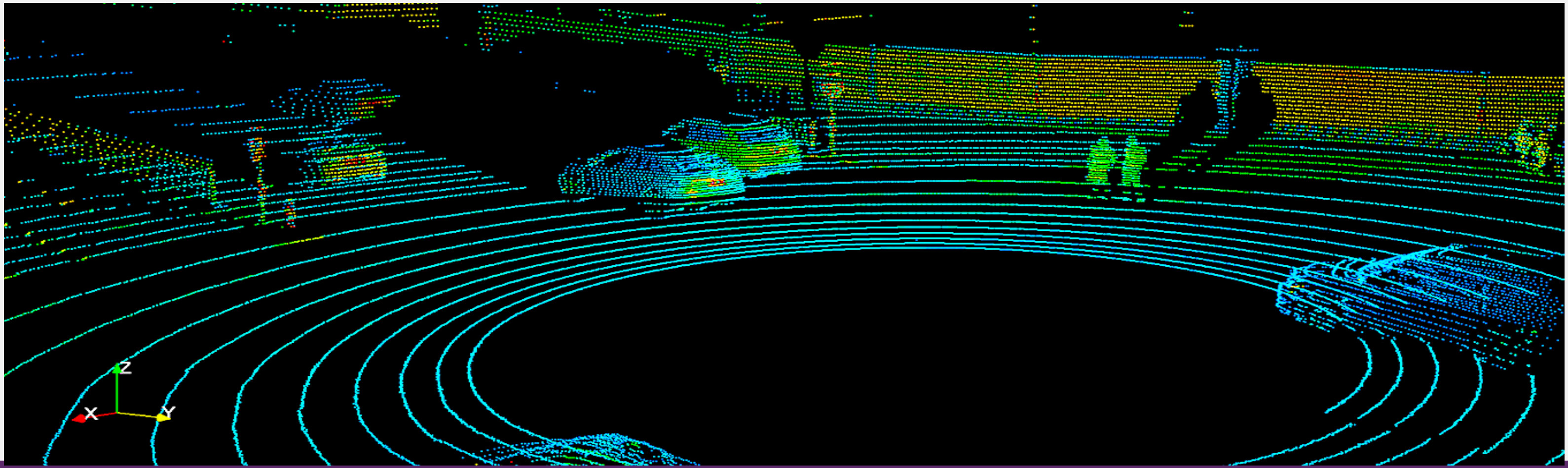
Challenges for Visual Understanding:



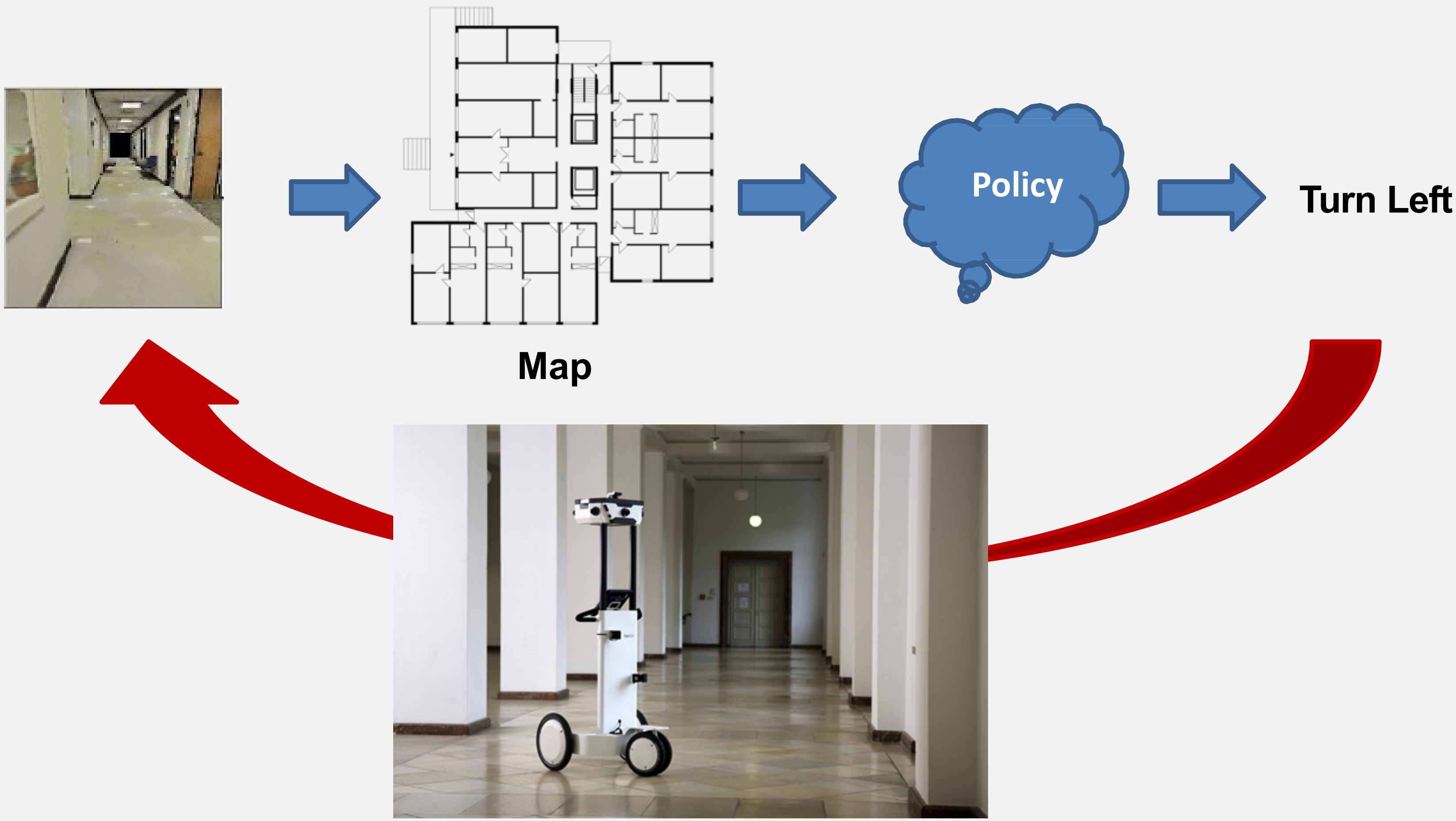
What will be going on next?

Challenges for Visual Understanding



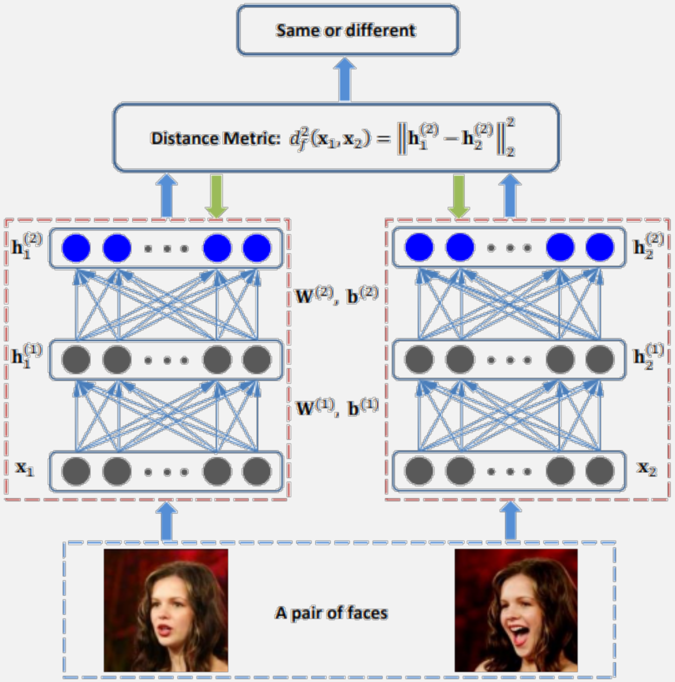


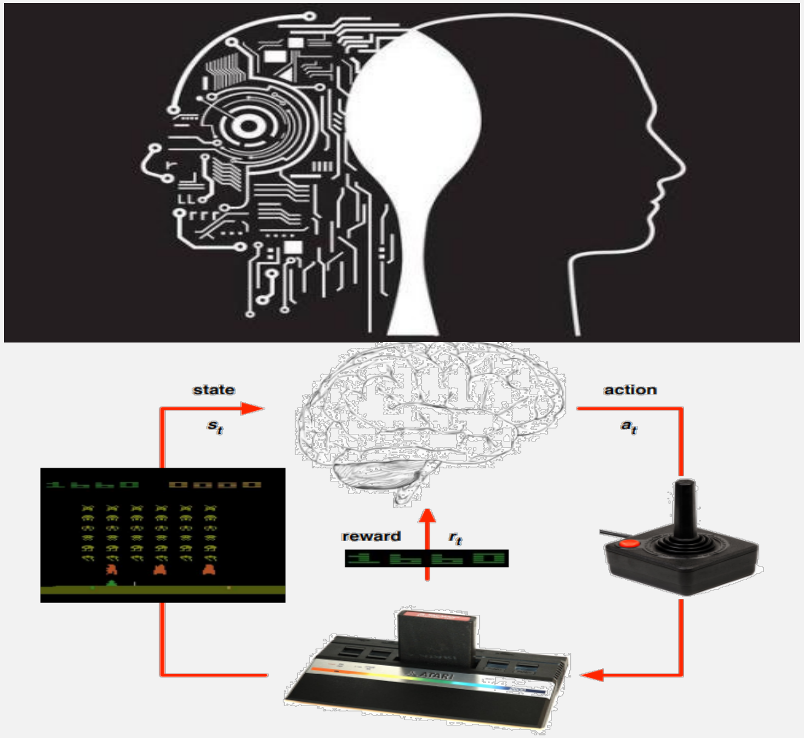
Challenges for Visual Understanding



Home-service Robot

Reinforcement Learning and Visual Understanding





To learn how human think

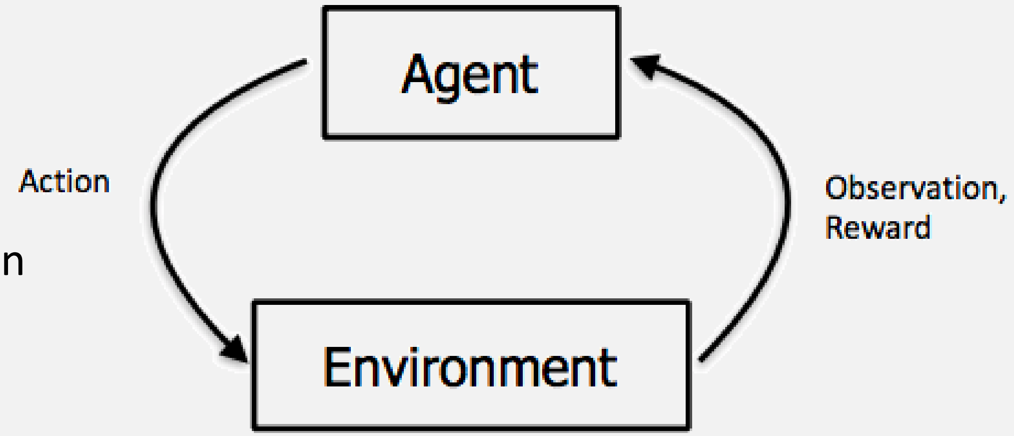
Reinforcement Learning

Policy: π

Reward Function: r

Value Function: Q

Models for Environment Interaction



Goal: Maximizing the expected rewards

Sutton R S, Barto A G. Introduction to reinforcement learning, Cambridge: MIT press, 1998.

Markov Decision Progress

- MDP (χ, A, p, q, p_0)

χ: State space

A: Action space

$\mathrm{p}(\cdot\,|x, a)$:probability over next state x_{t+1}

$q(\cdot\,|x, a)$:probability over rewards $R(x_t, a_t)$

p_o:Initial state distribution

- Policy: Mapping from the states to actions or distribution over actions

$$\mu(\cdot|x) = Pr(A)$$

Value Function

- State Value Function：

$$V^{\mu}(x) = \mathbb{E}_{\mu}\left[\sum_{t=0} \gamma^t \bar{R}(x_t, \mu(x_t)|x_0 = x)\right]$$

- State-Action Value Function:

$$Q^{\mu}(x) = \mathbb{E}_{\mu}\left[\sum_{t=0} \gamma^t \bar{R}(x_t, \mu(x_t)|x_0 = x, a_0 = a)\right]$$

Policy Evaluation

- Finding the value function of a policy

- Bellman Equations

$$V^\mu(x) = \sum_{a \in A} \mu(a|x) \left[\bar{R}(x,a) + \gamma \sum_{x' \in X} p(x'|x,a) V^\mu(x') \right]$$

$$Q^\mu(x) = \bar{R}(x,a) + \gamma \sum_{x' \in X} p(x'|x,a) \sum_{a' \in A} \mu(a'|x') Q^\mu(x',a')$$

Bellman Equations

- Bellman Optimality Equations

$$Q^*(x,a) = \bar{R}(x,a) + \gamma \sum_{x' \in X} p(x'|x,a) \max_{a' \in A} Q^{\mu}(x',a')$$

- If $Q^*(x,a) = Q^{\mu^*}(x,a)$ is available, then an optimal action for sate x is given by any

$$a^* \in \arg\max_a Q^*(x,a)$$

Policy Optimization

- Finding a policy μ^* maximizing $V^\mu(x), x \in \chi$

- Bellman Optimality Equations: $V^\mu(x) = V^*(x)$,

$$V^*(x) = \max_{a \in A} \left[\bar{R}(x,a) + \gamma \sum_{x' \in X} p(x'|x,a) V^\mu(x') \right]$$

$$\mu^* = \arg\max_{\mu} V^\mu(x)$$

Learning Methods

- Offline Learning
  Learning while interacting with a simulator

- Online learning
  Leaning while interacting with environment

Offline Learning

- Agent interacts with a simulator
- Rewards/costs do not matter
  no exploration/exploitation tradeoff
- Computation time between actions is not critical
- Simulator can produce as much as data we wish

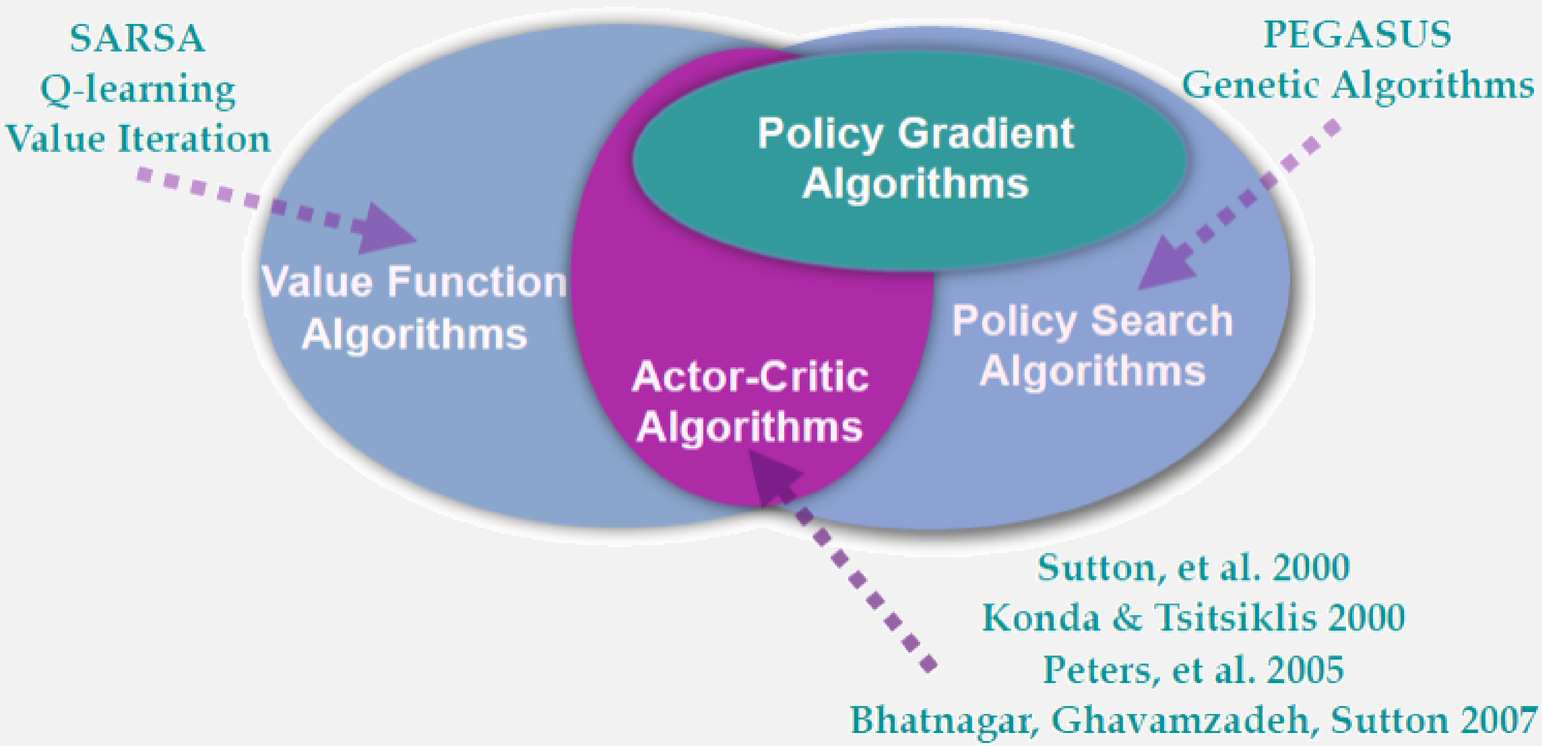
- Main Challenge
  How to minimize time to converge to optimal policy

Online Learning

- No simulator - Direct interaction with environment
- Agent receives reward/cost for each action
- Main Challenges
  - ➢Exploration/exploitation tradeoff

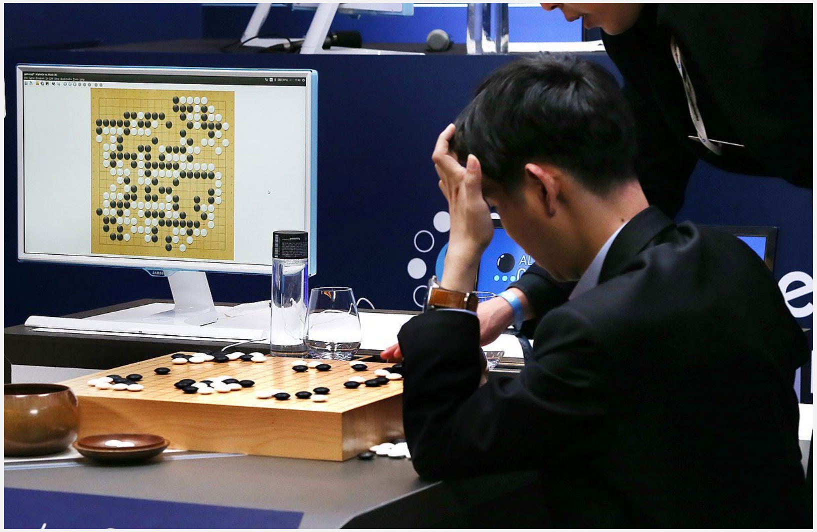
    Should actions be picked to maximize immediate reward or to maximize information gain to improve policy
  - ➢Real-time execution of actions
  - ➢Limited amount of data since interaction with environment is required

Solutions


SARSA
Q-learning
Value Iteration
PEGASUS
Genetic Algorithms
Policy Gradient Algorithms
Value Function Algorithms
Policy Search Algorithms
Actor-Critic Algorithms
Sutton, et al. 2000
Konda & Tsitsiklis 2000
Peters, et al. 2005
Bhatnagar, Ghavamzadeh, Sutton 2007

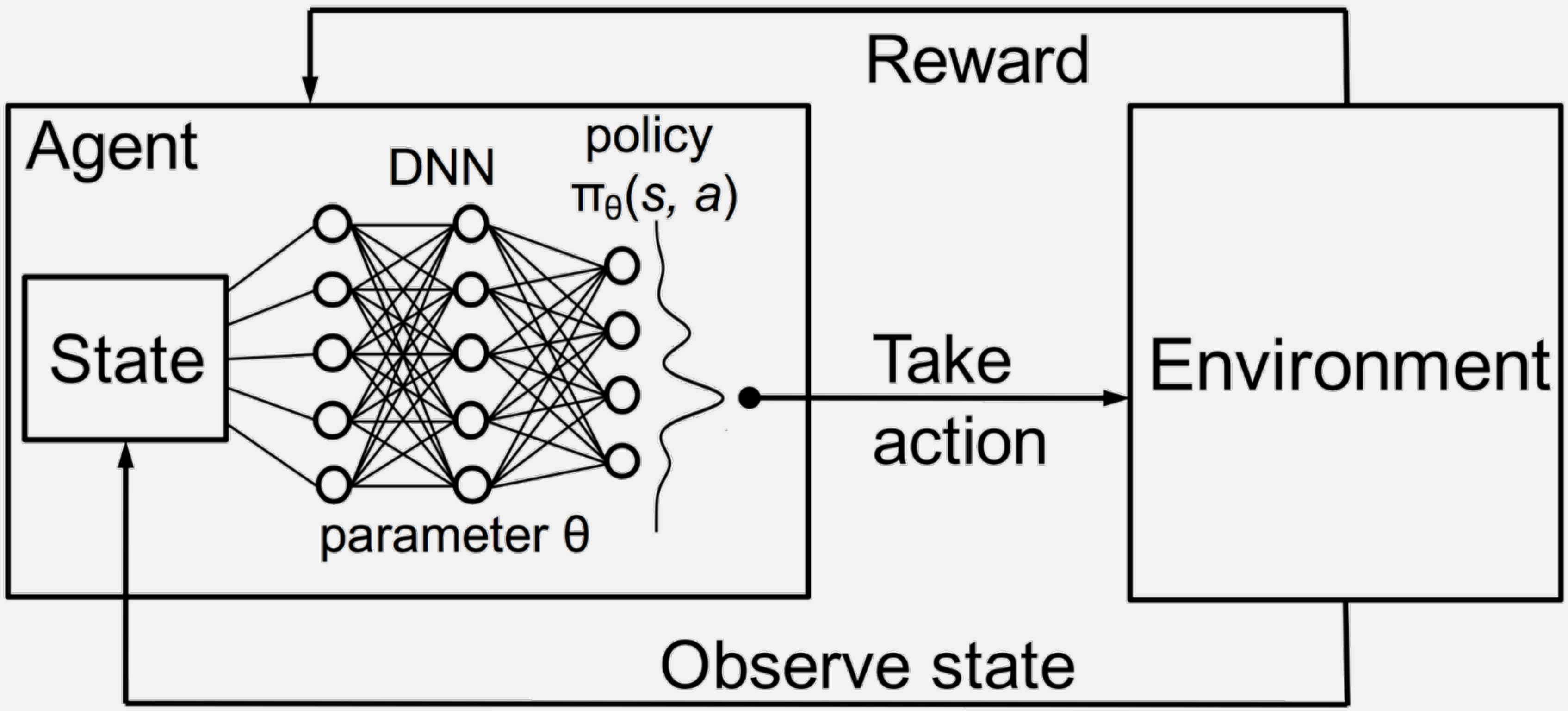
Deep Reinforcement Learning

Google DeepMind:
AlphaGo Defeated Lee Shishi

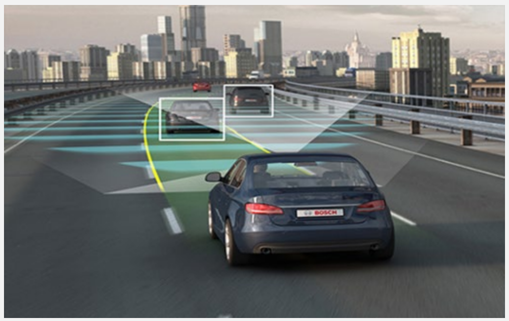


[1]Human-level control through deep reinforcement learning（Nature 2015）
[2]Mastering the game of Go with deep neural networks and tree search (*Nature* 2016).

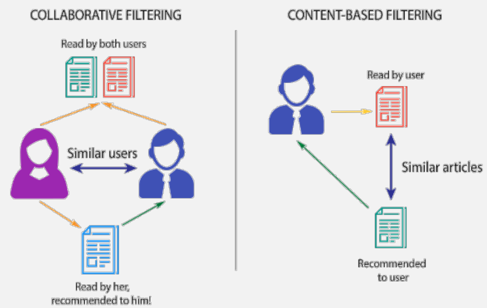
The Basic Model of Deep Reinforcement Learning



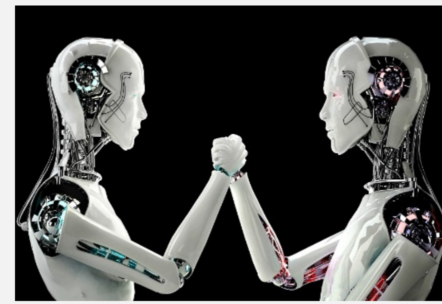
Applications for Deep Reinforcement Learning



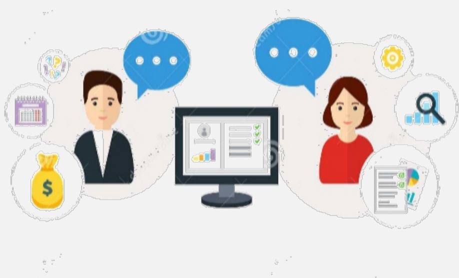
Autonomous Driving



Recommendation system



Robotics



Inventory management

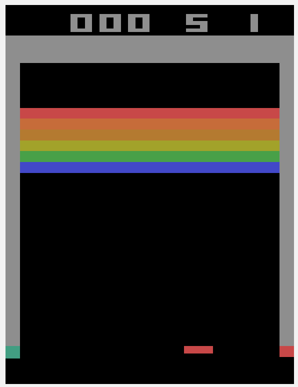
Financial investment

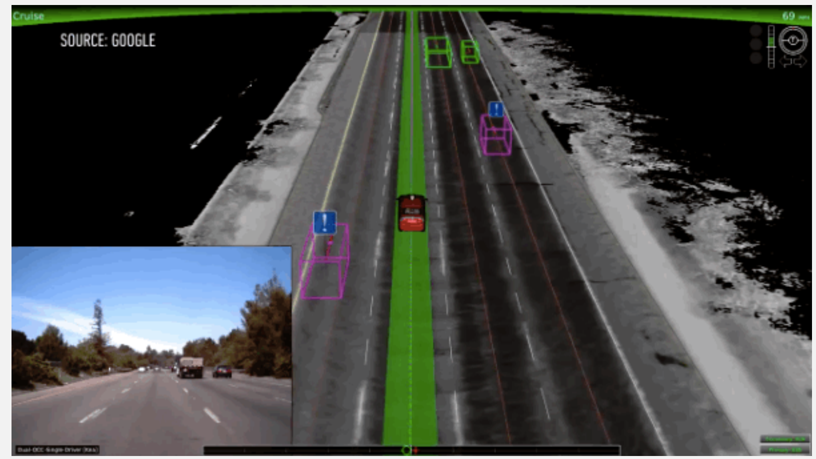
Medical assistant

Deep Reinforcement Learning

- **Policy Learning**

Atari Game





Autonomous Driving

- **Discrete Optimazation**

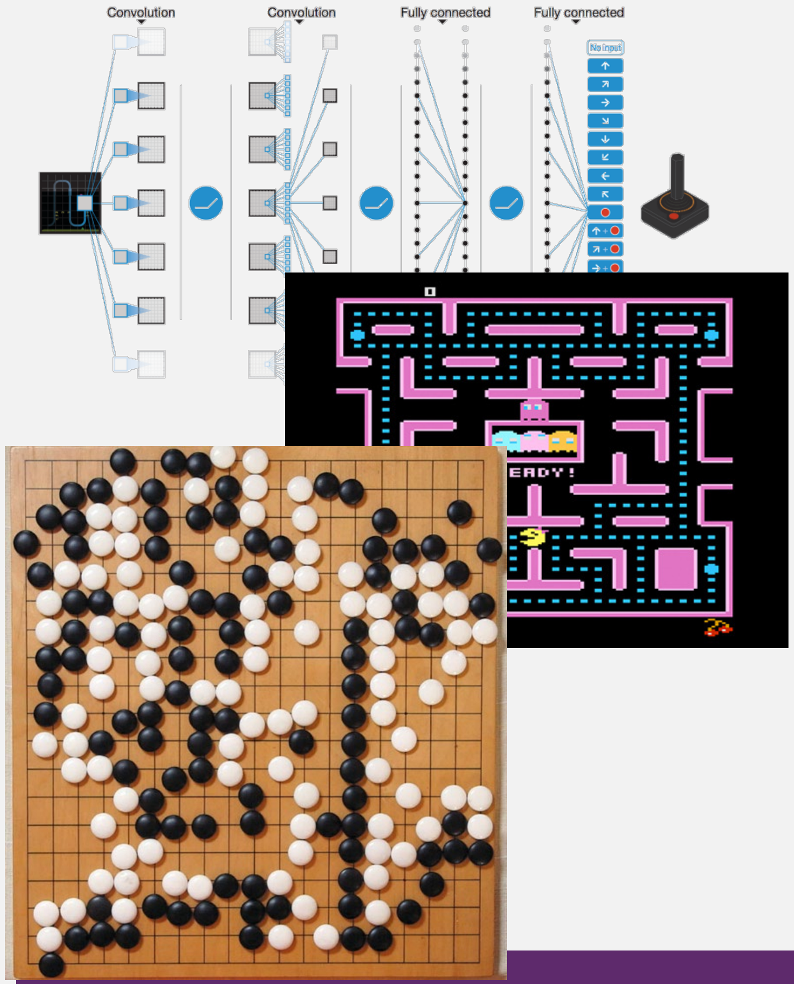
$$r\ \frac{1}{4}(\mu) = R(s;a)r\ \ \log \frac{1}{4}_{\mu}(s;a)$$

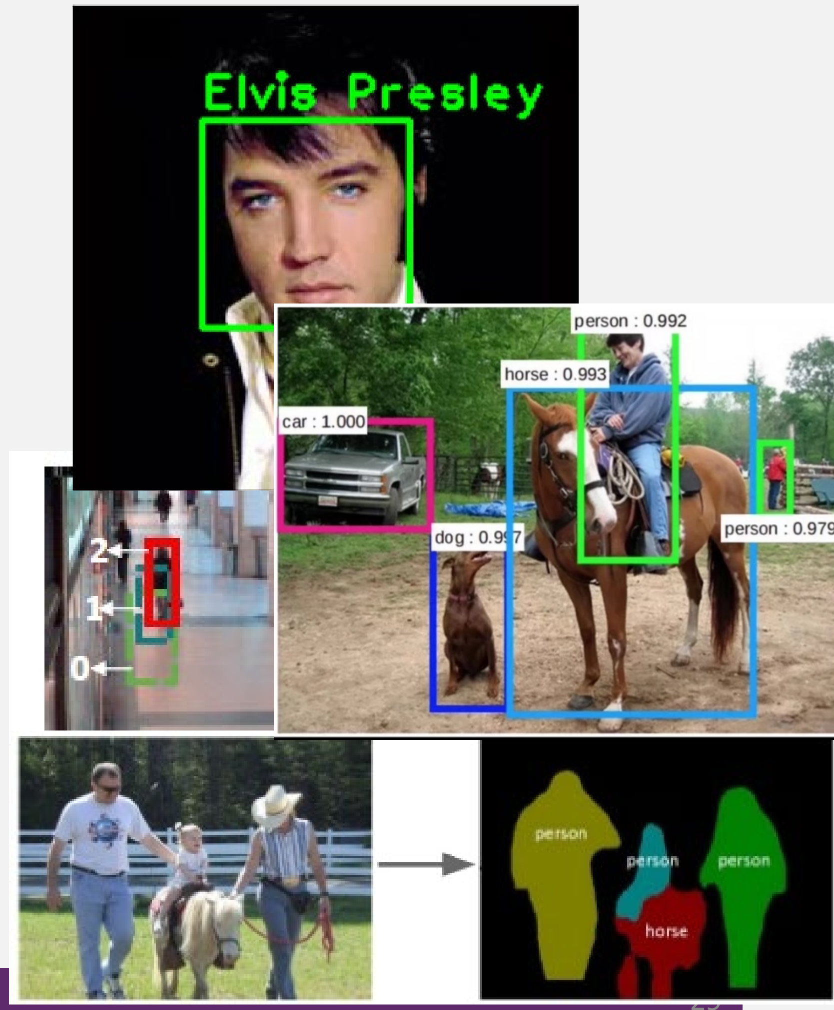
From reward function R(s,a) to the gradient of decision network

- **Unsupervised (Weakly supervised) Learning**

AlphaZero Learns from the rule of GO rather than chess manual

From Deep Reinforcement Learning to Computer Vision

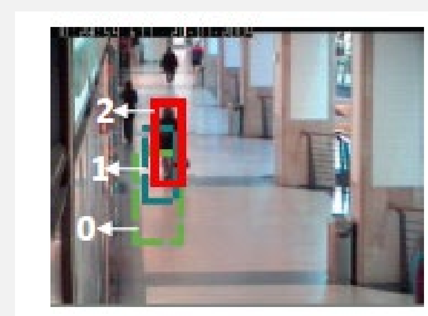




From Deep Reinforcement Learning to Computer Vision

- **Decision Problem: Markov Decision Process (MDP) Modeling**

	Atari Game	Object Tracking
State:	Current Game States (Ball's position, Bricks condition)	Input Image + Current Tracking state
Action:	turn left, turn right or stop	Move the bbx or finish tracking
Reward:	Game scores	Accuracy of Tracking



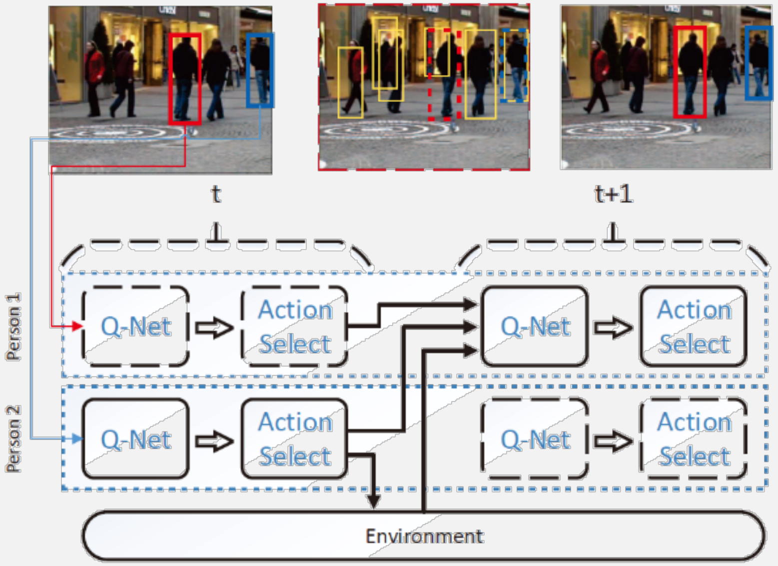
Part 2: DRL for Video Analysis

DRL for Video Analysis

- Object (face) Detection, Tracking, and Recognition
- Action Detection, Recognition, and Prediction
- Video Summary and Caption

Caption #1: A woman offers her dog some food.
Caption #2: A woman is eating and sharing food with her dog.
Caption #3: A woman is sharing a snack with a dog.

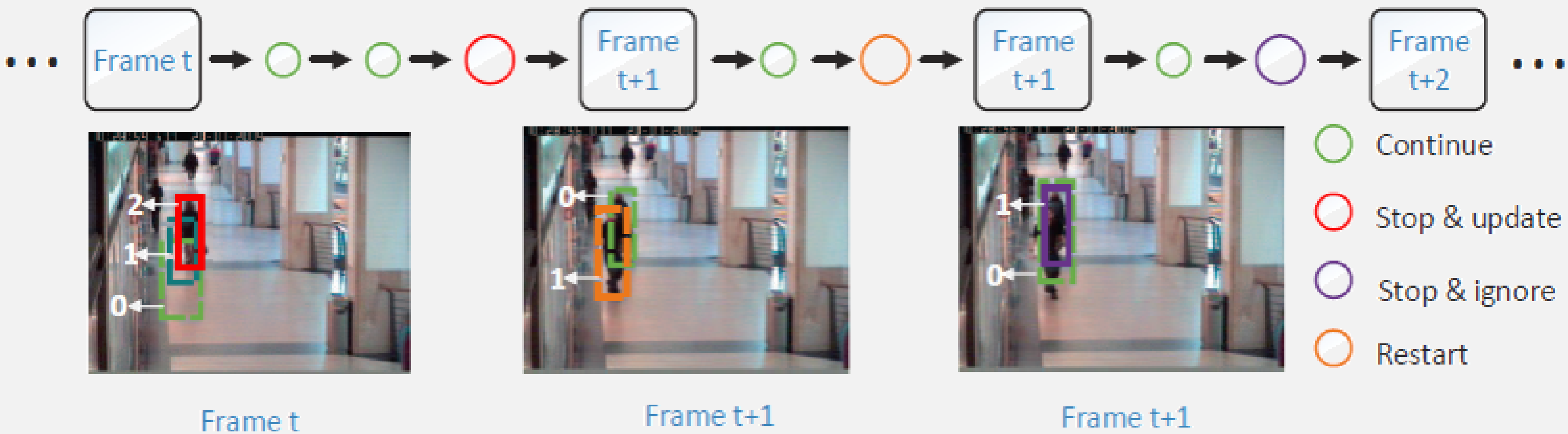
Caption: A person sits on a bed and puts a laptop into a bag. The person stands up, puts the bag on one shoulder, and walks out of the room.



Tsinghua University
i-VisionGroup

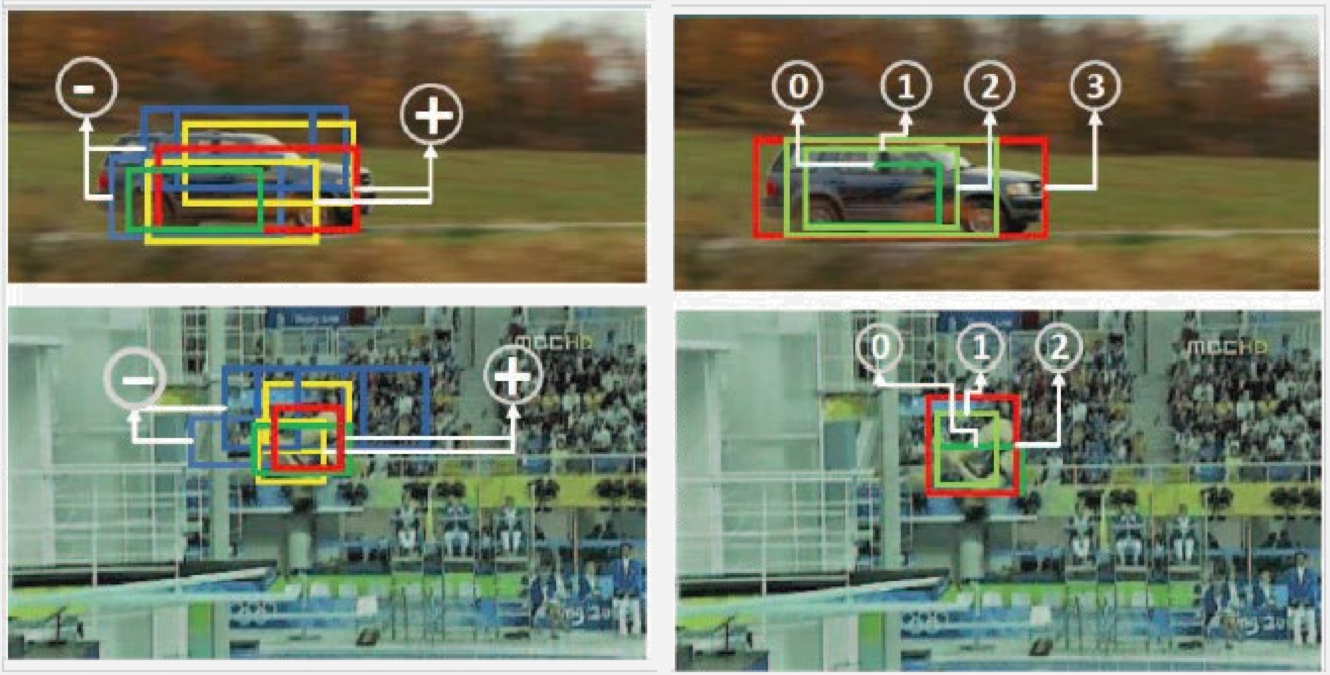
DRL for Video Analysis

- Video $\mathrm{V} = \{\mathrm{I_t} | \mathrm{i} = 0, 1, \cdots, N-1, N\}$
  $x_t = (I_t, h_t) \in \chi$: State space
  $a_t : \mathrm{h}_t \rightarrow h_{t+1}, \mathrm{a} \in A$: Action space
  $\mathrm{p}(\cdot | x, a)$:probability over next state x_{t+1}
  $q(\cdot | x, a)$:probability over rewards $R(x_t, a_t)$
- Policy: Mapping from the states to actions or distribution over actions



Object (face) Detection, Tracking, and Recognition

DRL with iterative shift for visual tracking



(a) Classification based methods (b) Iterative shift based method

Traditional methods: sampling candidate bbx and performs classification.
Problems: low efficiency, hard to overcome the quick shift and deformation.

Liangliang Ren, Xin Yuan, **Jiwen Lu**, Ming Yang, and Jie Zhou. "Deep Reinforcement Learning with Iterative Shift for Visual Tracking." ECCV2018.

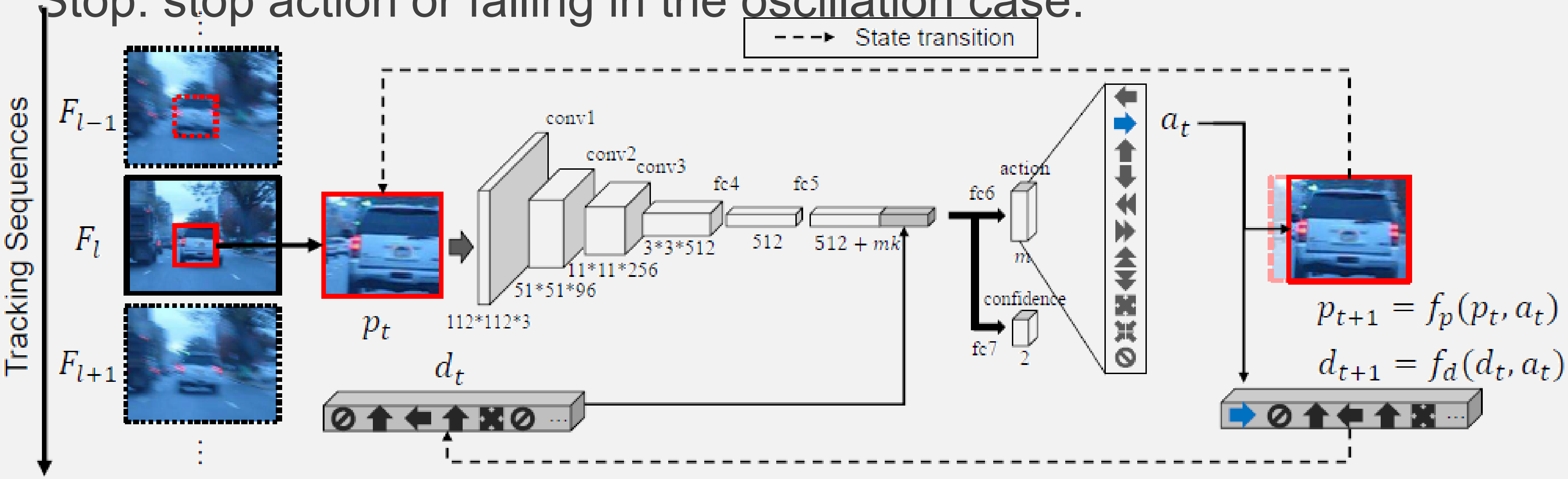
Action-Decision Networks for Visual Tracking with Deep Reinforcement Learning

- Sate:

$p_t \in R^{112*112*3} = \phi(b_t, F), b_t = \{x^{(t)}, y^{(t)}, w^{(t)}, h^{(t)}\}$:image patch within the bounding box

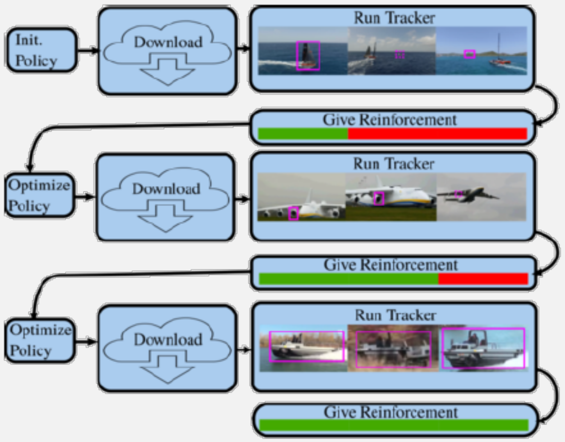
$d_t \in R^{110}$, k actions at t-th iteration,

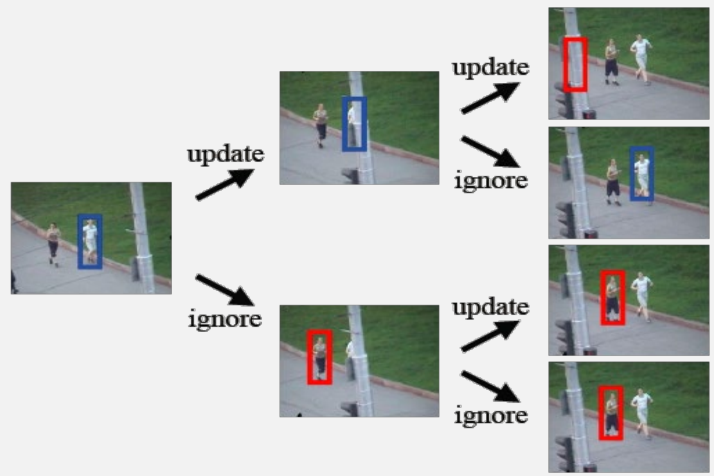
Stop: stop action or falling in the oscillation case.

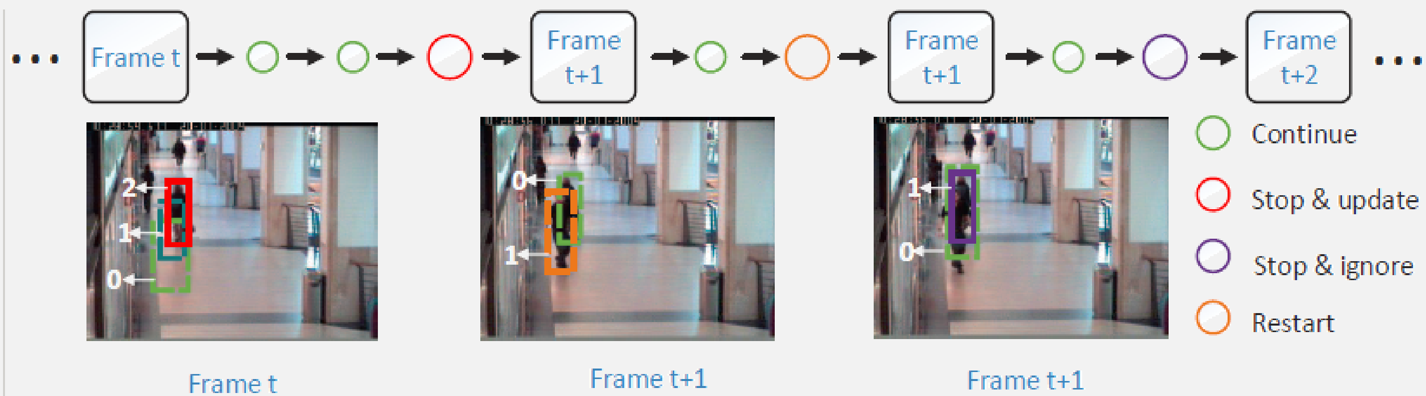


Yun, S., Choi, J., Yoo, Y., Yun, K., & Choi, J. Y. (2017, July). Action-Decision Networks for Visual Tracking with Deep Reinforcement Learning. *CVPR2017*

Tracking as Online Decision-Making:Learning a Policy from Streaming Videos with Reinforcement Learning



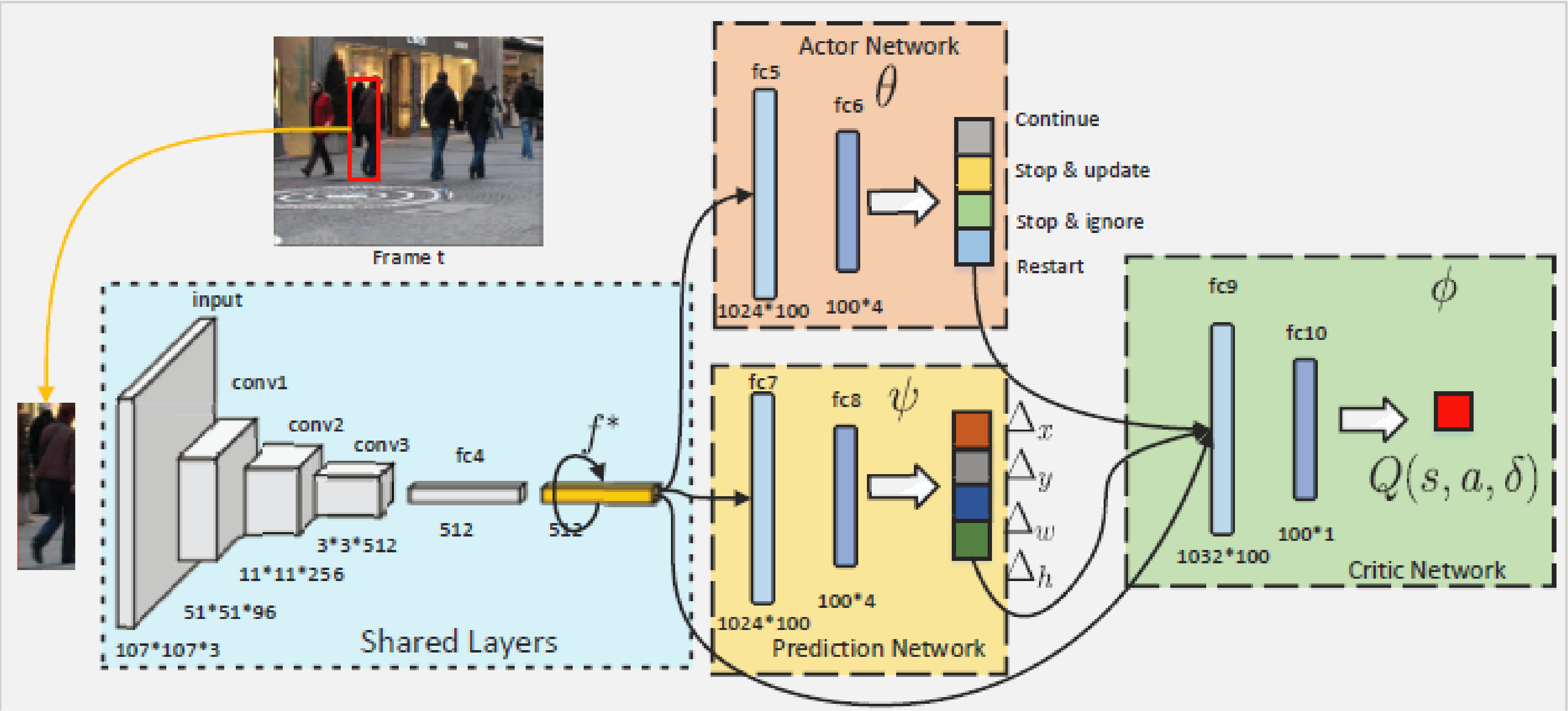




Supancic III, James Steven, and Deva Ramanan. "Tracking as Online Decision-Making: Learning a Policy from Streaming Videos with Reinforcement Learning." ICCV.2017.

DRL with iterative shift for visual tracking

Network Architecture



DRL with iterative shift for visual tracking

Actions:

$$A = \{\text{continue}, \text{stop \& update}, \text{stop \& ignore}, \text{restart}\}$$

Reward:

$$r_{t,K_t} = \begin{cases} -1 & g(l_t^*, l_{t,K_t}) \geq 0.7 \\ 0 & 0.4 \leq g(l_t^*, l_{t,K_t}) \leq 0.7 \\ 1 & \text{else} \end{cases}.$$

$$r_{t,K_t} = \begin{cases} 10/K_t & g(l_t^*, l_{t,K_t}) \geq 0.7 \\ 0 & 0.4 \leq g(l_t^*, l_{t,K_t}) \leq 0.7 \\ -5 & \text{else} \end{cases}.$$

$$r_{t,k} = \begin{cases} 1 & \Delta_{IoU} \geq \epsilon \\ 0 & -\epsilon < \Delta_{IoU} < \epsilon \\ -1 & \Delta_{IoU} \leq -\epsilon \end{cases},$$

Formulation:

$$\phi = \arg\min_{\phi} L(\phi) = E_{s,a}(Q(s,a|\phi) - r - \gamma Q(s', a'|\phi^-))^2,$$

$$\theta = \arg\min_{\theta} J(\theta) = -E_{s,a} \log(\pi(a; s|\theta))\hat{A}(s,a).$$

DRL with iterative shift for visual tracking

Experimental Results on the TC128 and VOT-2016 Dataset

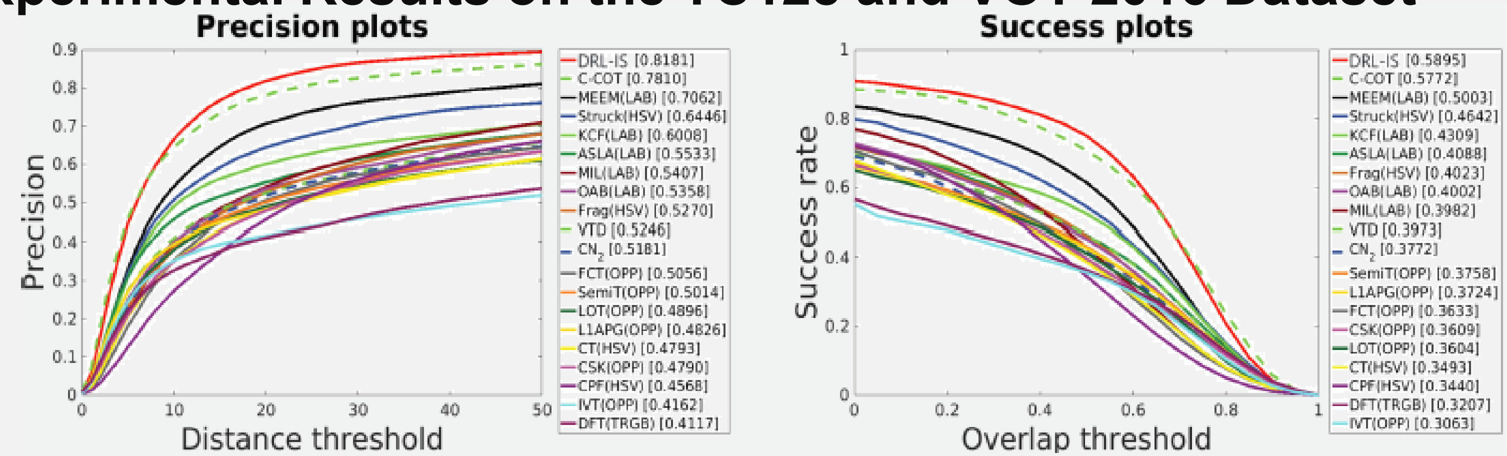


Fig. 7. The precision and success plots over all sequences by using one-pass evaluation on the Temple-Color Dataset. The legend contains the average distance precision score and the area-under-the-curve score for each tracker

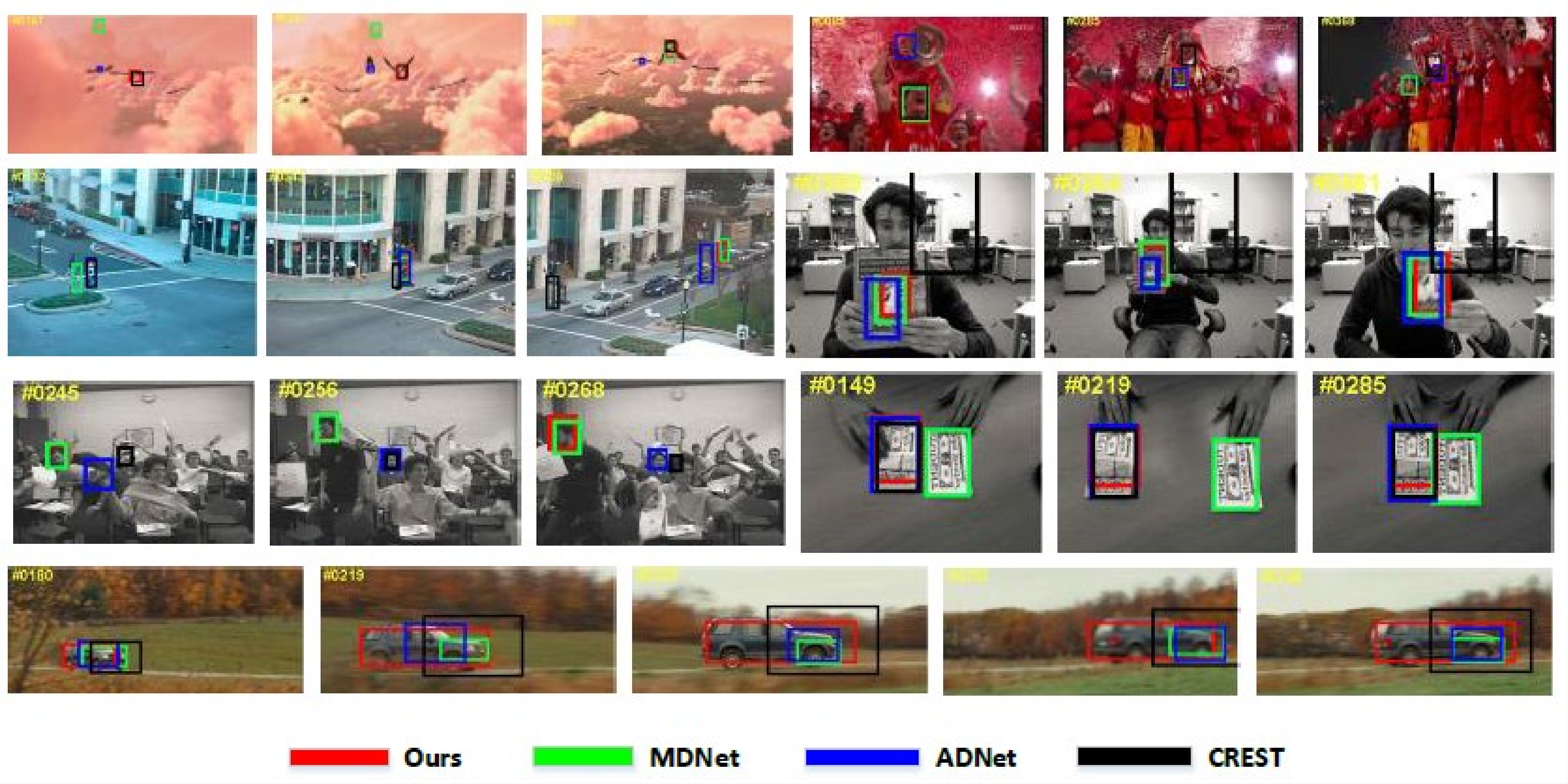
Table 1. Comparison with state-of-the-art methods in terms of robustness and accuracy ranking on the VOT-2016 dataset(the lower the better)

Baseline	MDNet_N	DeepSRDCF	Staple	MLDF	SSAT	TCNN	C-COT	DRL-IS
Robustness	5.75	5.92	5.70	4.23	4.60	4.18	2.92	**2.70**
Accuracy	4.63	4.88	4.23	6.17	**3.42**	4.22	4.85	3.60

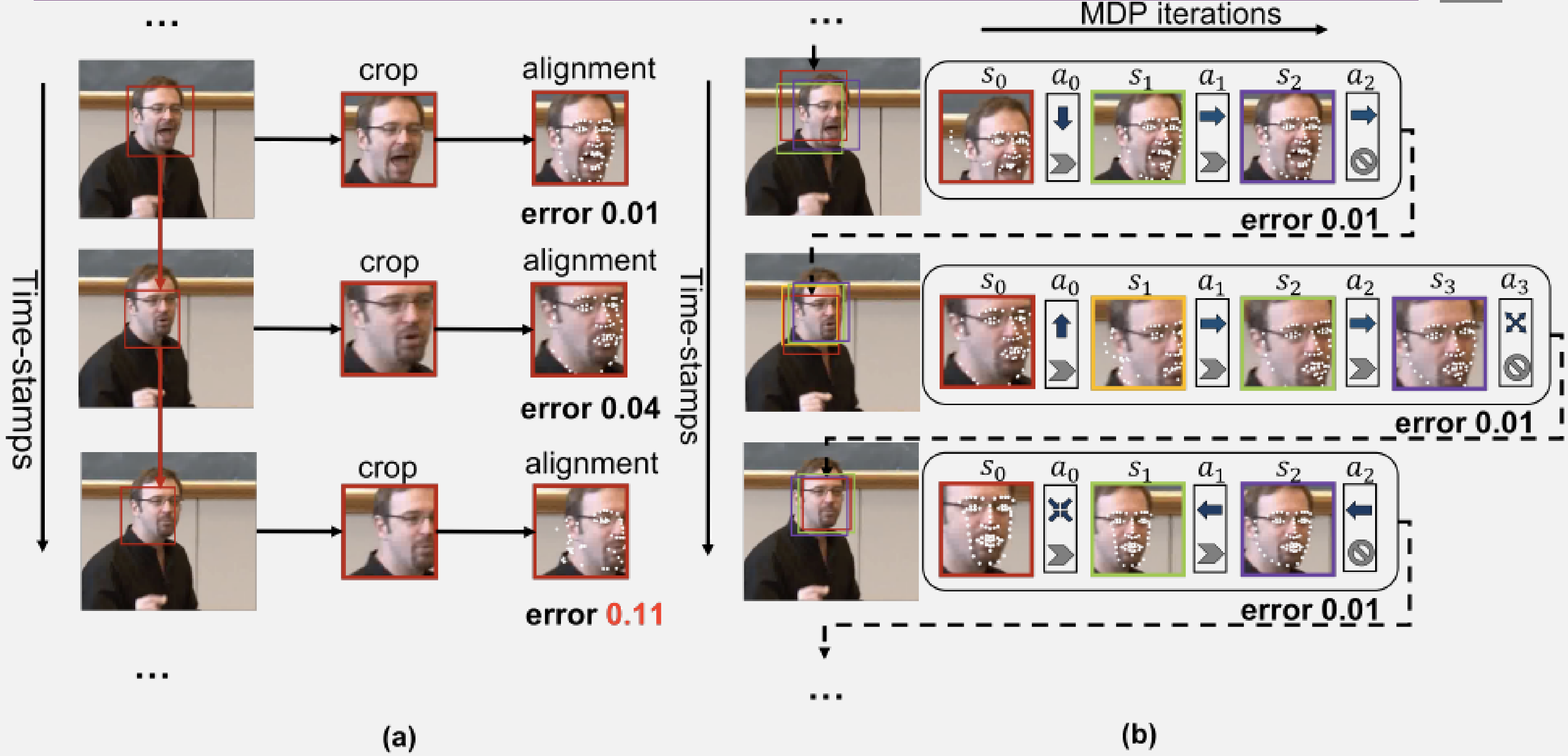
DRL with iterative shift for visual tracking

DRL with iterative shift for visual tracking

Visualization:



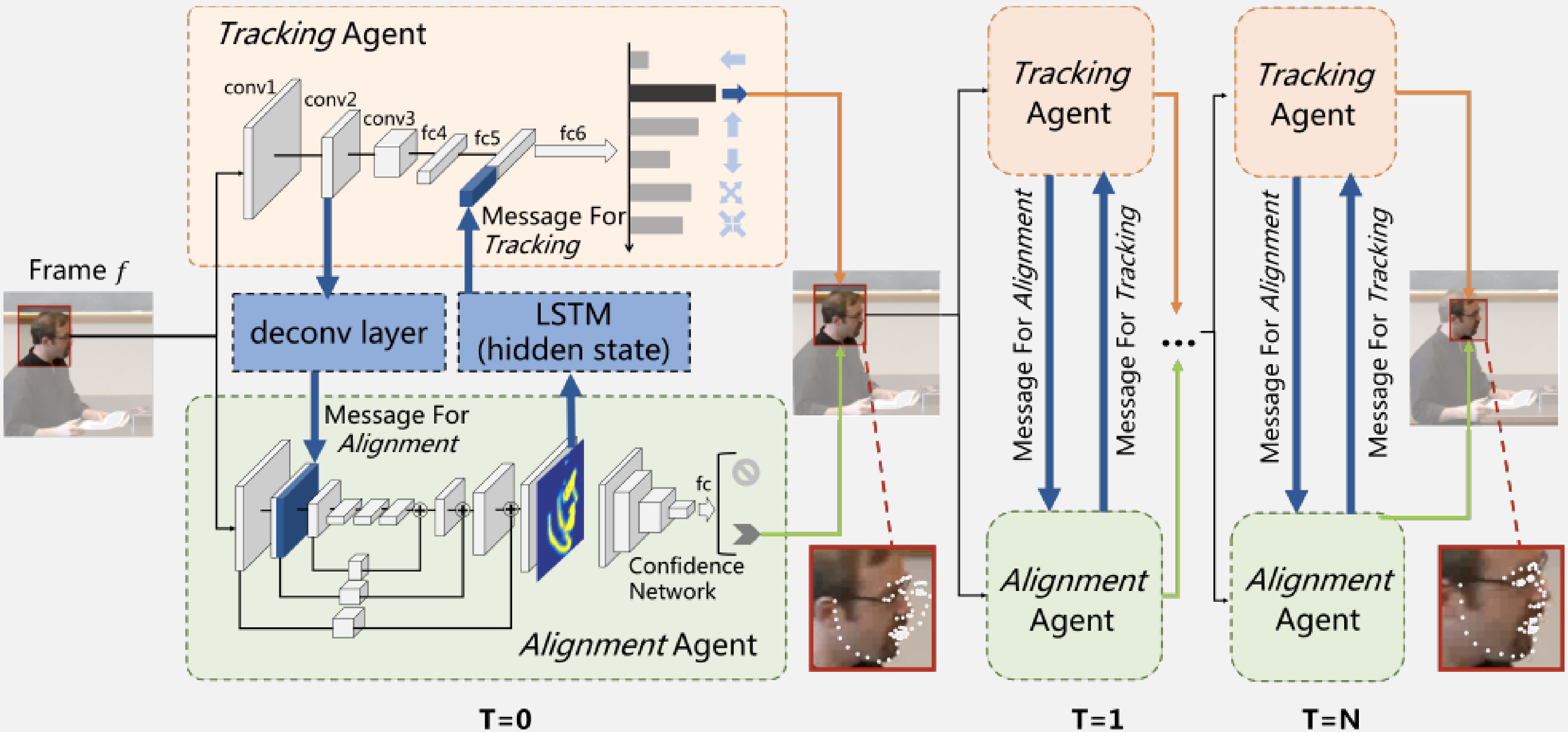
Dual-Agent DRL for Deformable Face Tracking



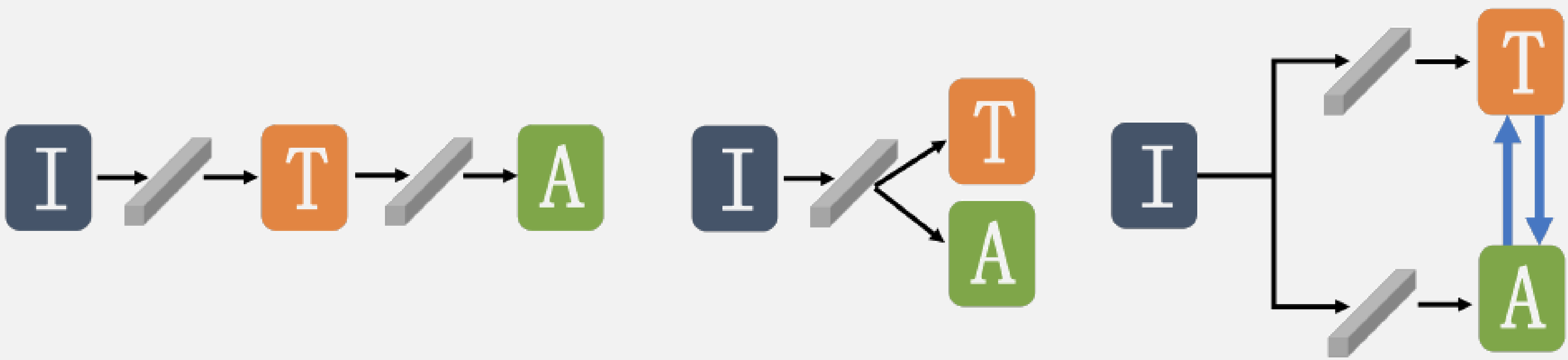
Guo, Minghao, **Jiwen Lu**, and Jie Zhou. "Dual-Agent Deep Reinforcement Learning f
Deformable Face Tracking." ECCV2018

Dual-Agent DRL for Deformable Face Tracking

Approach



Dual-Agent DRL for Deformable Face Tracking

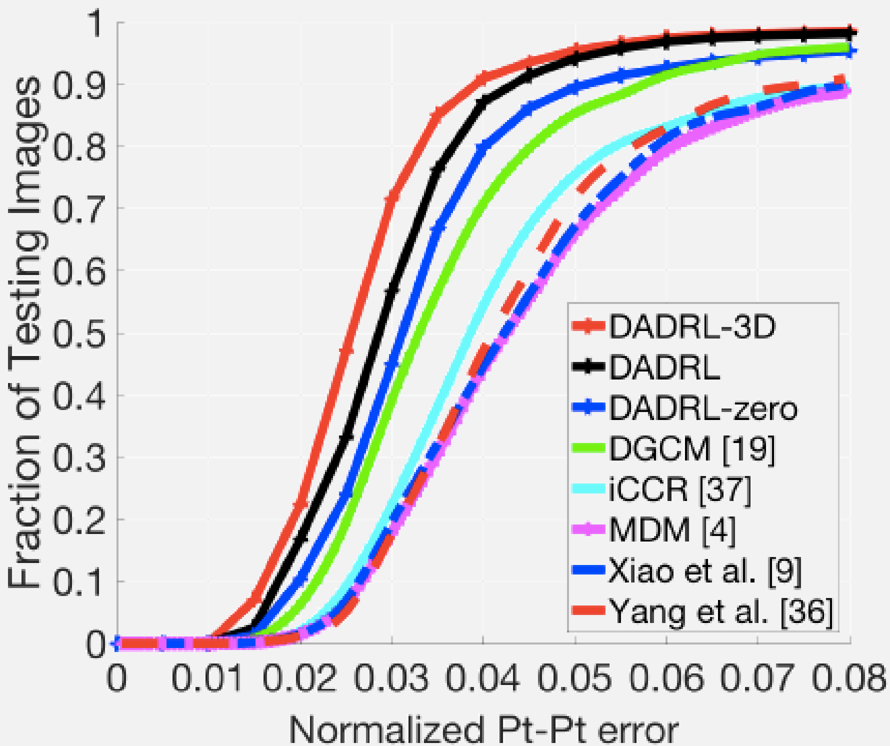


(a) Tracking-by-detection (b) Multi-task learning (c) DADRL

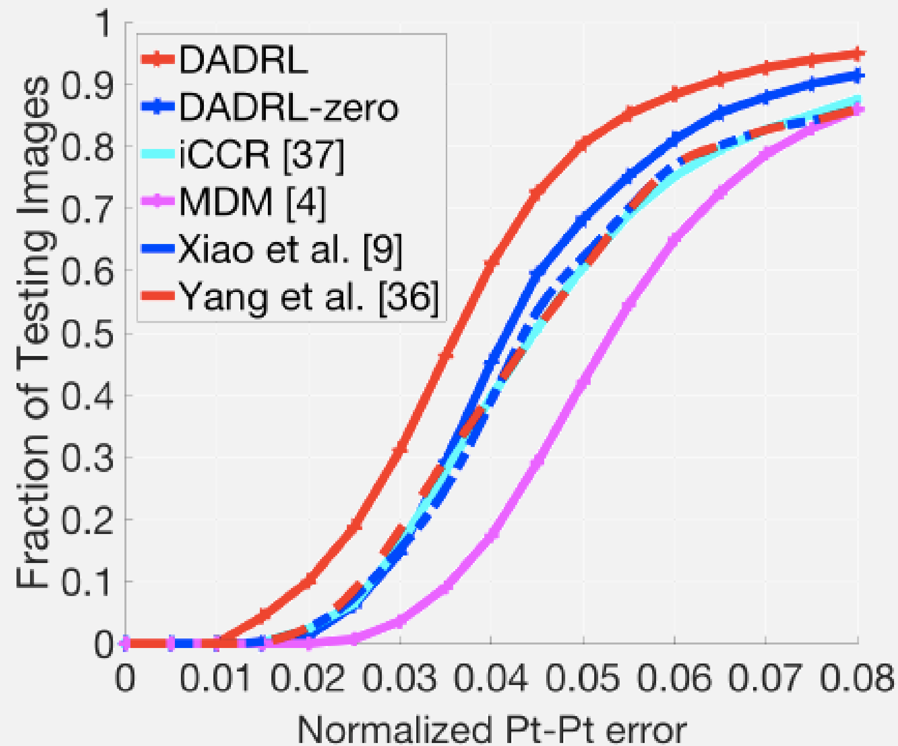
- No guarantee to hold the probabilistic duality.
- Assumes that two task share the same input space, which is too strong in many real applications.
- Explicitly exploits the synergy between these two tasks.

Dual-Agent DRL for Deformable Face Tracking

Experimental Results on the 300VW:

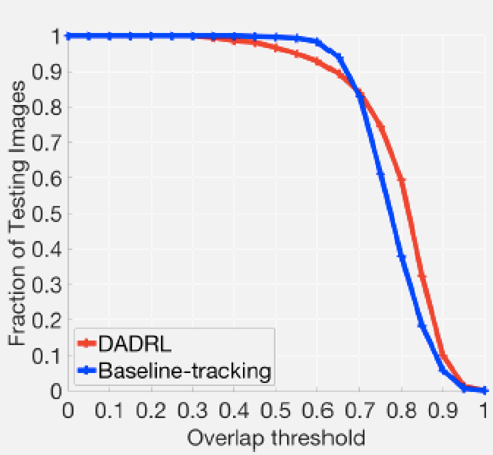


(a)

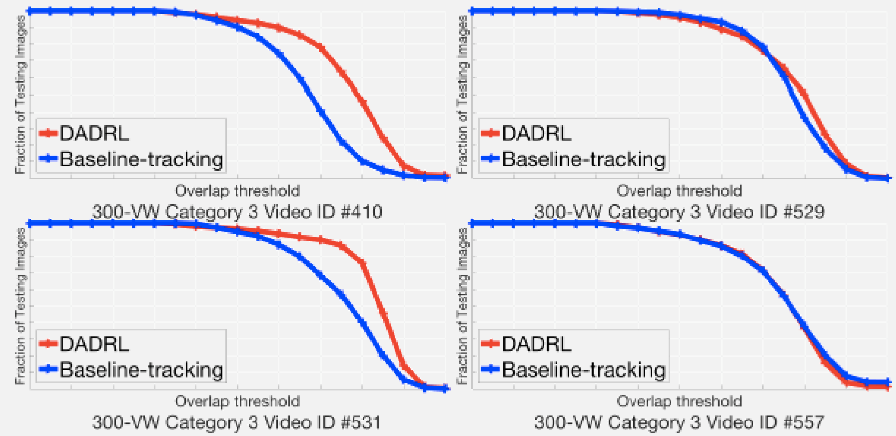


(b)

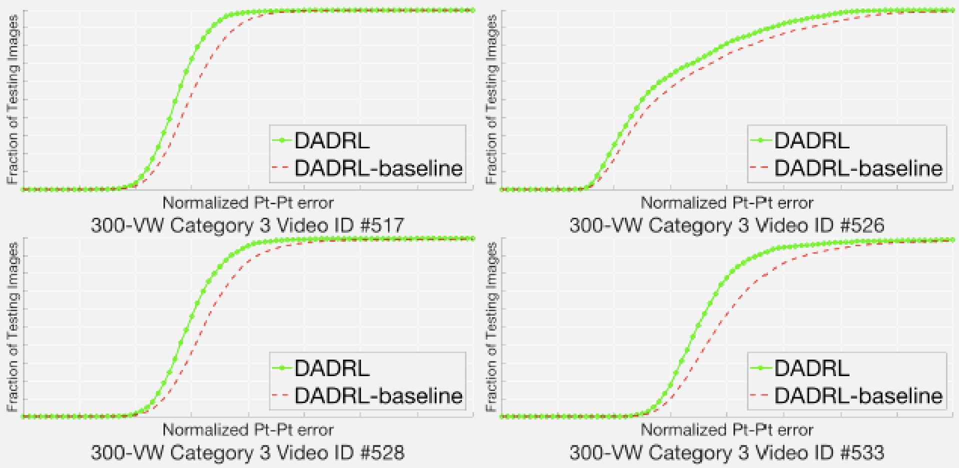
Dual-Agent DRL for Deformable Face Tracking



(a)



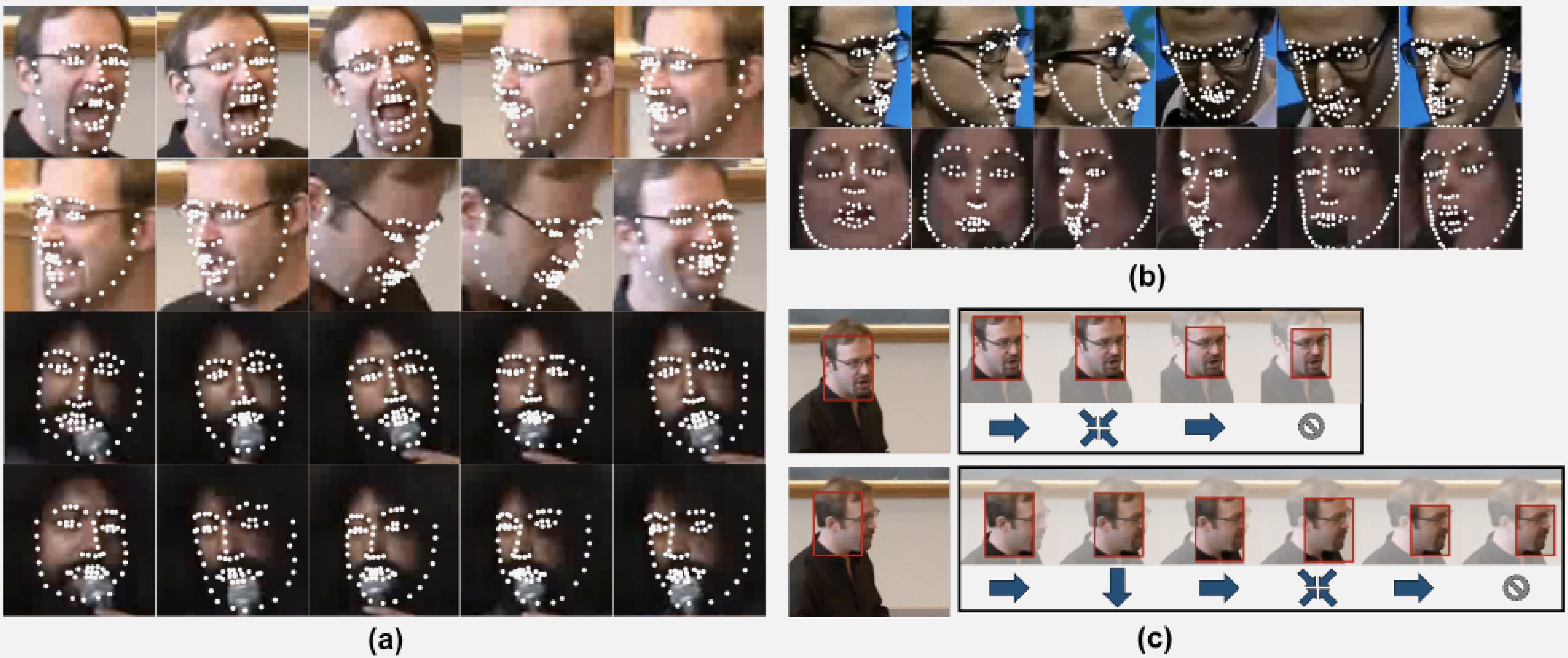
(b)



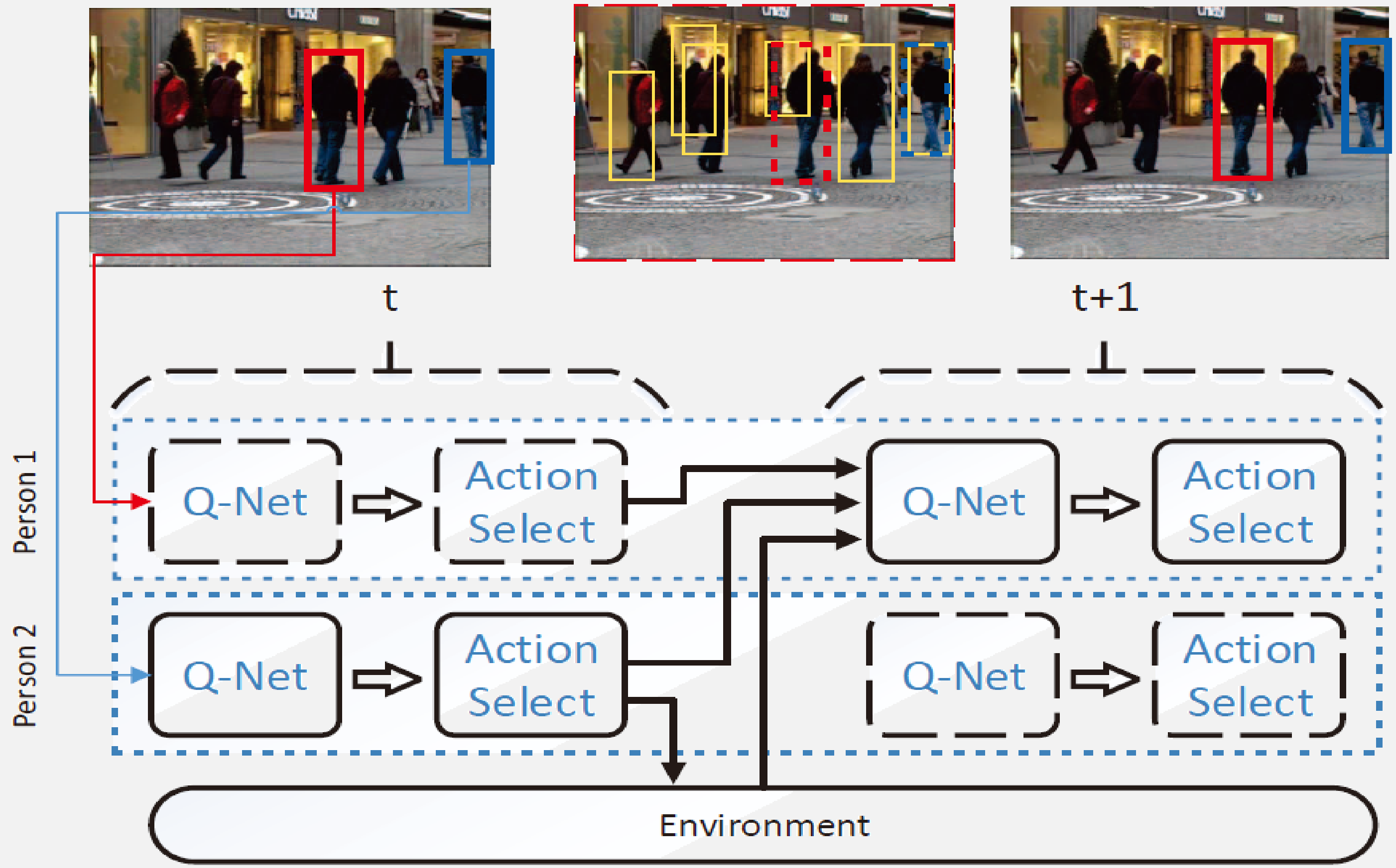
Vi./Meth.	Baseline	DADRL
#517	3.29	**2.75**
#526	3.60	**2.92**
#528	3.70	**3.01**
#533	3.78	**3.68**

Dual-Agent DRL for Deformable Face Tracking

Experimental Results



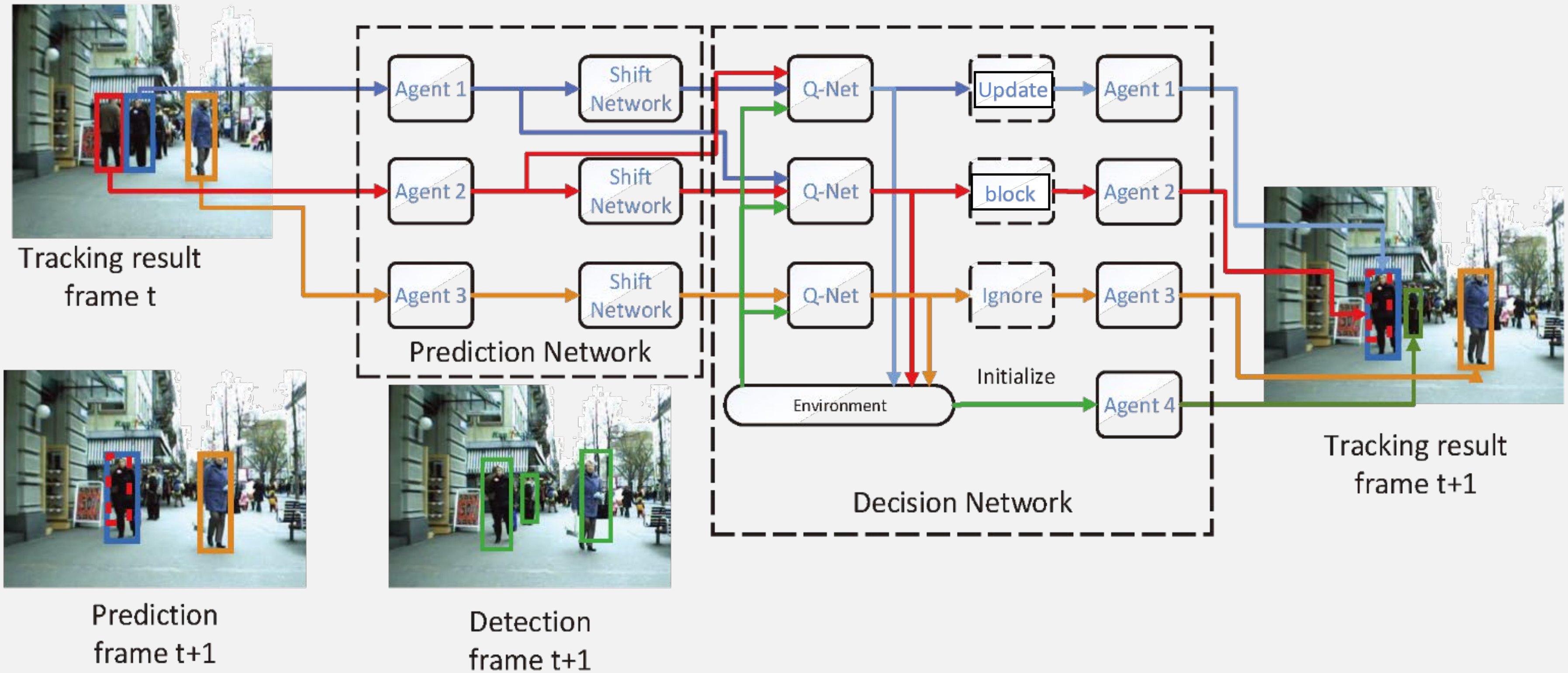
Collaborative DRL for Multi-Object Tracking



Liangliang Ren, **Jiwen Lu**, Zifeng Wang, Qi Tian, and Jie Zhou. "Collaborative Deep Reinforcement Learning for Multi-object Tracking." ECCV2018.

Collaborative DRL for Multi-Object Tracking

Network Architecture



Prediction Network：reduce the influence of detection

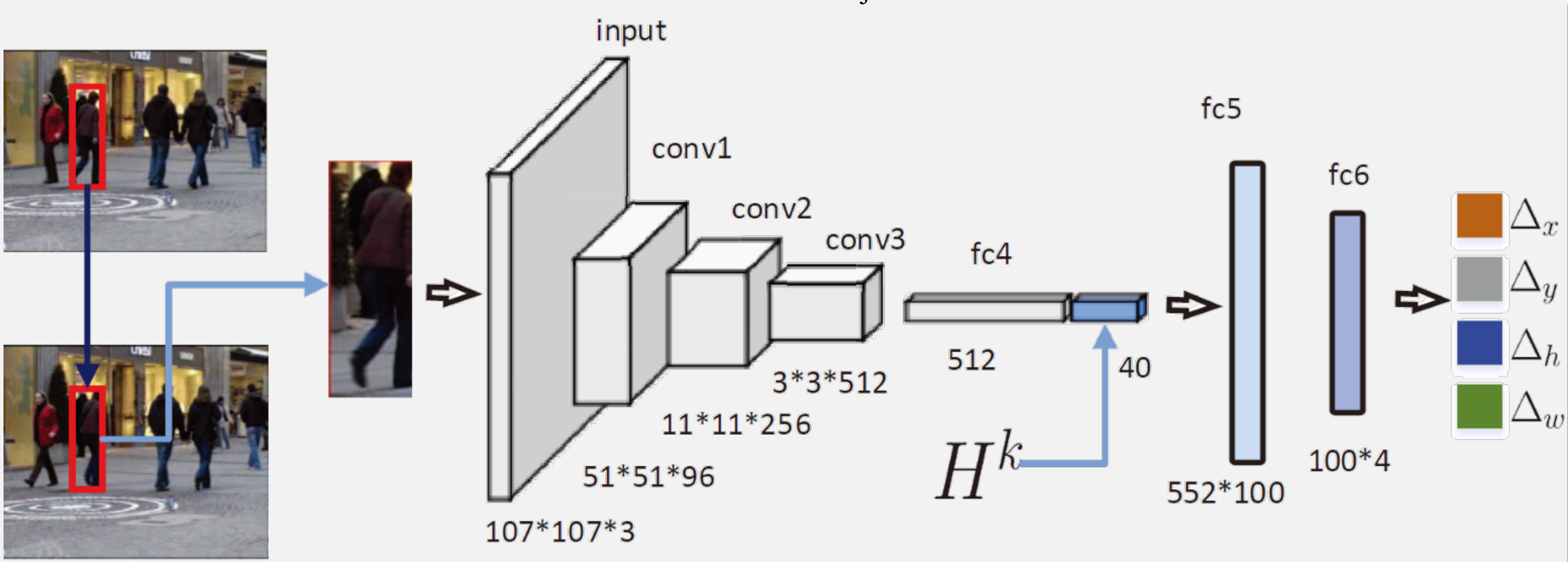
Information Interaction：reduce the influence of occulution

Collaborative DRL for Multi-Object Tracking

Prediction Network:

$$\arg\max_{\phi} J(\phi) = \sum_{i,t} \sum_{b \in B_{i,t}} g(b^{*}_{i,t+1}, b + \phi(I_t, b, H_t))$$

$$g(b_i, b_j) = \frac{b_i \cap b_j}{b_i \cup b_j}$$



Collaborative DRL for Multi-Object Tracking

Actions:

$$A = \{update; ignore; block; delete\}$$

Reward:

$$r_{i,t} = \begin{cases} 1 & \text{if } IoU \geq 0.7 \\ 0 & \text{if } 0.5 \leq IoU \leq 0.7 \\ -1 & \text{else} \end{cases}$$

$$r_{delete} = \begin{cases} 1 & \text{if object disappeared} \\ -1 & \text{else} \end{cases}$$

$$r^{*}_{i,t} = r_{i,t} + \beta r_{j,t+1} \qquad Q_{i,t} = r^{*}_{i,t} + \gamma Q_{i,t+1}$$

Formulation:

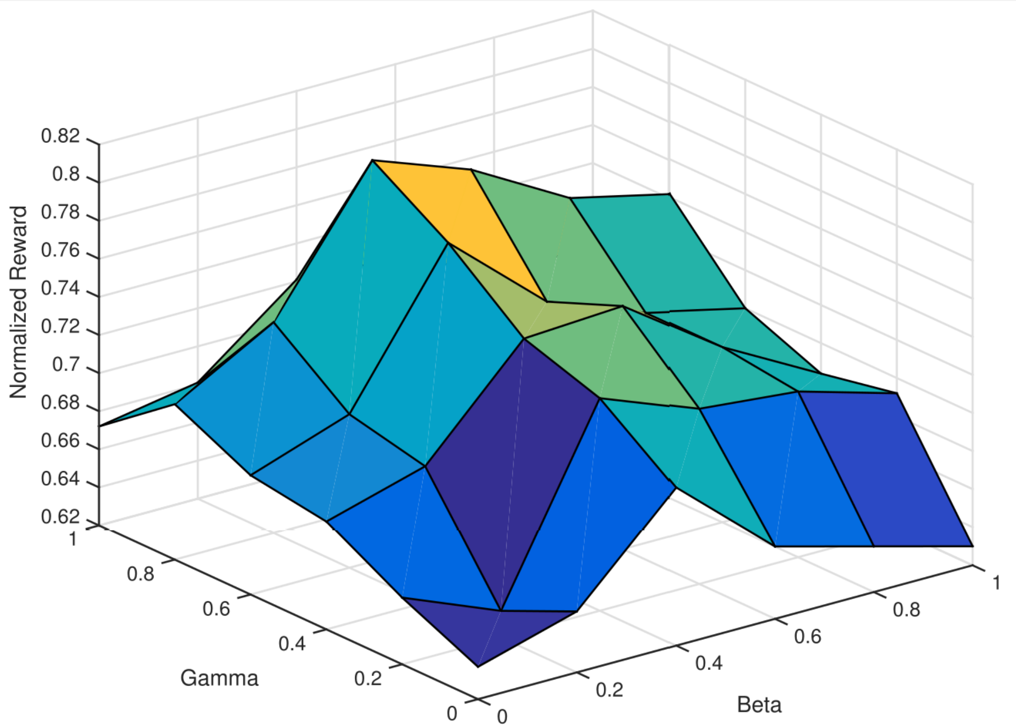
$$\arg\max_{\theta} L(\theta) = E_{s,a} \log(\pi(a|s;\theta)) Q(s;a);$$

Experimental Results on the MOT16 Dataset

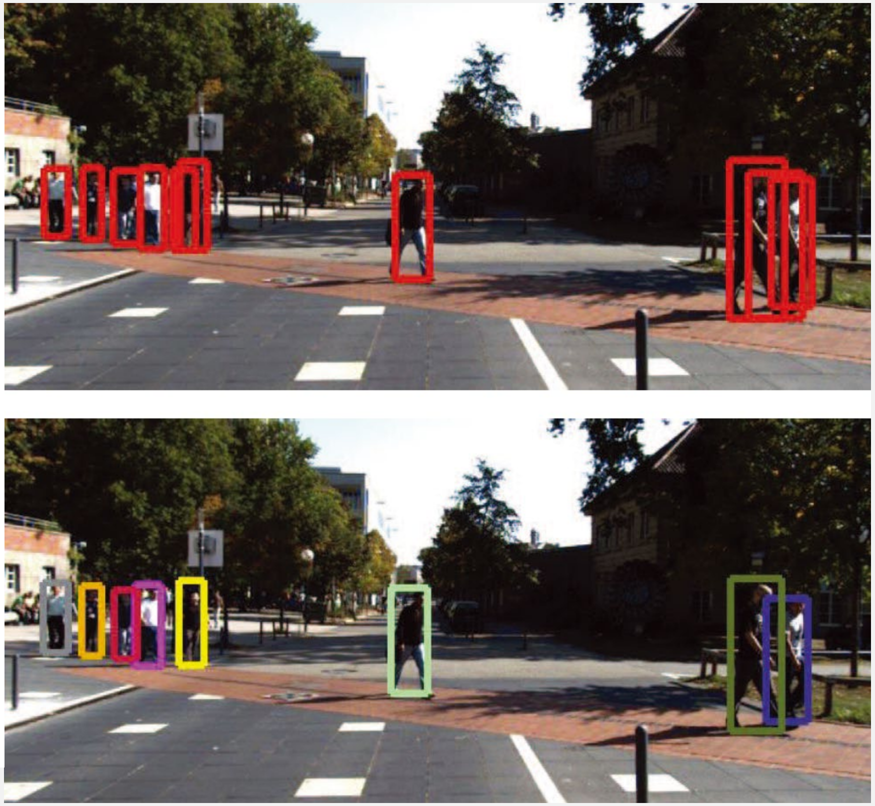
Mode	Method	MOTA↑	MOTP↑	FAF↓	MT(%)↑	ML(%)↓	FP↓	FN↓
Offline	TBD [48]	33.7	76.5	1.0	7.2	54.2	5804	112587
	LTTSC-CRF [49]	37.6	75.9	2.0	9.6	55.2	11969	101343
	LINF1 [42]	41.0	74.8	1.3	11.6	51.3	7896	99224
	MHT_DAM_16 [44]	45.8	76.3	1.1	16.2	43.2	6412	91758
	NOMT [7]	46.4	76.7	1.6	**18.3**	41.4	9753	87565
	NLLMPa [50]	47.6	78.5	1.0	17.0	40.4	5844	89093
	LMP [51]	**48.8**	**79.0**	1.1	18.2	40.1	6654	**86245**
Online	OVBT [52]	38.4	75.4	1.9	7.5	47.3	11517	99463
	EAMTT_pub [53]	38.8	75.1	1.4	7.9	49.1	8114	102452
	CDA_DDALv2 [47]	43.9	74.7	1.1	10.7	44.4	6450	95175
	AMIR [9]	47.2	75.8	**0.5**	14.0	41.6	**2681**	92856
	Ours	47.3	74.6	1.1	17.4	**39.9**	6375	88543

Collaborative DRL for Multi-Object Tracking

Ablation studies:

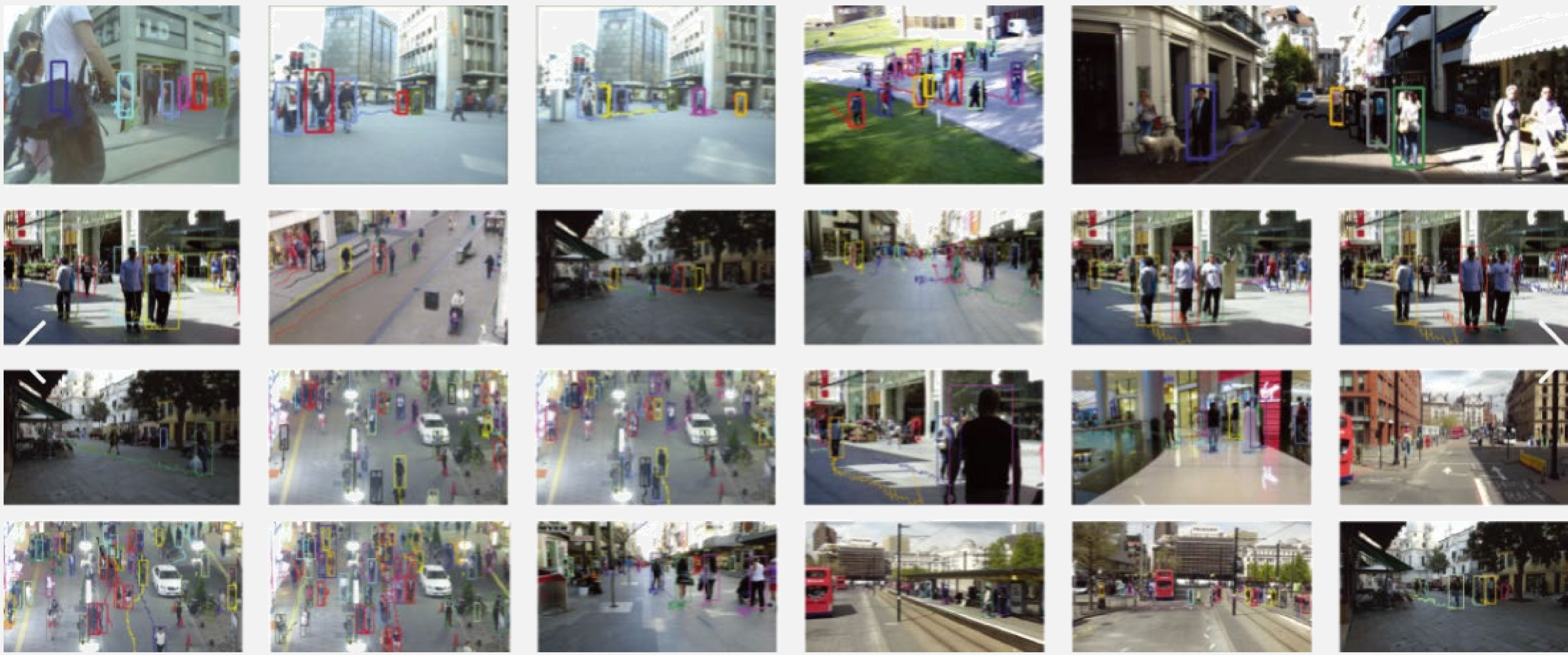


The average normalized rewards versus different β and γ.



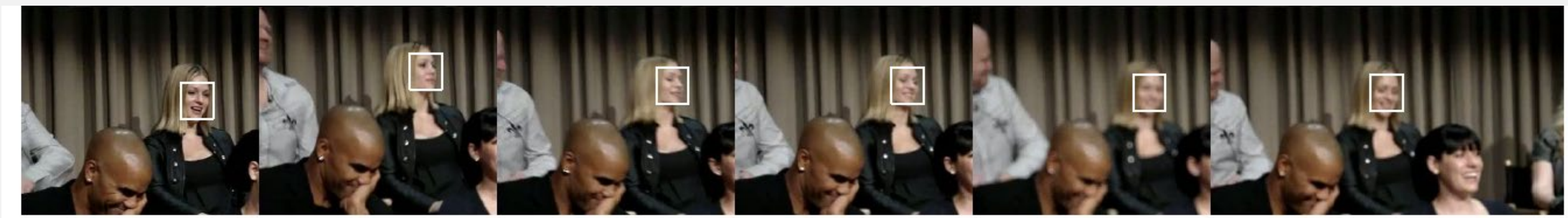
False positives eliminating

Experimental Results

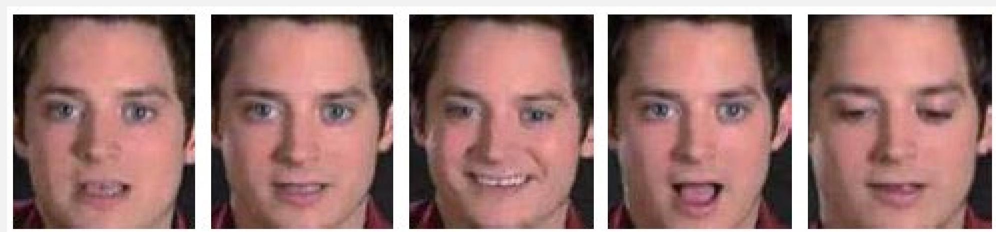


Attention-aware DRL for video based face recognition

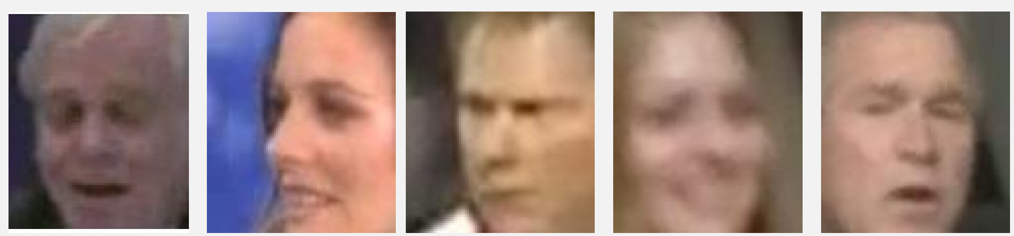
Video Face Recognition:



➢**Redundancy**

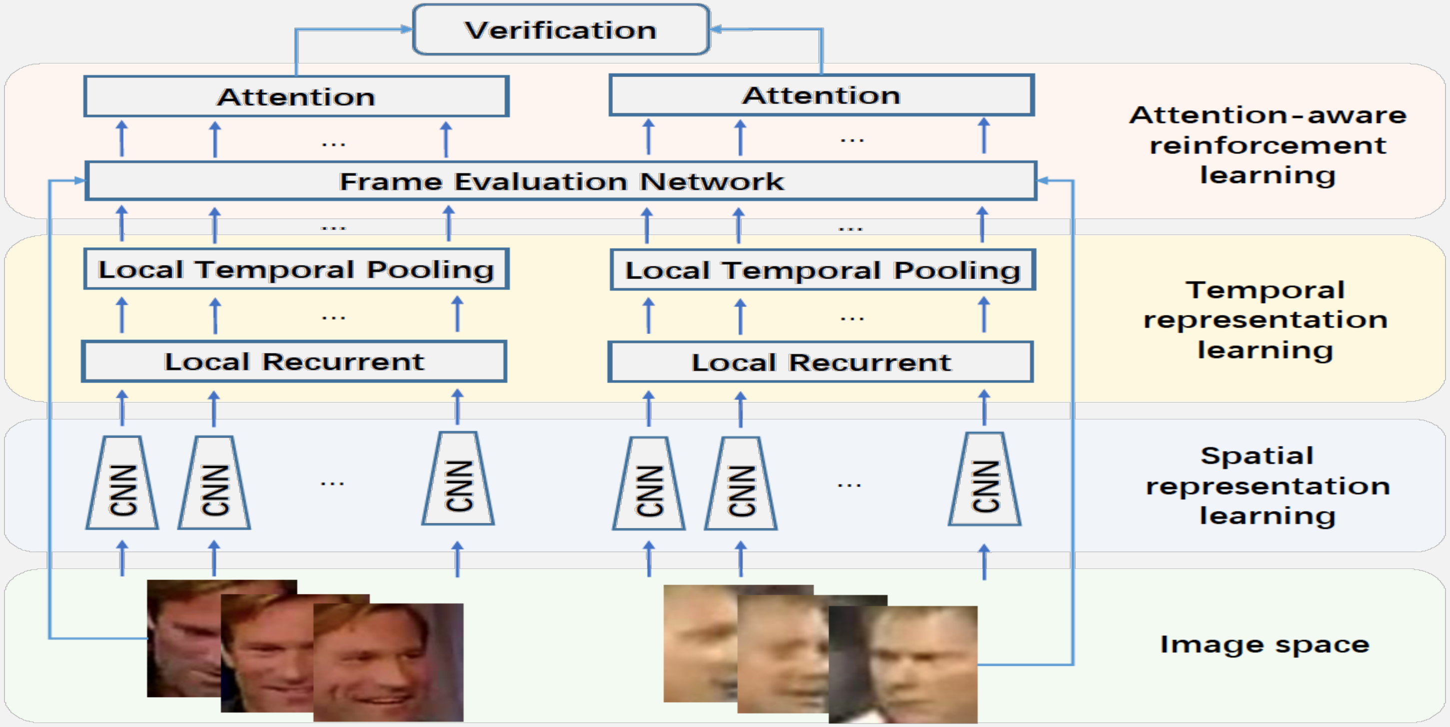


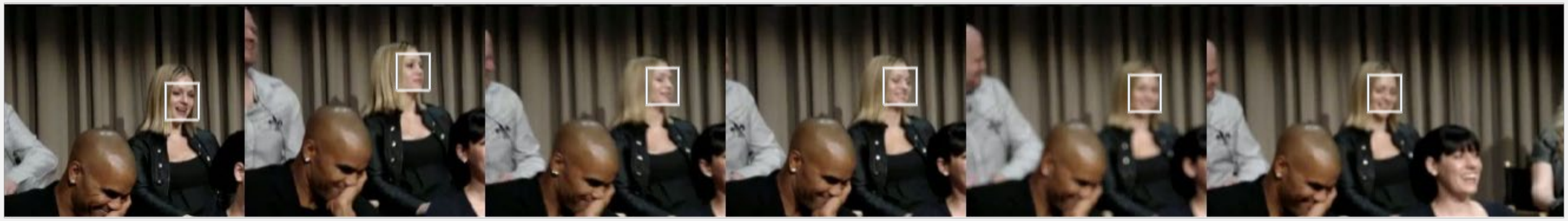
➢**Noisy**



Yongming Rao, Jiwen Lu, and Jie Zhou. "Attention-aware deep reinforcement learning for video face recognition." ICCV2017.

Attention-aware DRL for video based face recognition

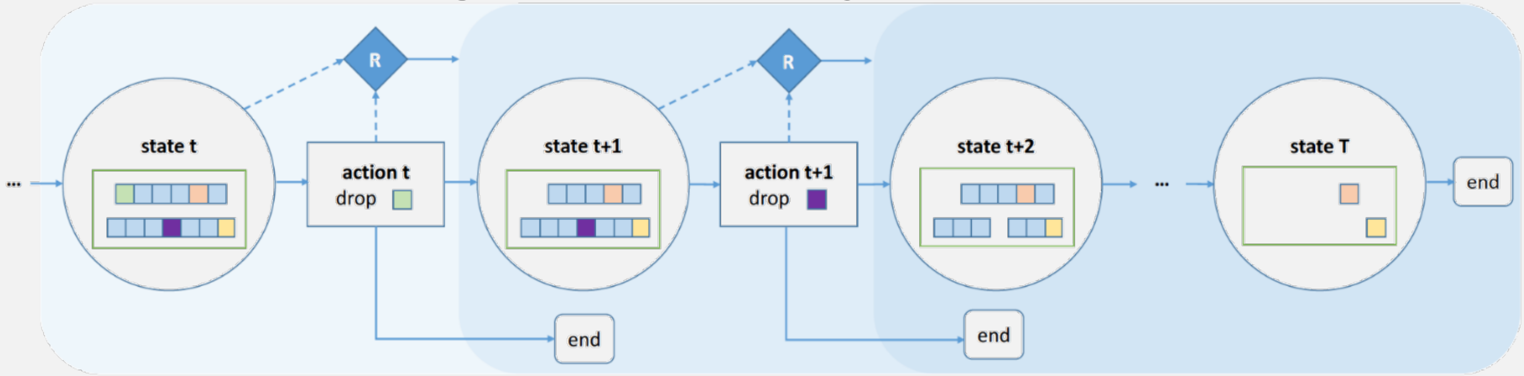




Attention-aware DRL for video based face recognition

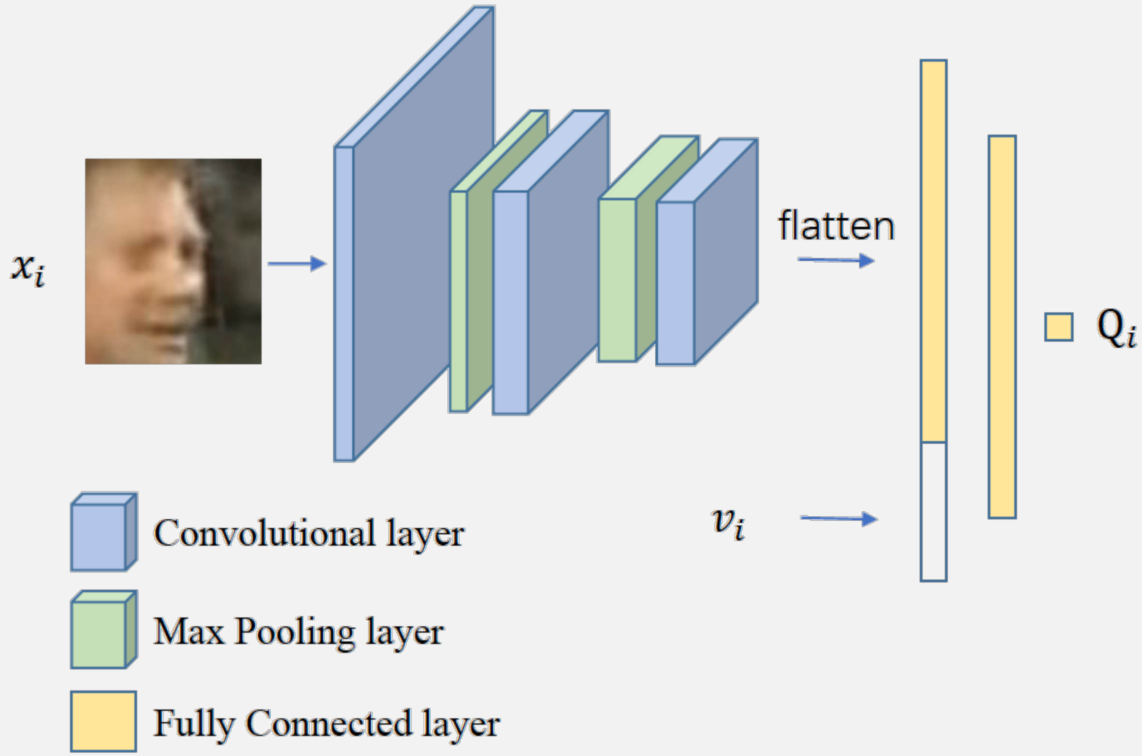
Approach

- Finding key information in the video by attention model
- Train an agent to imitate human actions to find key information
  - Markov Decision Model： deleting frames progressively
    - State: current frames
    - Action: delete one frame or stop
    - Reward: recognition accuracy （No extra label data required）



Attention-aware DRL for video based face recognition

Attention Agent：Frame Evaluation Network



Attention-aware DRL for video based face recognition

Experimental results:

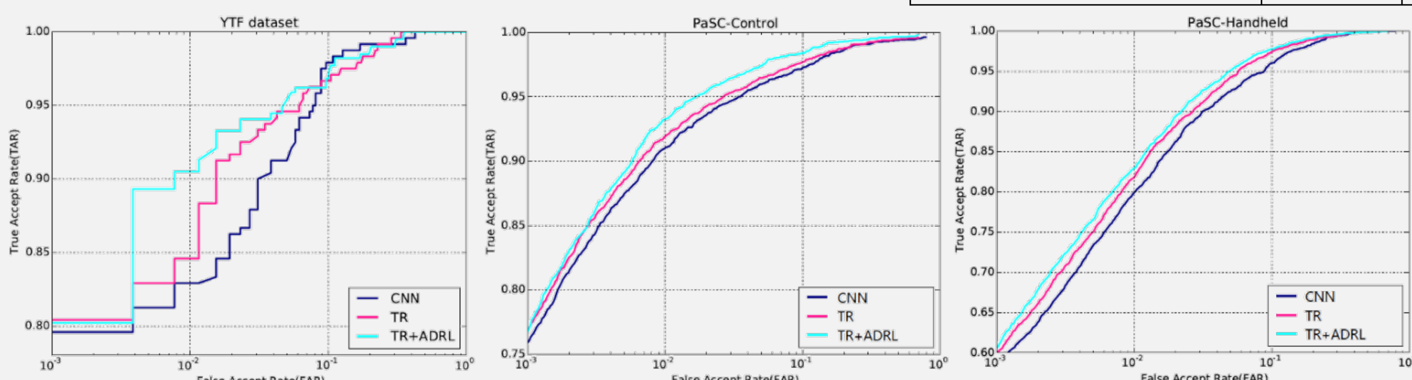
YTF:

Method	Accuracy	Year
LM3L [16]	81.3 ± 1.2	2014
DDML [15]	82.3 ± 1.2	2014
EigenPEP [24]	84.8 ± 1.4	2014
DeepFace-single [35]	91.4 ± 1.1	2015
DeepID2+ [34]	93.2 ± 0.2	2015
FaceNet [32]	95.12 ± 0.39	2015
Deep FR [31]	97.3	2015
NAN [43]	95.72 ± 0.64	2016
Wen *et al.* [40]	94.9	2016
TBE-CNN [10]	94.96 ± 0.31	2017
ADRL	95.96 ± 0.59	
ADRL-finetune	96.52 ± 0.54	

PaSC:

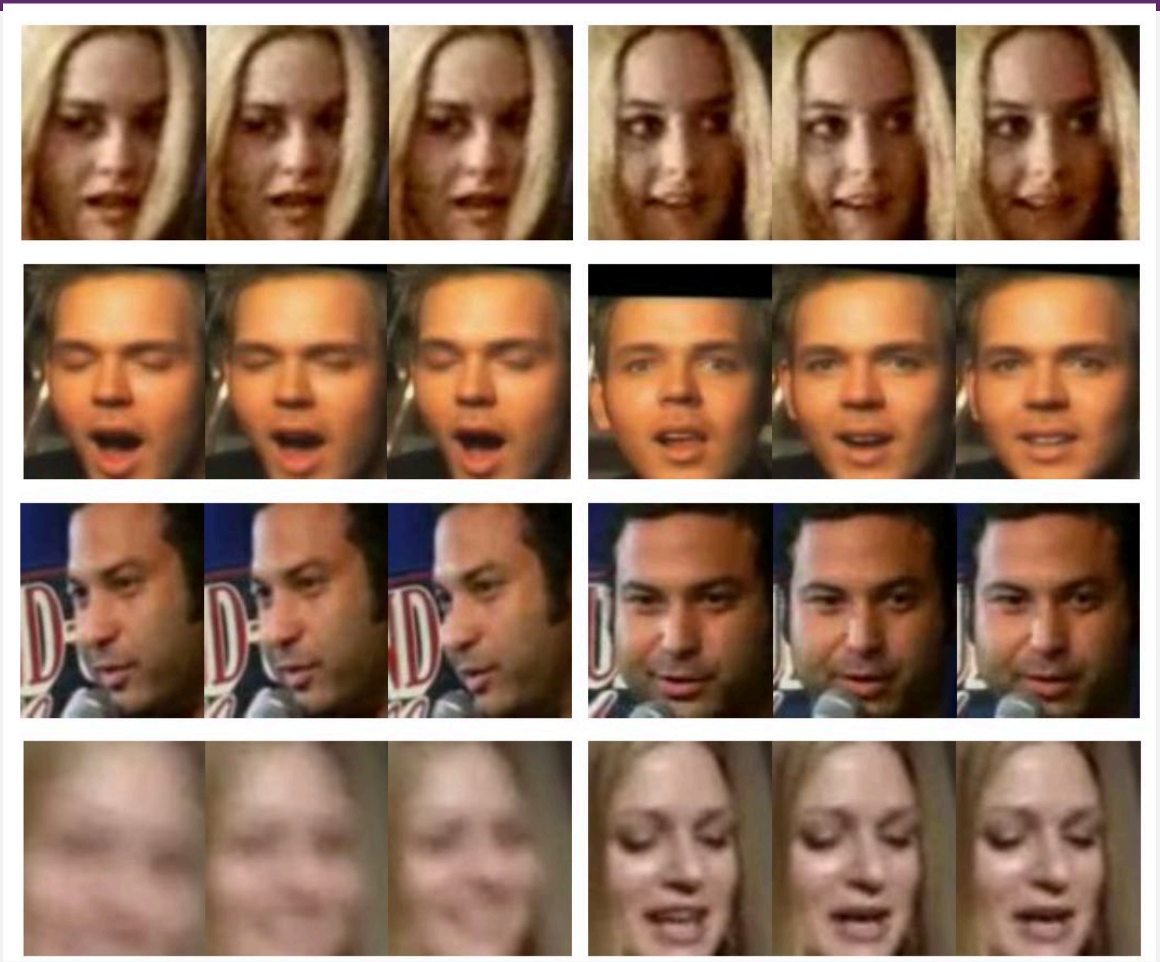
Method	Control	Handheld
PittPatt	48.00	38.00
DeepO2P [22]	68.76	60.14
VGGFace	78.82	68.24
SPDNet [18]	80.12	72.83
GrNet [21]	80.52	72.76
TBE-CNN [10]	97.80	96.12
Ours-CNN	91.02	79.91
Ours-CNN (finetuned)	93.76	91.34
Ours-TR	91.92	82.43
Ours-ADRL	93.13	83.69
Ours-ADRL (finetuned)	95.67	93.78

ROC:



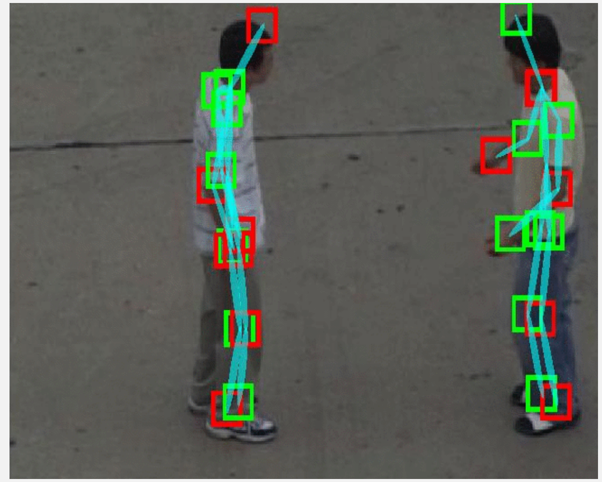
(a) (b)

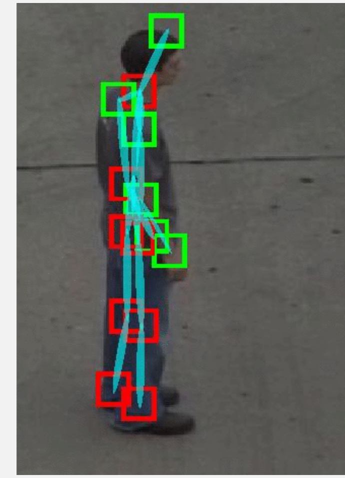
Attention-aware DRL for video based face recognition

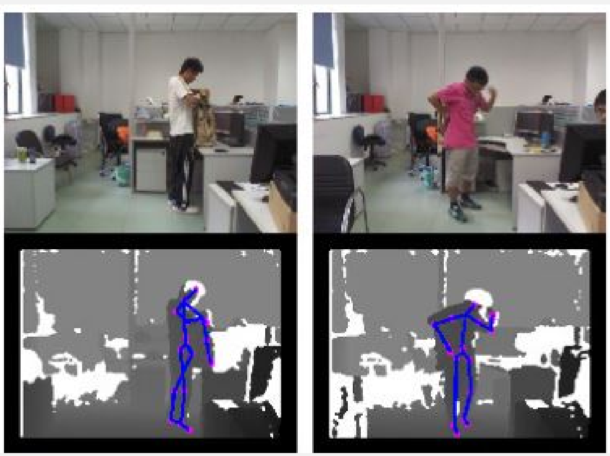


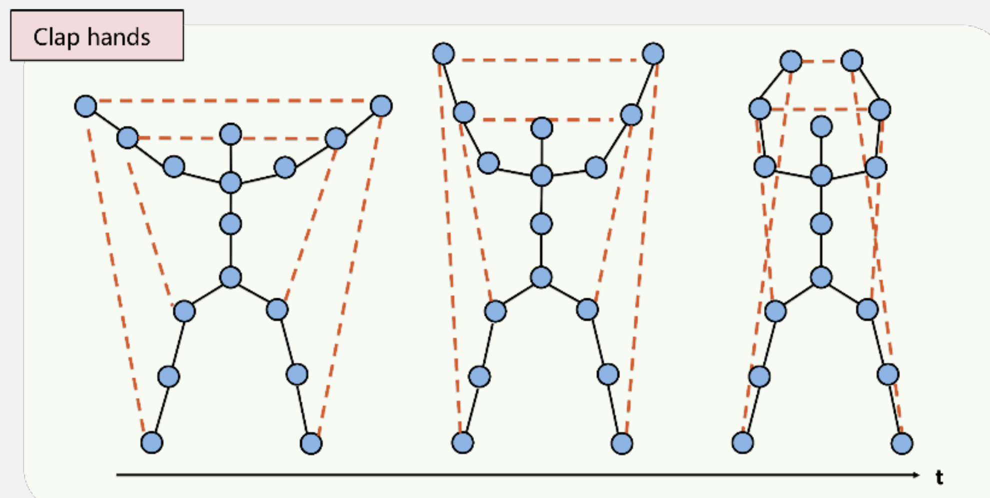
Frames with: low scores and high scores

Action Detection, Recognition, and Prediction



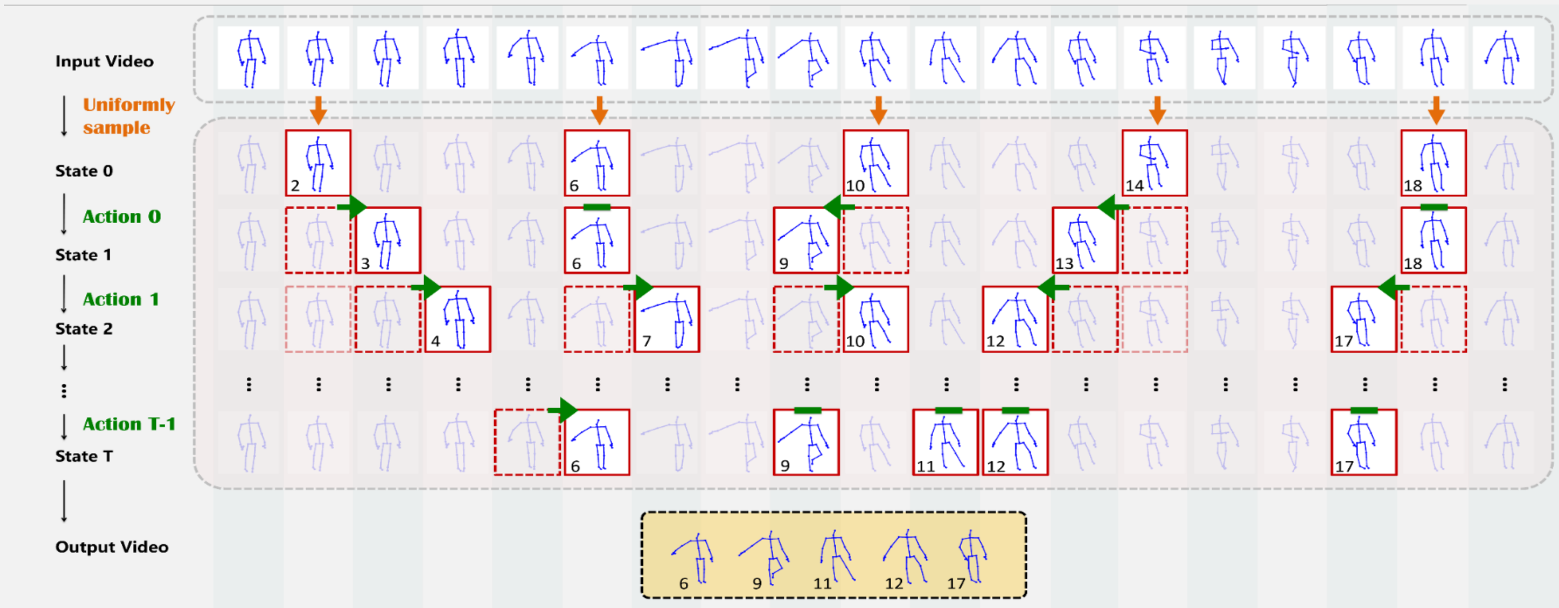






DRL for Skeleton-based Action Recognition

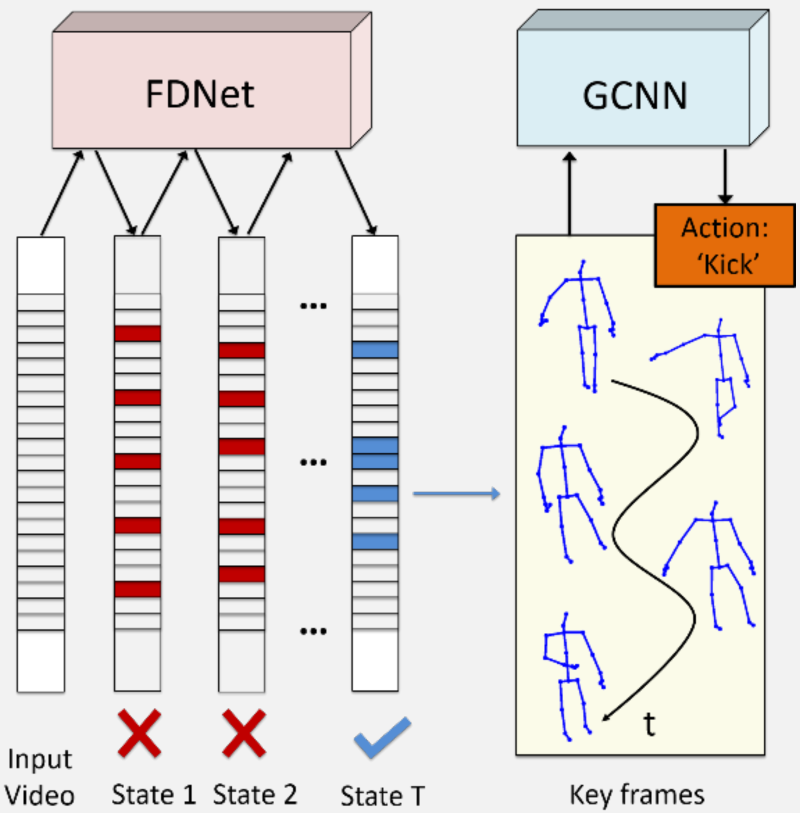
- Temporal Domain: distil the most informative frames by DRL.
- Spatial Domain: capture the dependency between the joints by graph convolutional neural network.



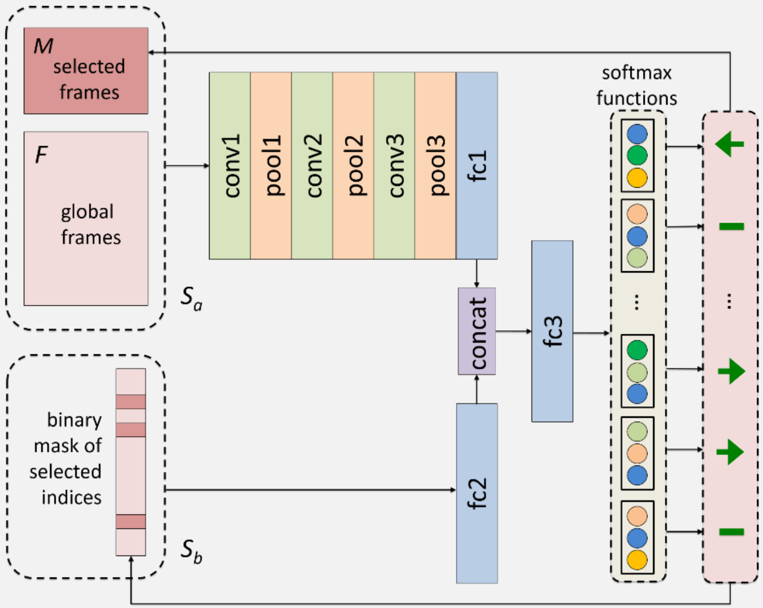
Tang, Yansong, Yi Tian, **Jiwen Lu**, Peiyang Li, and Jie Zhou. "Deep Progressive Reinforcement Learning for Skeleton-Based Action Recognition." CVPR2018

DRL for Skeleton-based Action Recognition

Framework:



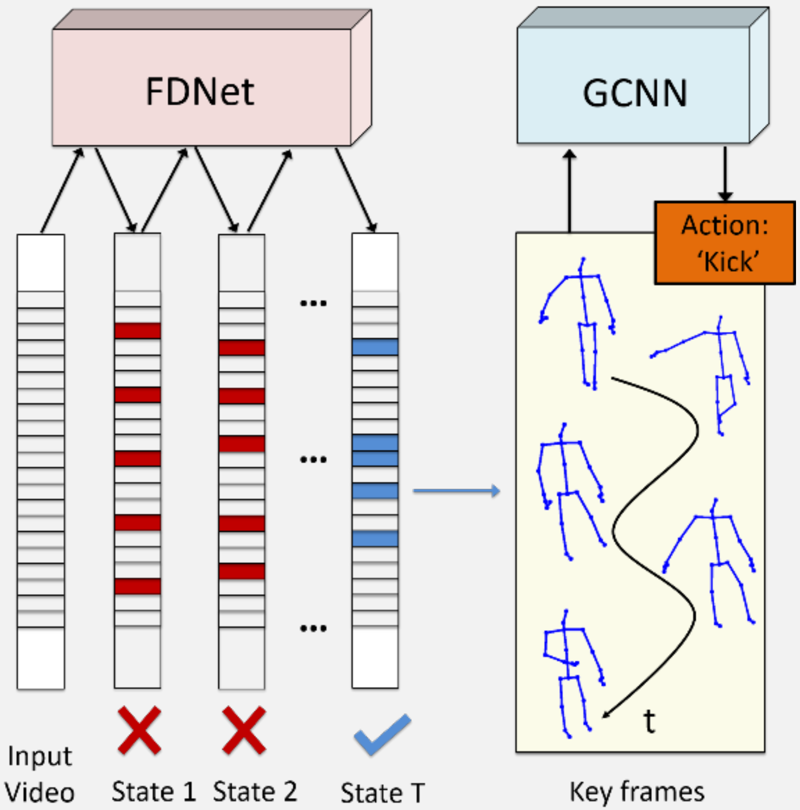
Frame Distillation Network (FDNet)：



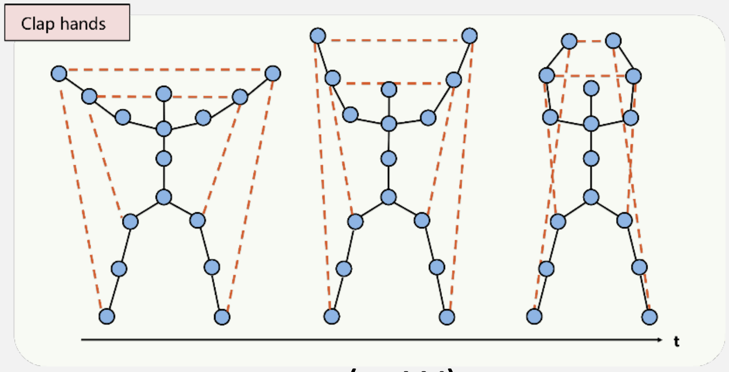
- States： Selected and Global Frames, and their relationship
- Actions： adjustment direction of each selected frame
- Rewards： influence on recognition

DRL for Skeleton-based Action Recognition

Framework:



Graph Convolutional Neural Network(GCNN)：



g(x, W)

$$w_{ij} = \begin{cases} 0, & \text{if } i = j \\ \alpha, & \text{if joint } i \text{ and joint } j \text{ are connected} \\ \beta, & \text{if joint } i \text{ and joint } j \text{ are disconnected} \end{cases}$$

DRL for Skeleton-based Action Recognition

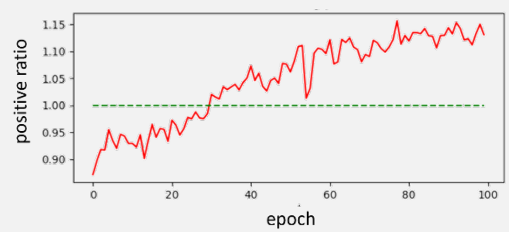
Experimental results:

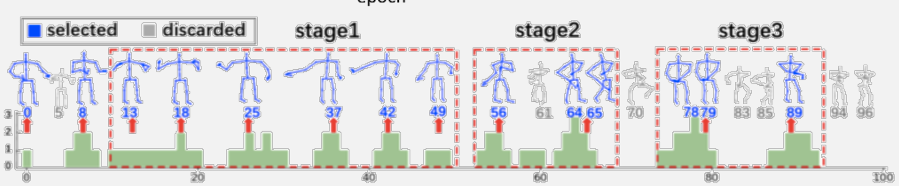
Action recognition accuracy (%) on NTU-RGBD dataset：

Method	CS	CV	Year
Skeleton Quads [10]	38.6	41.4	2014
Lie Group [44]	50.1	52.8	2014
Dynamic Skeletons [20]	60.2	65.2	2015
HBRNN-L [9]	59.1	64.0	2015
Part-aware LSTM [38]	62.9	70.3	2016
ST-LSTM + Trust Gate [31]	69.2	77.7	2016
STA-LSTM [42]	73.4	81.2	2017
LieNet-3Blocks [21]	61.4	67.0	2017
Two-Stream RNN [46]	71.3	79.5	2017
Clips + CNN + MTLN [25]	79.6	84.8	2017
VA-LSTM [55]	79.2	87.7	2017
View invariant [33]	80.0	87.2	2017
Two-Stream CNN [29]	83.2	89.3	2017
LSTM-CNN [28]	82.9	91.0	2017
Ours-CNN	79.7	84.9	
Ours-DPRL	82.3	87.7	
Ours-DPRL+graph[1]	82.5	88.1	
Ours-DPRL+graph[2]	82.8	88.9	
Ours-DPRL+graph	**83.5**	**89.8**	

Action recognition accuracy (%) on SYSU dataset

Method	Acc.	Year
LAFF(SKL) [19]	54.2	2016
Dynamic Skeletons [20]	75.5	2015
ST-LSTM(Tree) [31]	73.4	2017
ST-LSTM(Tree) + Trust Gate [31]	76.5	2017
Ours-CNN	75.5	
Ours-DPRL	76.7	
Ours-DPRL+graph	**76.9**	

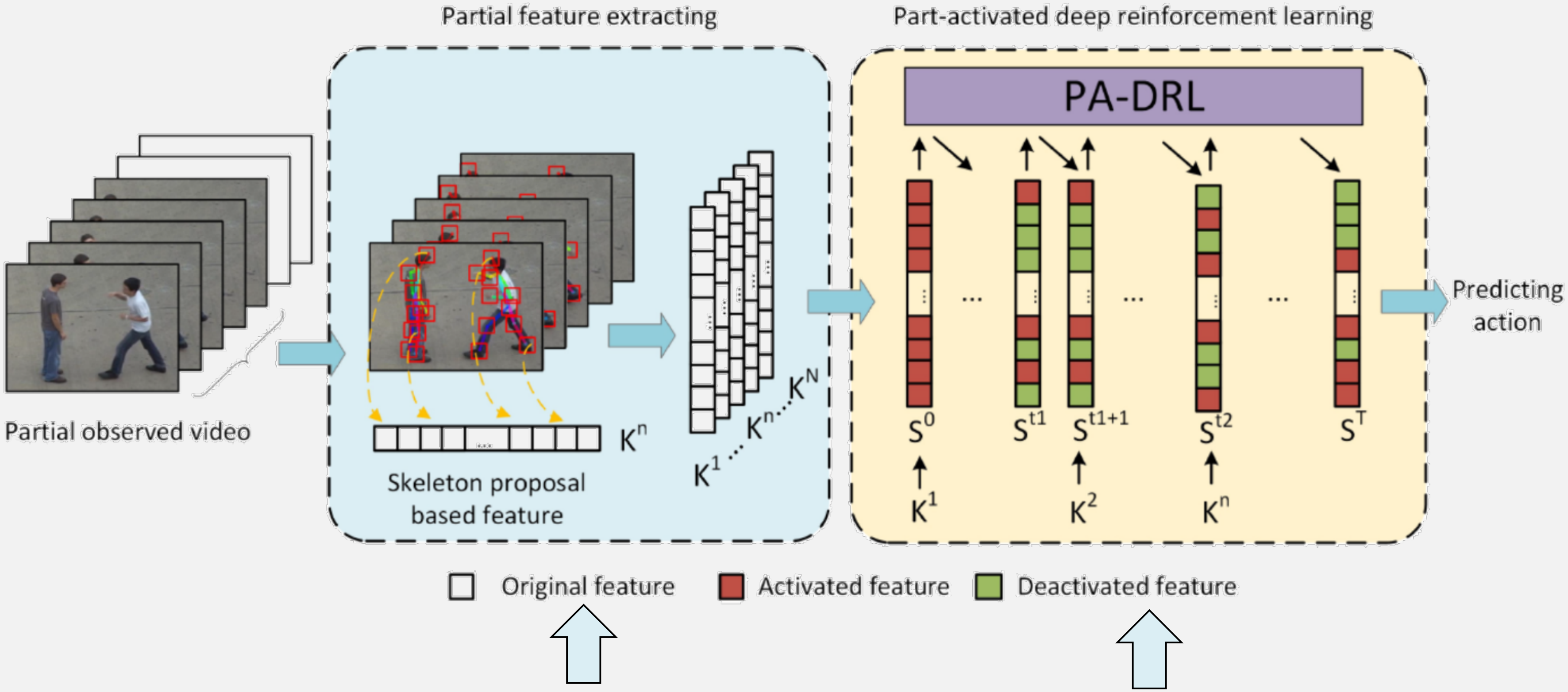




(a) take off jacket

Visualization Results on the selected frames

Part-Activated DRL for Action Prediction



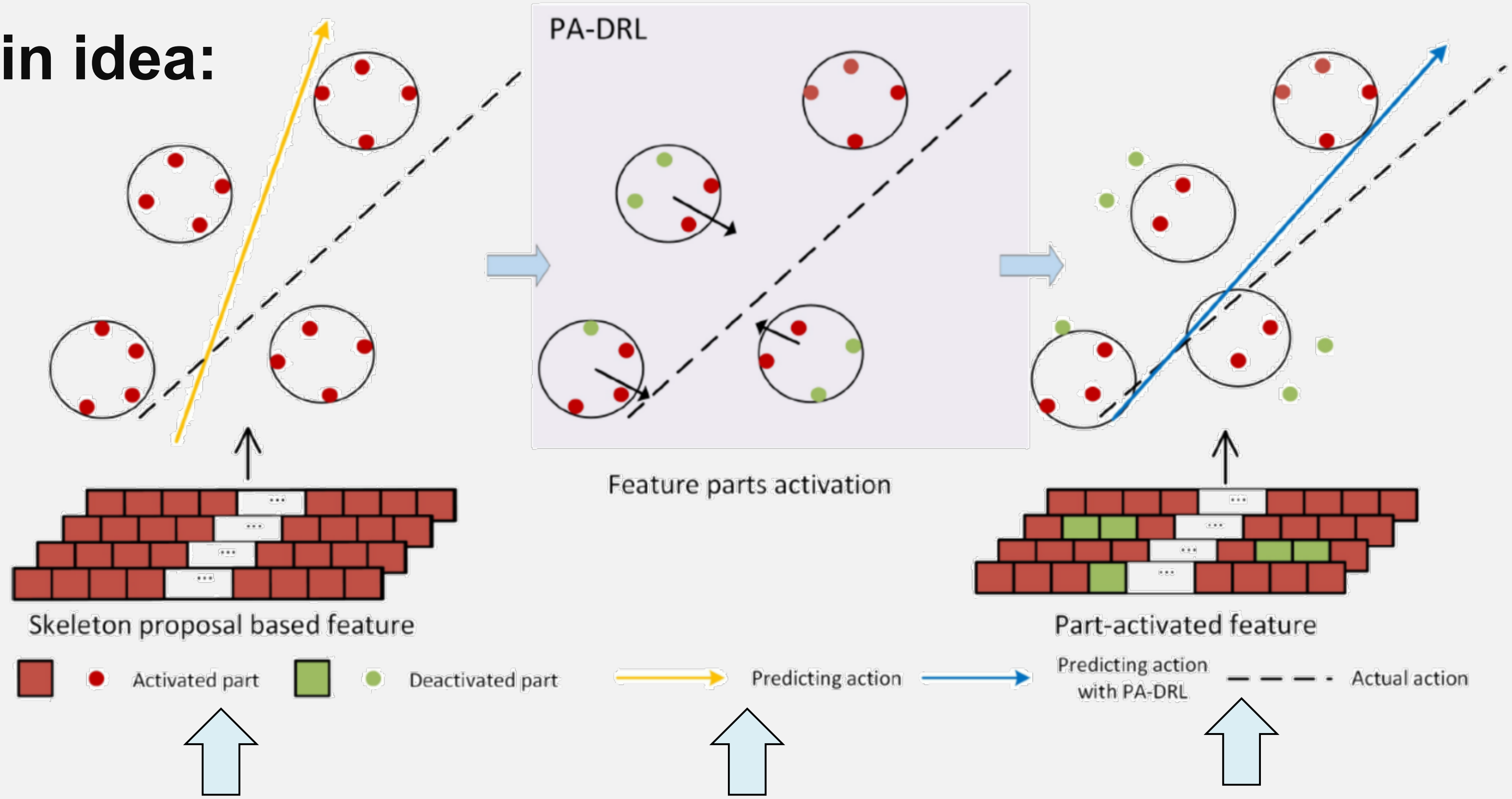
Local Feature Extraction

Activated Features Choosing

Chen Lei, **Jiwen Lu**, Zhanjie Song, and Jie Zhou. "Part-Activated Deep Reinforcement Learning for Action Prediction." ECCV2018.

Part-Activated DRL for Action Prediction

Main idea:



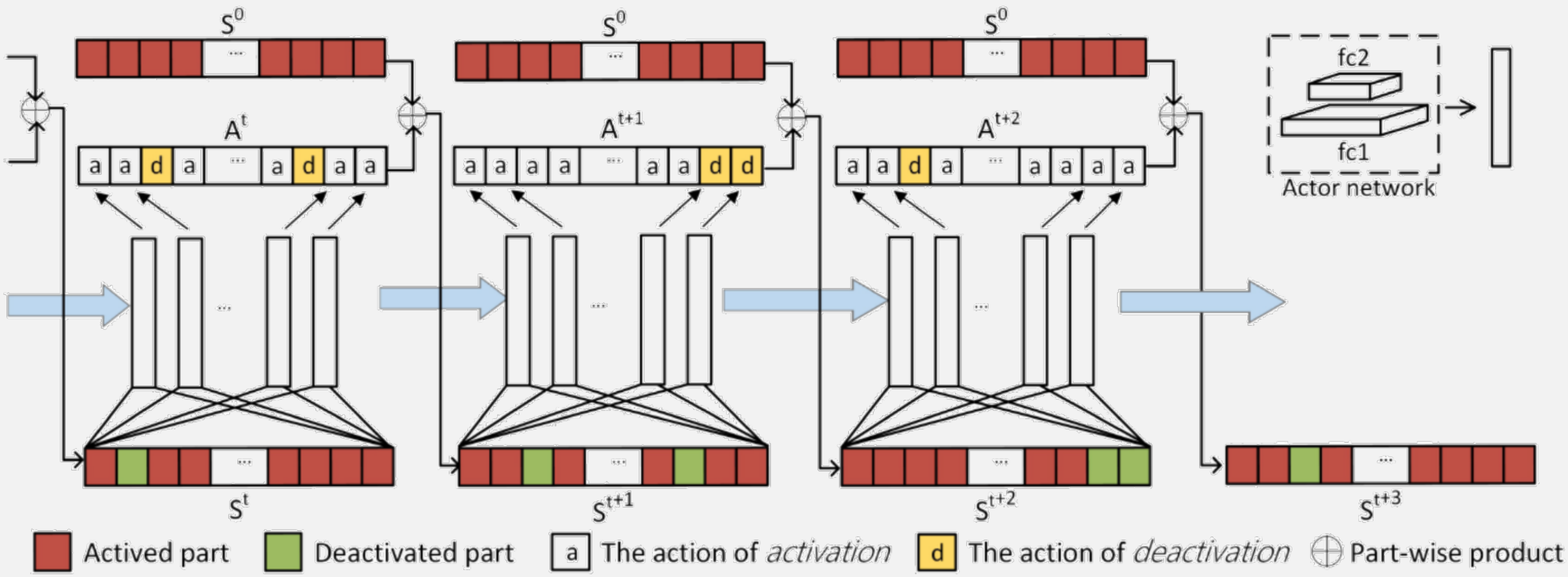
Some feature may influence the final prediction result.

To restrain the irrelevant features by PA-DRL.

The activated part can predict the action more accurately.

Part-Activated DRL for Action Prediction

State transition：



State

$$S_w^t = \Gamma_{u \in U}\left(\beta_{u,w}^t\right)$$

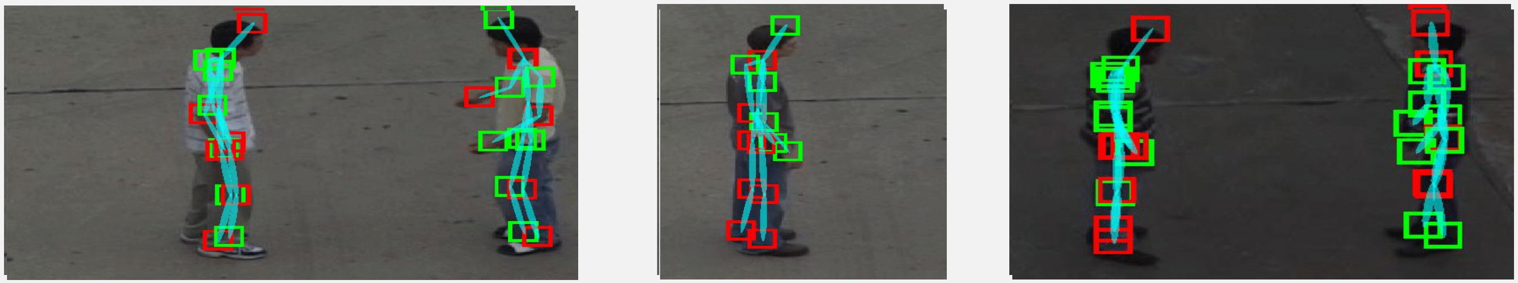
State Transition

$$A_w^{t-1} = \Pi_\theta(S_w^{t-1}),$$
$$S_w^t = S_w^0 \odot A_w^{t-1}$$

Final Reward

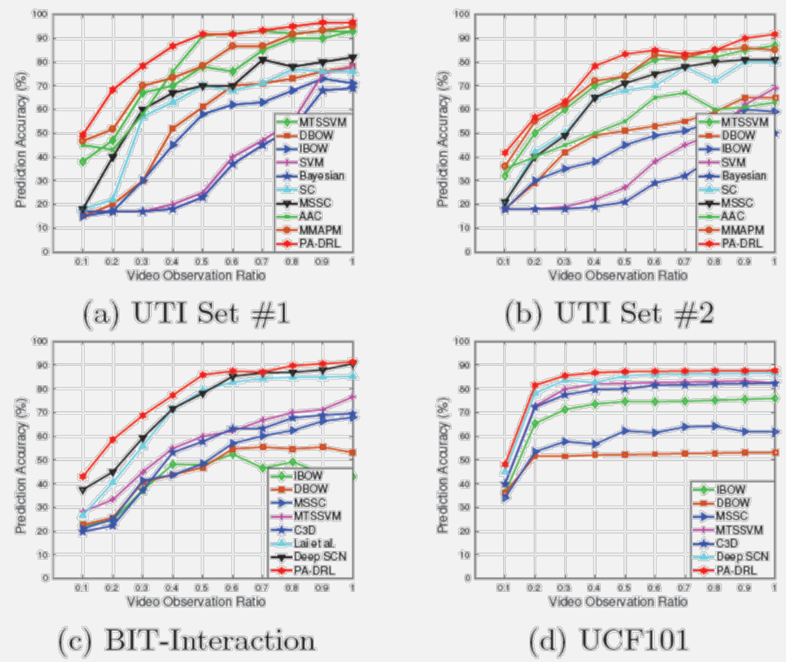
$$R(w) = \frac{1}{T}\sum_{t \in T} r(A_w^t)$$

Part-Activated DRL for Action Prediction

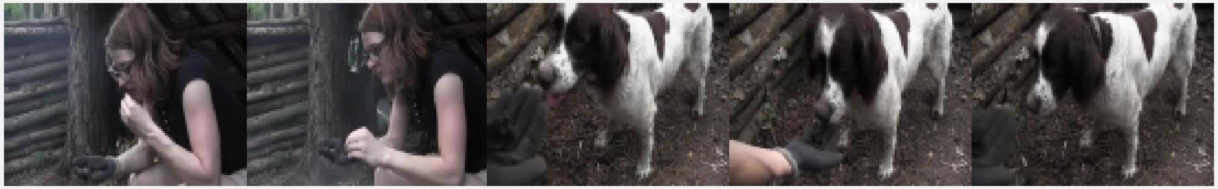


Methods	UTI Set #1		UTI Set #2	
	OR=0.5	OR=1.0	OR=0.5	OR=1.0
SVM [26]	25.3	69.2	27.2	69.2
Bayesian [26]	20.9	78.0	21.8	50.7
IBoW [26]	65.0	81.7	45.7	59.3
DBoW [26]	70.0	85.0	51.2	65.3
SC [28]	70.0	76.7	68.5	80.0
MSSC [28]	70.0	83.3	71.0	81.5
Lan [29]	83.1	88.4	78.3	82.0
MTSSVM [27]	78.3	95.0	74.3	87.3
AAC [17]	88.3	95.0	75.6	63.9
MMAPM [30]	78.3	95.0	75.0	87.3
PA-DRL	**91.7**	**96.7**	**83.3**	**91.7**

Methods	BIT dataset		UCF101	
	OR=0.5	OR=1.0	OR=0.5	OR=1.0
IBoW [26]	49.2	43.0	74.6	76.0
DBoW [26]	46.9	53.1	53.2	53.2
MSSC [28]	48.4	68.0	62.6	61.9
MTSSVM [27]	60.0	76.6	82.3	82.5
Lai *et. al* [5]	79.4	85.3	-	-
Deep SCN [4]	78.1	90.6	85.5	86.7
C3D [46]	57.8	69.6	80.0	82.4
PA-DRL	**85.9**	**91.4**	**87.3**	**87.7**



Video Summary and Caption



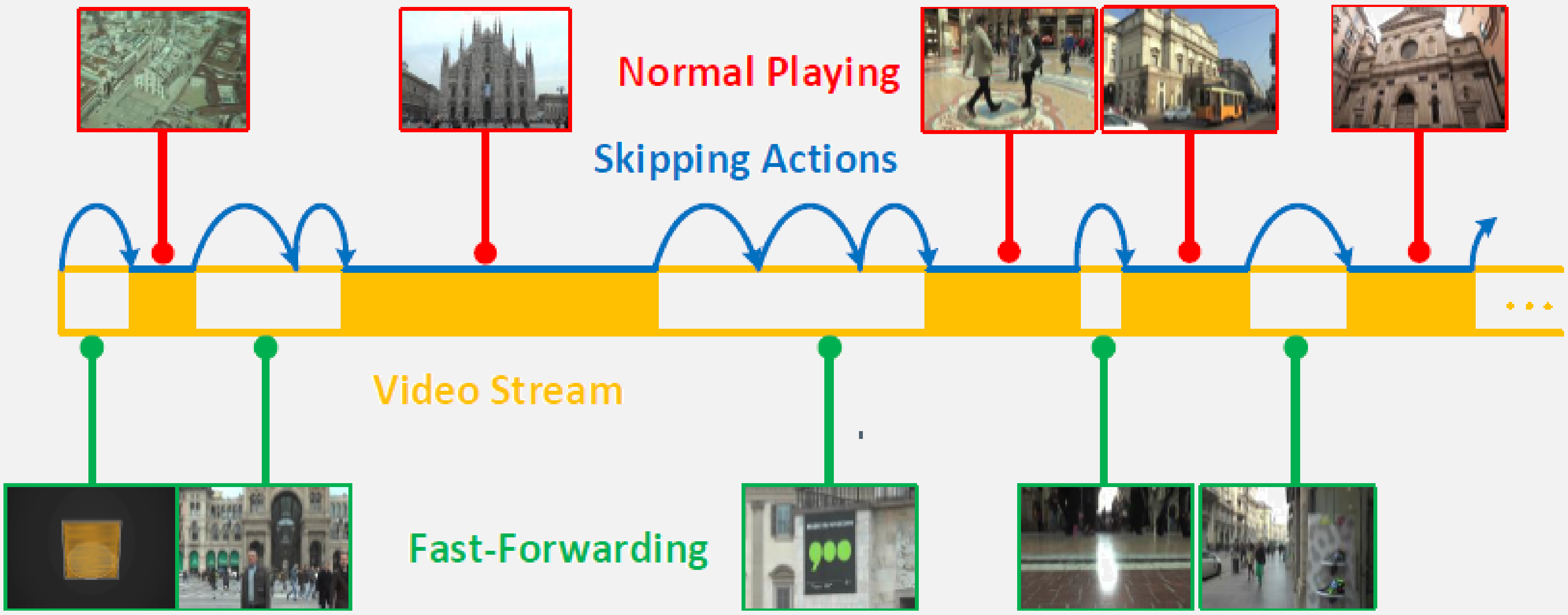
Caption #1: A woman offers her dog some food.

Caption #2: A woman is eating and sharing food with her dog.

Caption #3: A woman is sharing a snack with a dog.

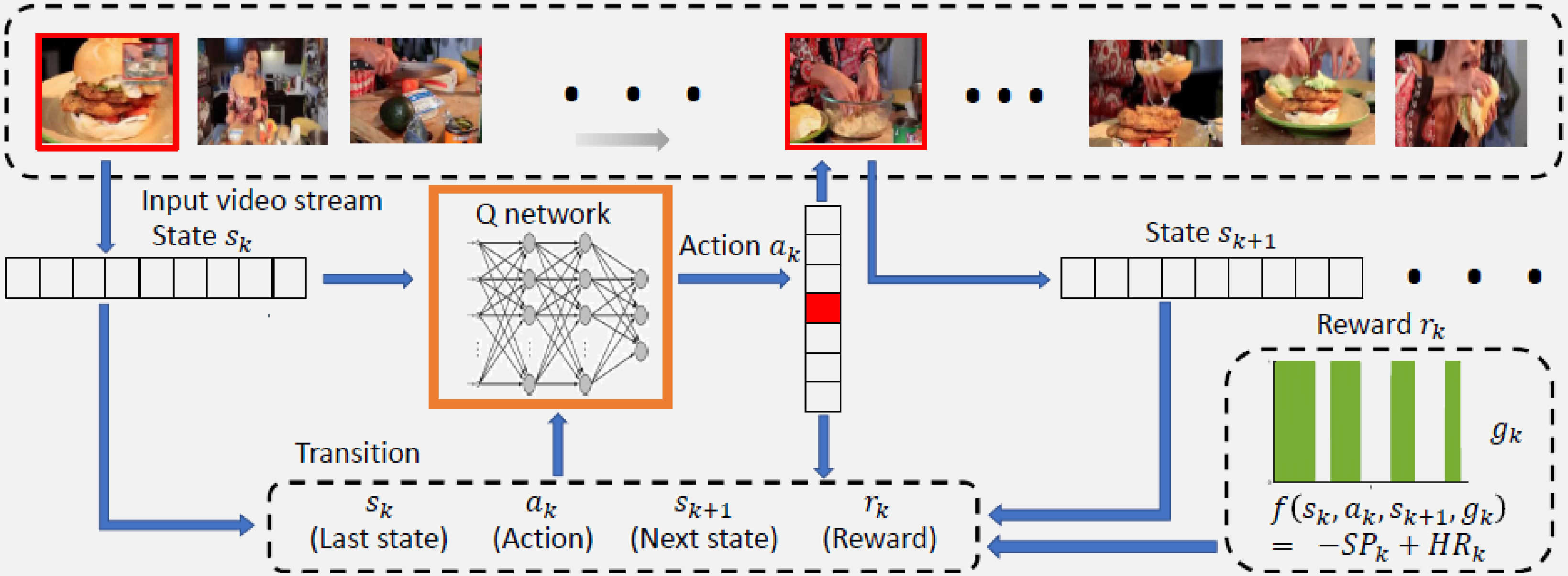
Caption: A person sits on a bed and puts a laptop into a bag. The person stands up, puts the bag on one shoulder, and walks out of the room.

FFNet: Video Fast-Forwarding via Reinforcement Learning



Lan, Shuyue, et al. "FFNet: Video fast-forwarding via reinforcement learning." *CVPR*2018.

FFNet: Video Fast-Forwarding via Reinforcement Learning



$$r_k = -SP_k + HR_k \qquad R = \sum_k \gamma^{k-1} r_k = \sum_k \gamma^{k-1} r(s_k, a_k, s_{k+1})$$

$$SP_k = \frac{\sum_{i \in t_k} \mathbf{1}(l(i)=1)}{T} - \beta \frac{\sum_{i \in t_k} \mathbf{1}(l(i)=0)}{T}$$

$$HR_k = \sum_{i=z-w}^{z+w} \mathbf{1}(l(i)=1) \cdot f_i(z) \qquad f_i(t) = \frac{1}{\sqrt{2\pi\sigma^2}} exp(-\frac{(t-i)^2}{2\sigma^2}), t \in [i-w, i+w]$$

FFNet: Video Fast-Forwarding via Reinforcement Learning

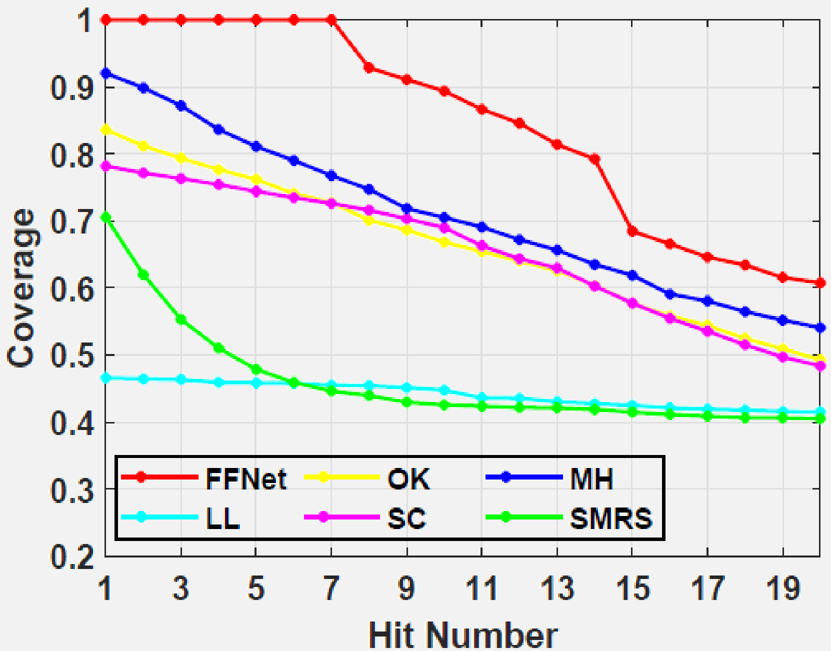


Figure 3. **Segment-level coverage on Tour20 dataset with different hit number thresholds.** Our FFNet (red line on top) outperforms all other methods by a significant margin.

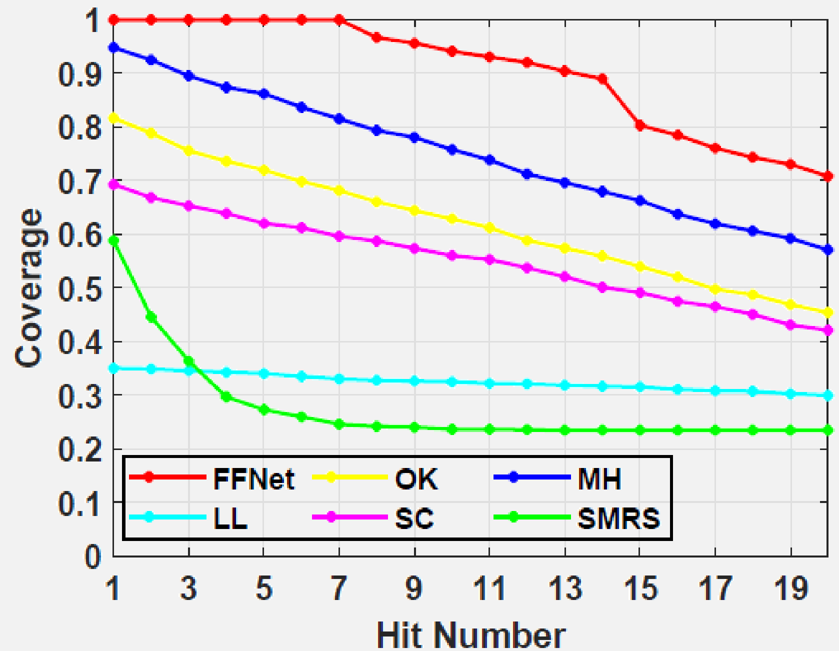


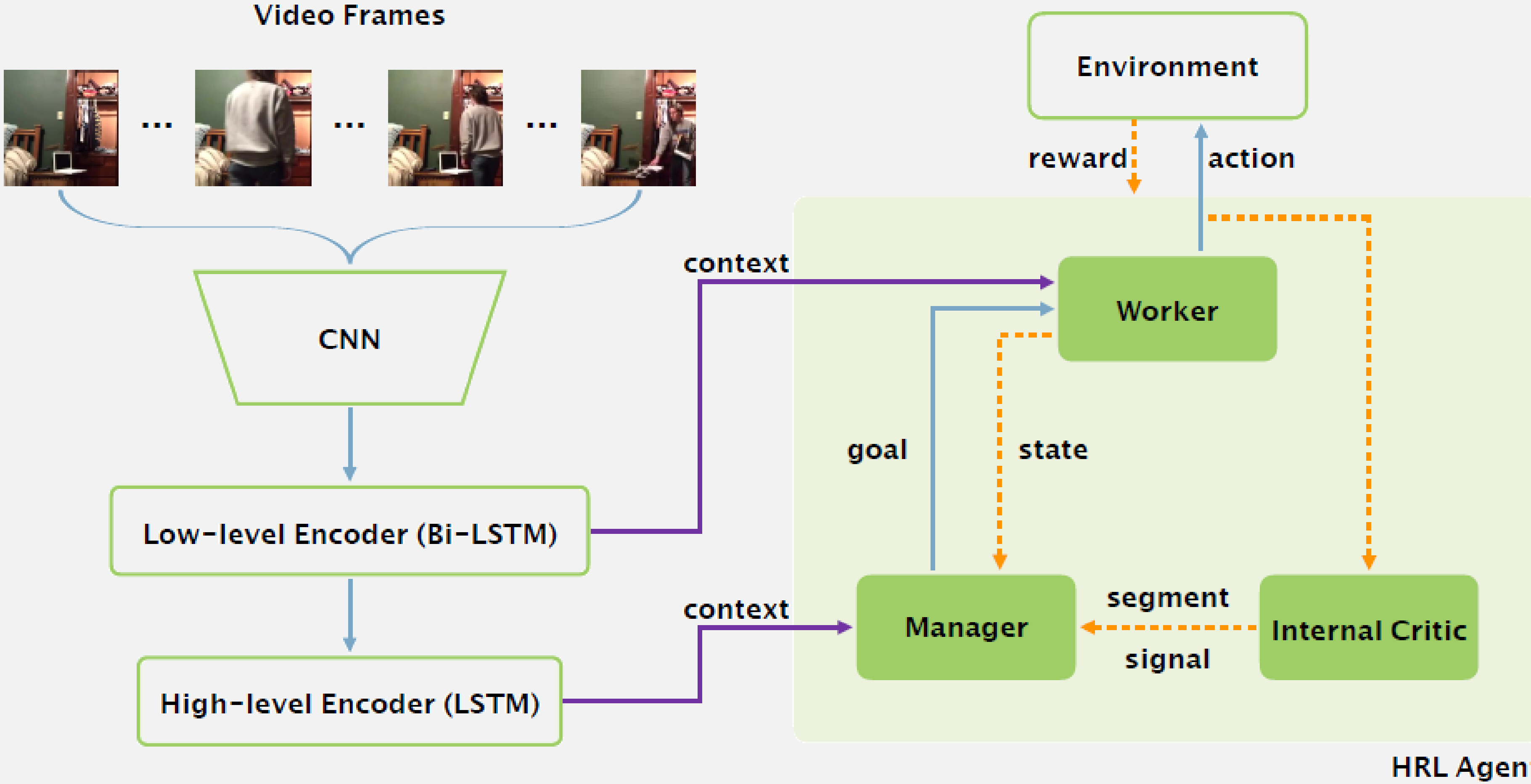
Figure 4. **Segment-level coverage on TVSum dataset under different hit number thresholds.** Our FFNet (red line on top) outperforms all other methods by a significant margin.

Video Captioning via Hierarchical Reinforcement Learning

- High-level： manager
  - operates at a lower temporal resolution and emits a goal for the worker to accomplish
- Low-level： worker
  - generates a word for each time step by following the goal
    proposed by the manager
- Internal Critic
  - to determine whether the worker has accomplished a goal

Wang, Xin, et al. "Video captioning via hierarchical reinforcement learning." *CVPR*2018.

Video Captioning via Hierarchical Reinforcement Learning



Video Captioning via Hierarchical Reinforcement Learning

- Worker:

$$f(x) = \mathrm{CIDEr}(sent + x) - \mathrm{CIDEr}(sent)$$

$$a_t \ (a_t \in V)$$

$$R(a_t) = \sum_{k=0}^{\infty} \gamma^k f(a_{t+k})$$

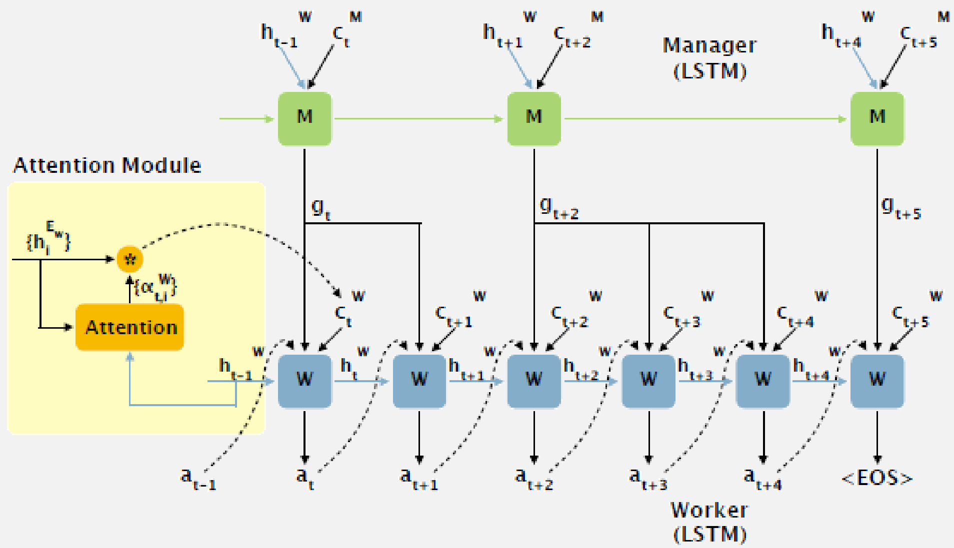
$$L(\theta_w) = -\mathbb{E}_{a_t \sim \pi_{\theta_w}}[R(a_t)]$$

- Manager:

$$e_{t,c} = a_t a_{t+1} \ldots a_{t+c-1}$$

$$R(e_t) = \sum_{n=0}^{\infty} \gamma^n f(e_{t+n})$$

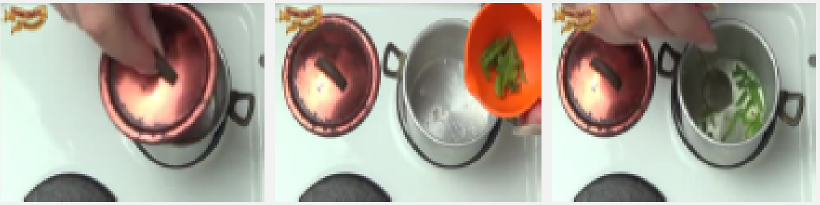
$$L(\theta_m) = -\mathbb{E}_{g_t}[R(e_t)\pi(e_{t,c}; s_t, g_t = \mu_{\theta_m}(s_t))]$$



Video Captioning via Hierarchical Reinforcement Learning

GROUND TRUTH:
people dancing and singing on the beach .
young men and women sing and dance in beach party fashion .
XE-BASELINE:
people are dancing .
RL-BASELINE:
a group of people are dancing .
HRL:
a group of people | are dancing on the beach .

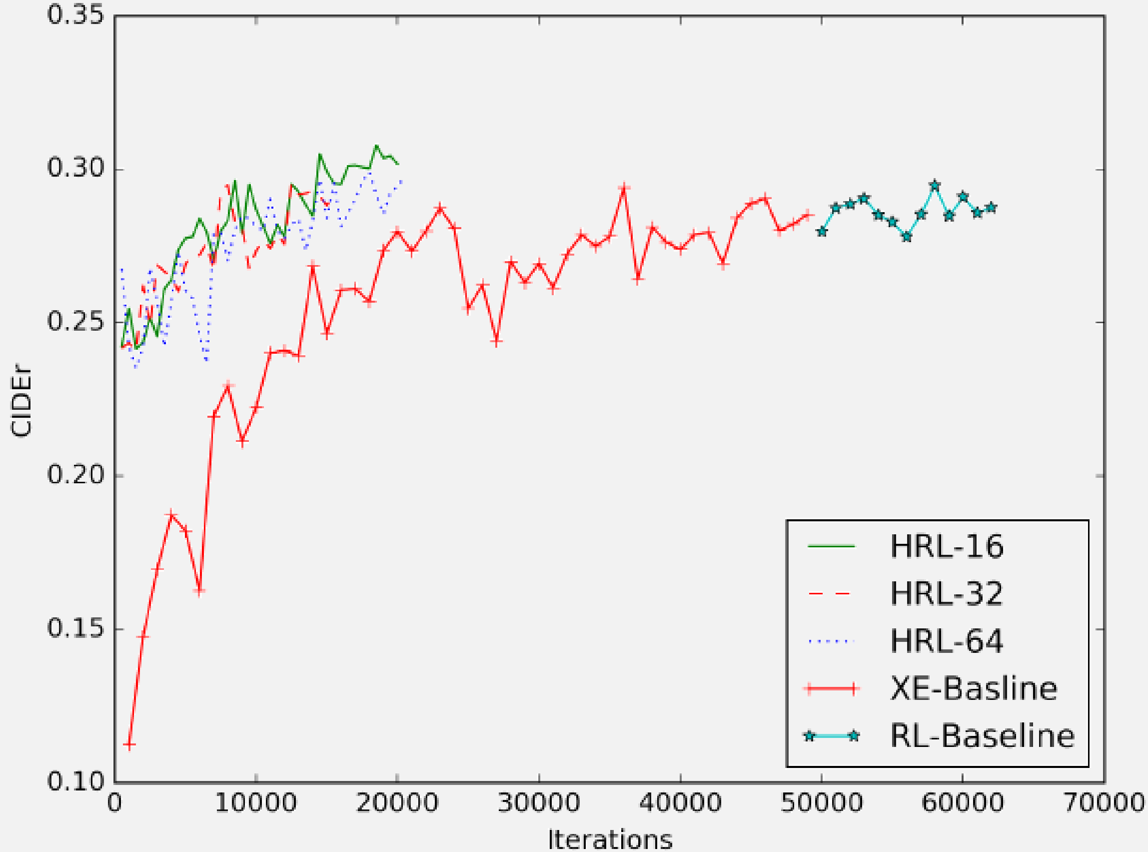
(a)



GROUND TRUTH:
a person is mixing some food .
a woman adds green vegetables to a tiny pot of boiling water .
XE-BASELINE:
there is a woman is making a dish .
RL-BASELINE:
a woman is cooking in a pot in the kitchen .
HRL:
a woman | is cooking in a bowl | and mixing the water .

(b)

Goal Dimension



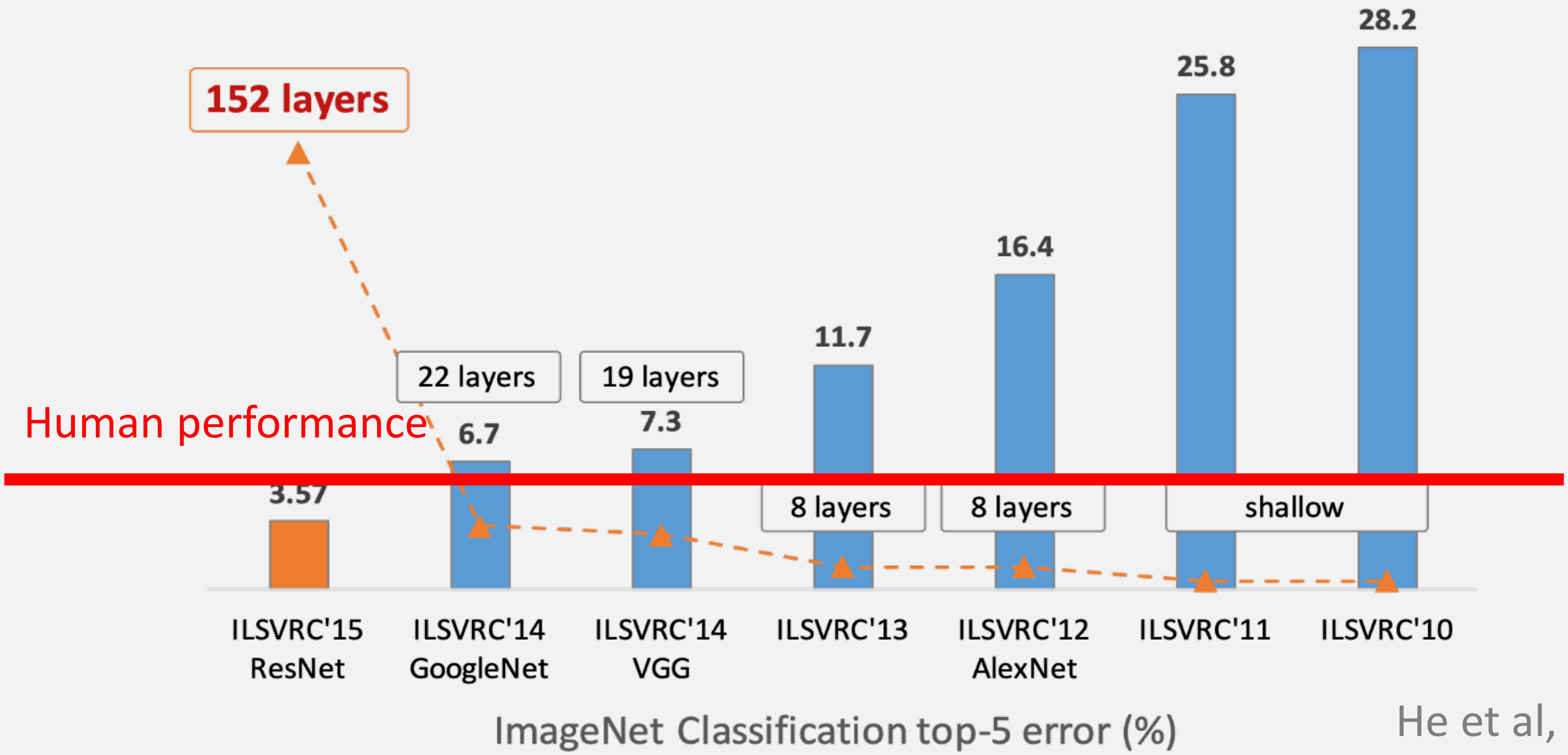
THANK YOU
Q&A

--------------------Short Break: 30 minutes (15:00-15:30)---------------------

Part 3:
DRL for Network Structure Learning

DRL for Network Structure Learning

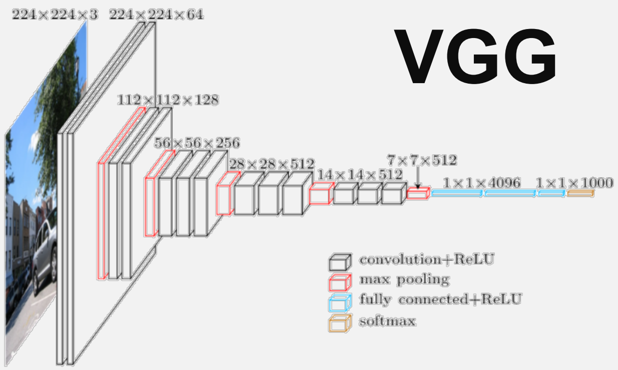
Recent success in visual recognition is mainly driven by the advances in deep network architecture design.



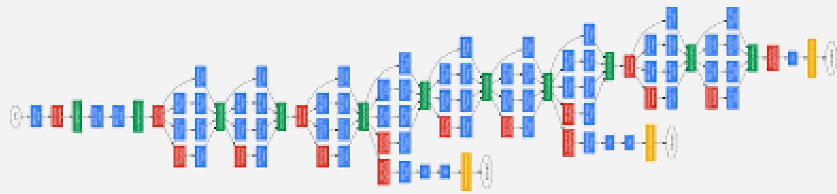
He et al, 2015

DRL for Network Structure Learning

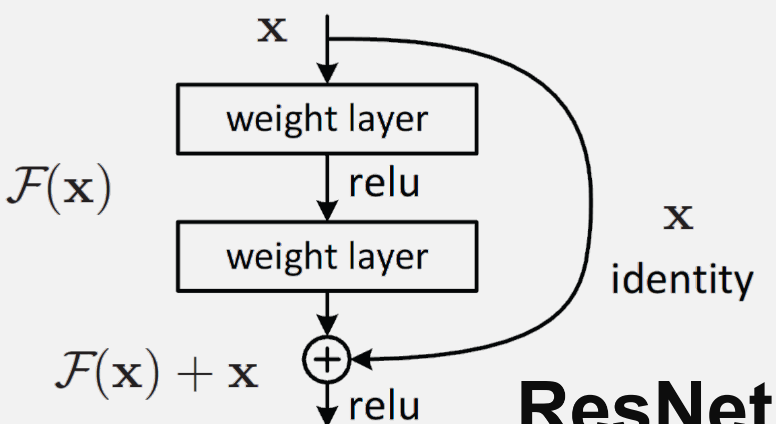
High-performance CNNs for visual recognition



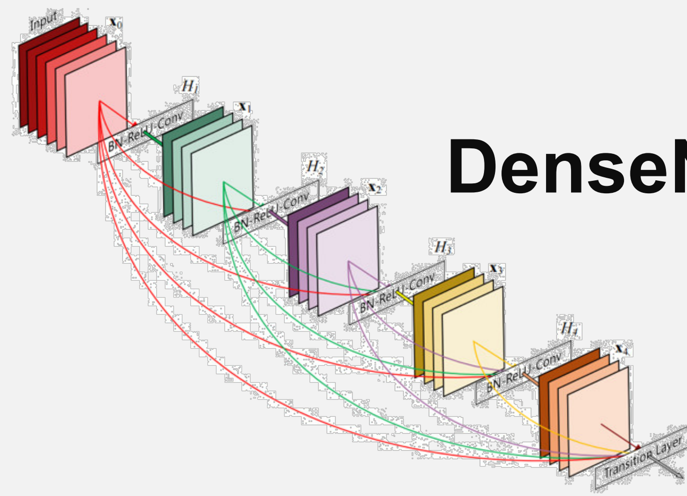
VGG



GoogLeNet



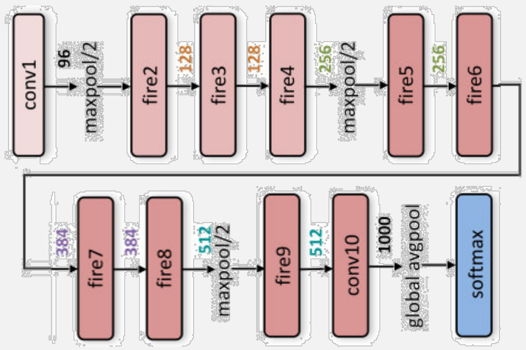
ResNet



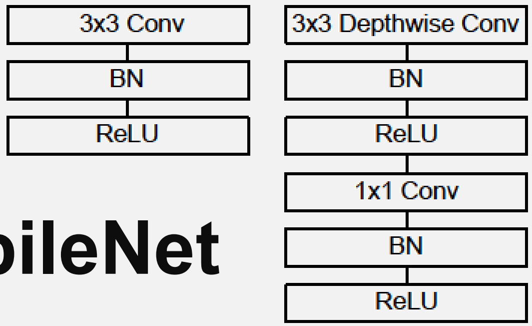
DenseNet

DRL for Network Structure Learning

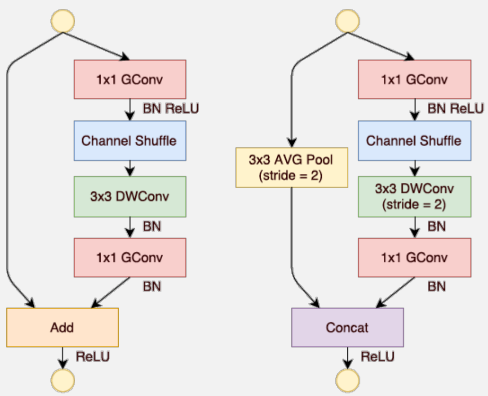
Efficient CNNs for Mobile Vision Applications



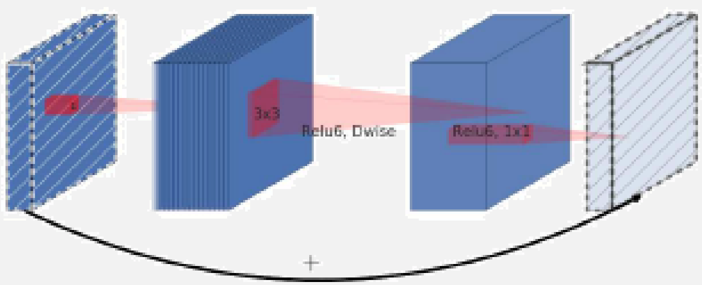
SqueezeNet



MobileNet



ShuffleNet

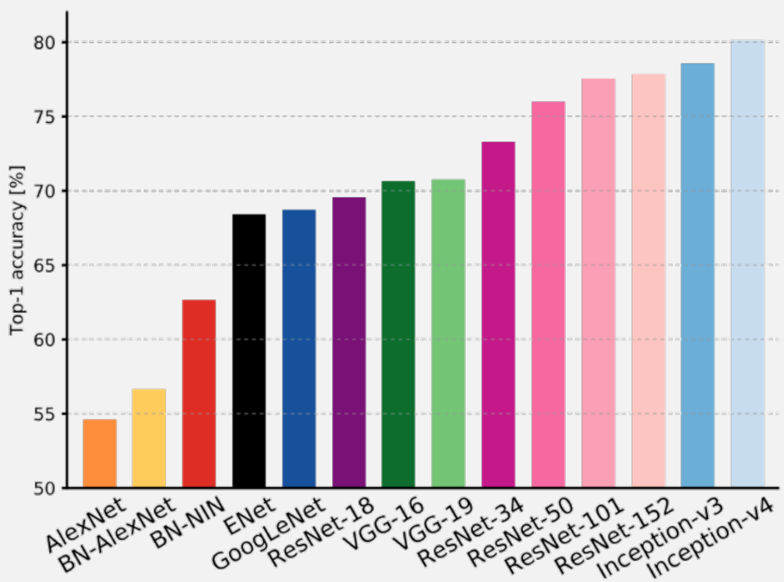


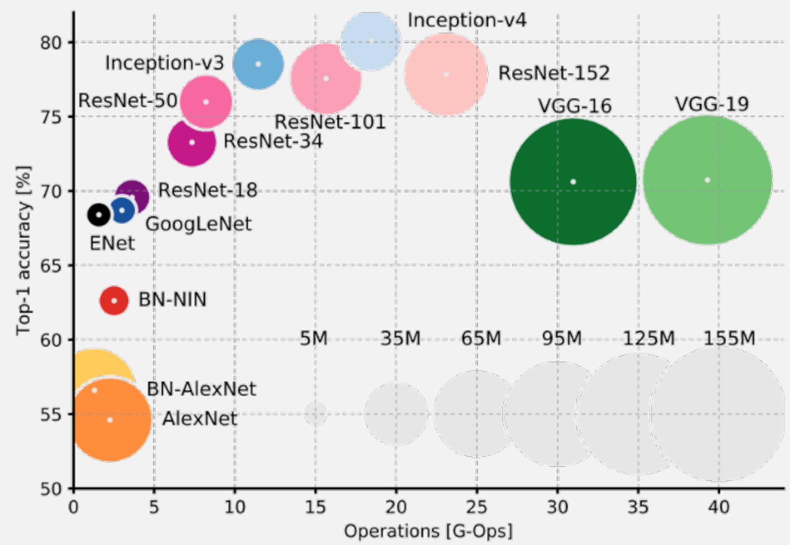
MobileNetV2

DRL for Architecture Search

Automated architecture design through DRL:

- designing neural network architectures is hard
- there is not a lot of intuition into how to design them well





Canziani et al, 2017

DRL for Architecture Search

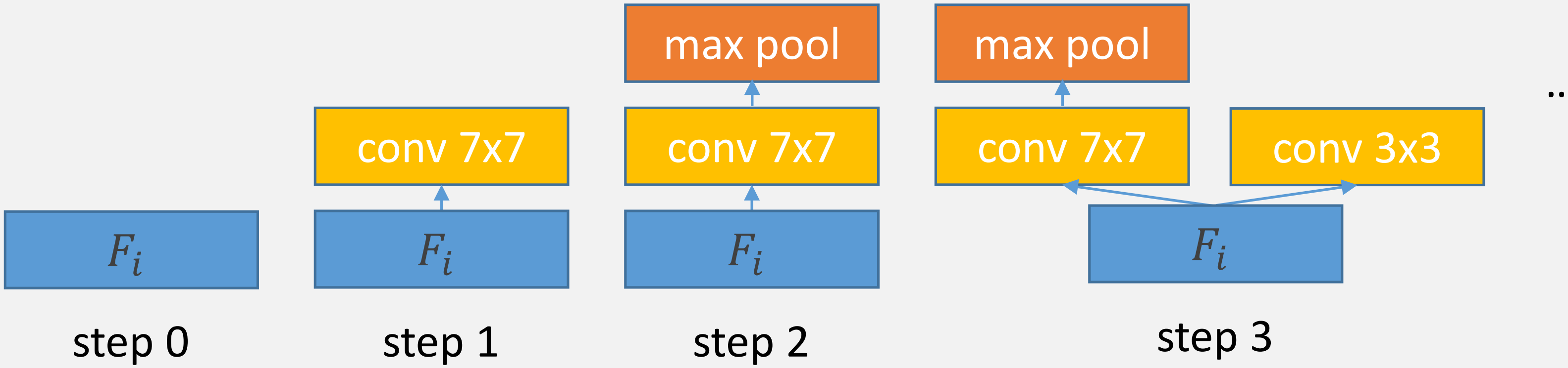
Automated architecture design through DRL:

- Empty architecture $A_0 = \emptyset$

  State $x_t = (A_t, h_t) \in \chi$ current network architecture

  Action $a_t: x_t \rightarrow x_{t+1}, \mathrm{a} \in A$ the type of the current layer

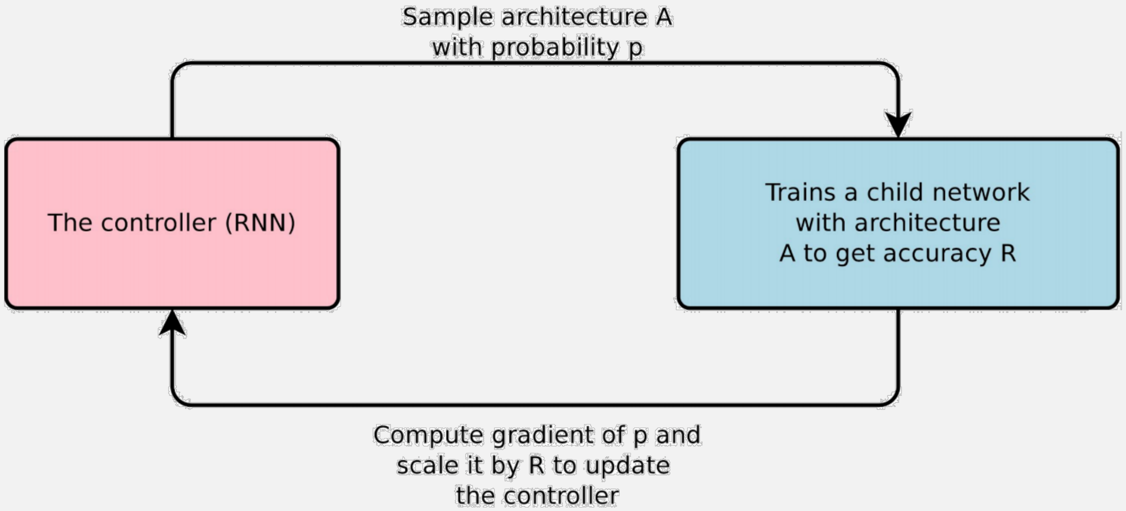
  Reward: the performance of the sampled network architecture



DRL for Architecture Search

Neural Architecture Search with Reinforcement Learning

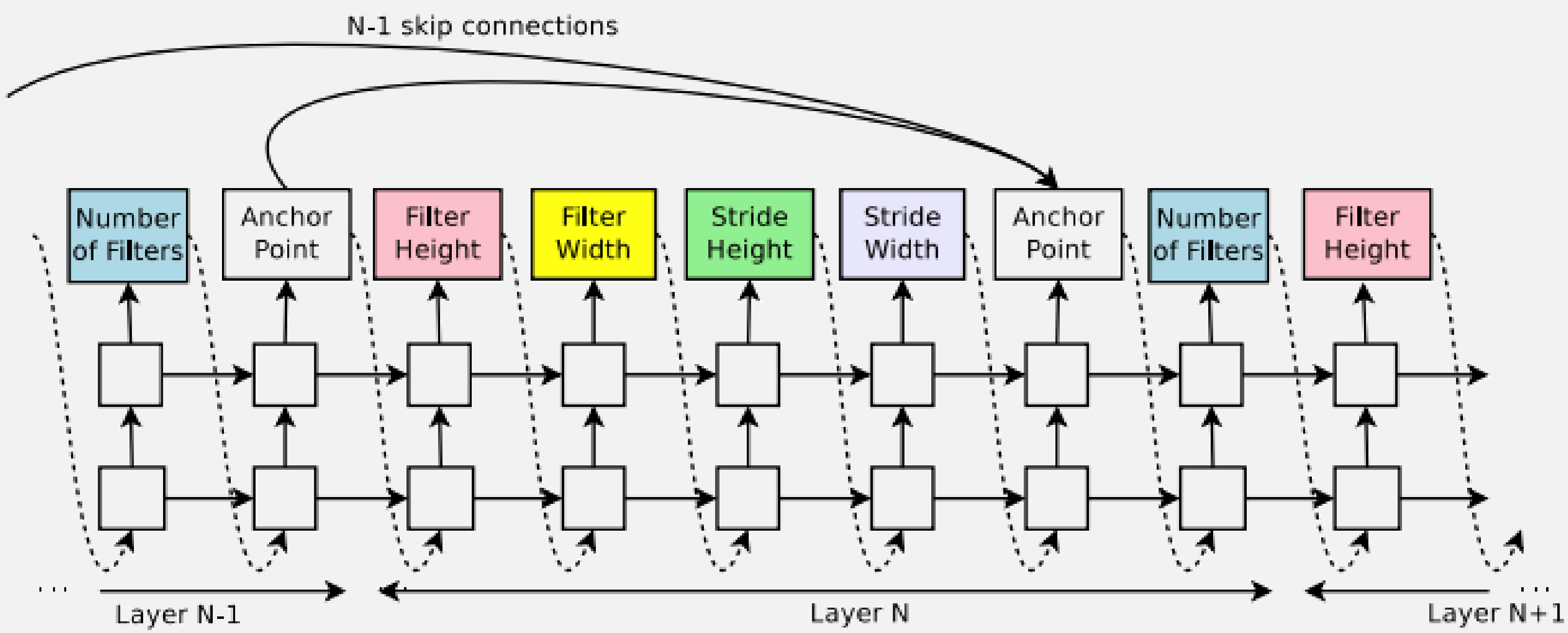
- ❖ use a RNN (Controller) to generate the structures and connectivity that specifies a neural network architecture
- ❖ use the validation accuracy as reward to update the Controller



Zoph B, Le Q V. Neural architecture search with reinforcement learning. arXiv preprint arXiv:1611.01578, 2016.

DRL for Architecture Search

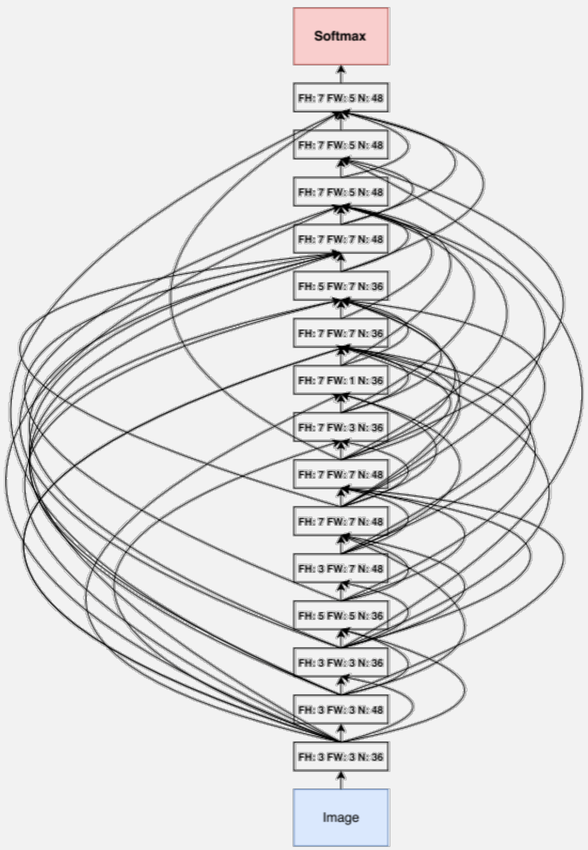
Neural Architecture Search with Reinforcement Learning



Zoph B, Le Q V. Neural architecture search with reinforcement learning. arXiv preprint arXiv:1611.01578, 2016.

DRL for Architecture Search

Neural Architecture Search with Reinforcement Learning

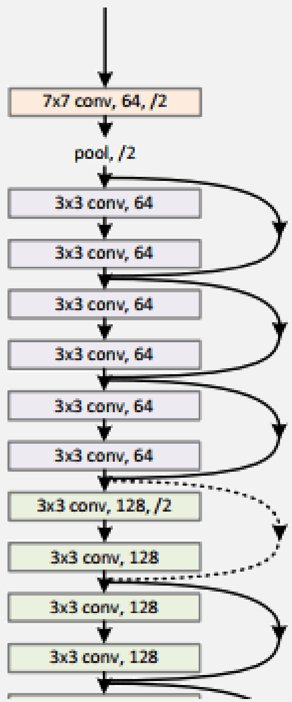


Model	Depth	Parameters	Error rate (%)
Network in Network (Lin et al., 2013)	-	-	8.81
All-CNN (Springenberg et al., 2014)	-	-	7.25
Deeply Supervised Net (Lee et al., 2015)	-	-	7.97
Highway Network (Srivastava et al., 2015)	-	-	7.72
Scalable Bayesian Optimization (Snoek et al., 2015)	-	-	6.37
FractalNet (Larsson et al., 2016)	21	38.6M	5.22
with Dropout/Drop-path	21	38.6M	4.60
ResNet (He et al., 2016a)	110	1.7M	6.61
ResNet (reported by Huang et al. (2016c))	110	1.7M	6.41
ResNet with Stochastic Depth (Huang et al., 2016c)	110	1.7M	5.23
	1202	10.2M	4.91
Wide ResNet (Zagoruyko & Komodakis, 2016)	16	11.0M	4.81
	28	36.5M	4.17
ResNet (pre-activation) (He et al., 2016b)	164	1.7M	5.46
	1001	10.2M	4.62
DenseNet ($L = 40, k = 12$) Huang et al. (2016a)	40	1.0M	5.24
DenseNet($L = 100, k = 12$) Huang et al. (2016a)	100	7.0M	4.10
DenseNet ($L = 100, k = 24$) Huang et al. (2016a)	100	27.2M	3.74
DenseNet-BC ($L = 100, k = 40$) Huang et al. (2016b)	190	25.6M	3.46
Neural Architecture Search v1 no stride or pooling	15	4.2M	5.50
Neural Architecture Search v2 predicting strides	20	2.5M	6.01
Neural Architecture Search v3 max pooling	39	7.1M	4.47
Neural Architecture Search v3 max pooling + more filters	39	37.4M	3.65

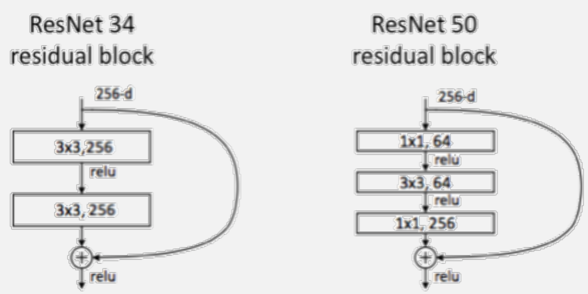
Performance on CIFAR-10

DRL for Architecture Search

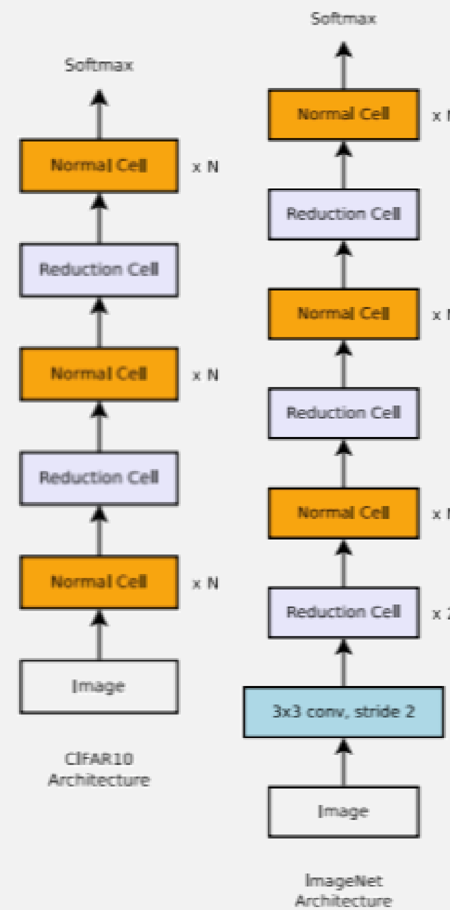
Learning Transferable Architectures for Scalable Image Recognition



ResNet



repeated block (cell) in ResNet



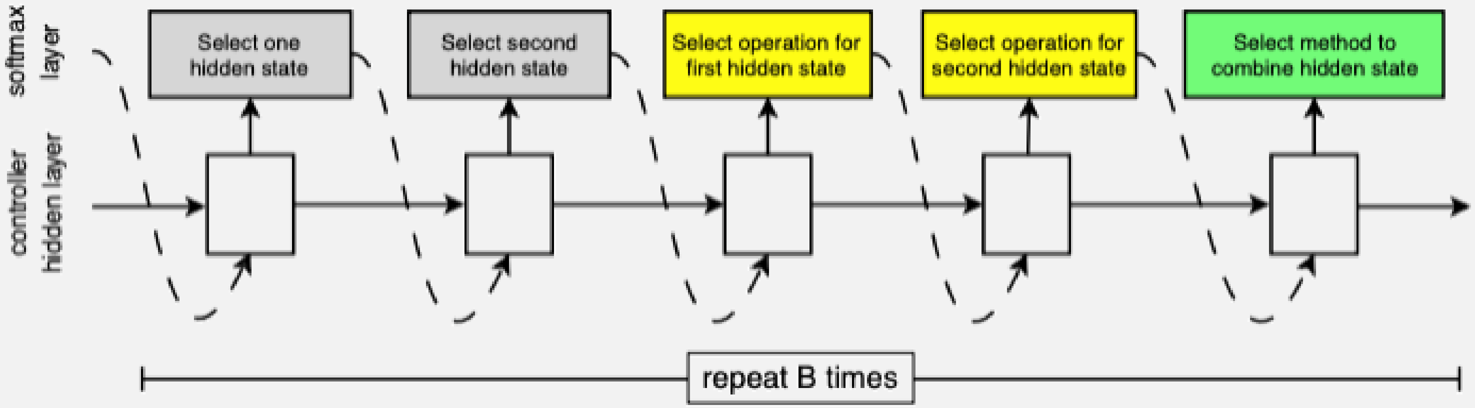
Zoph B, Vasudevan V, Shlens J, et al. Learning transferable architectures for scalable image recognition. CVPR. 2018.

DRL for Architecture Search

Learning Transferable Architectures for Scalable Image Recognition

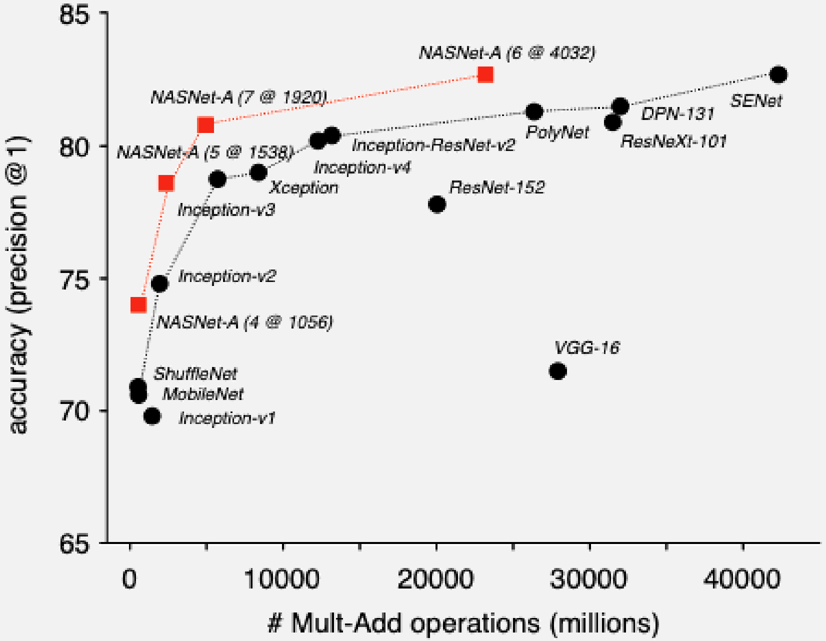
Operation set:

- identity
- 1x7 then 7x1 convolution
- 3x3 average pooling
- 5x5 max pooling
- 1x1 convolution
- 3x3 depthwise-separable conv
- 7x7 depthwise-separable conv
- 1x3 then 3x1 convolution
- 3x3 dilated convolution
- 3x3 max pooling
- 7x7 max pooling
- 3x3 convolution
- 5x5 depthwise-seperable conv

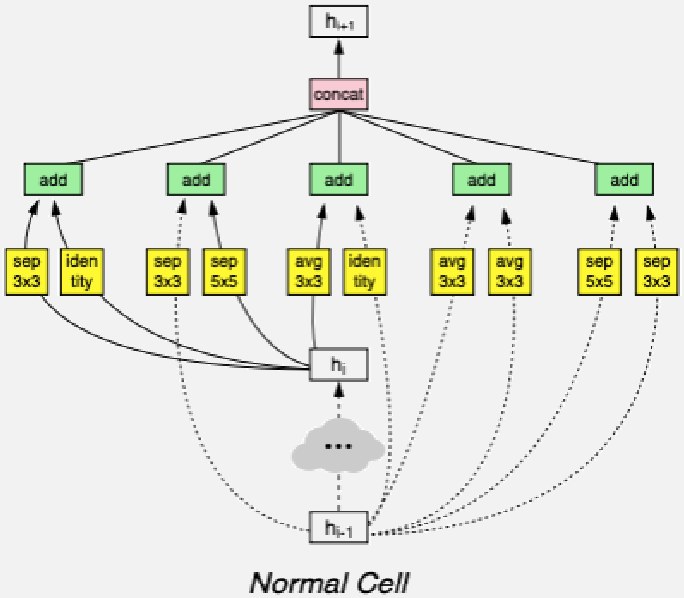


DRL for Architecture Search

Learning Transferable Architectures for Scalable Image Recognition



Performance on ImageNet



DRL for Efficient Network Design

DRL for efficient network design:

❖ Light-weight network architecture search

Model	# parameters	Mult-Adds	Top 1 Acc. (%)	Top 5 Acc. (%)
Inception V1 [58]	6.6M	1,448 M	69.8	89.9
MobileNet-224 [24]	4.2 M	569 M	70.6	89.5
ShuffleNet (2x) [69]	$\sim$ 5M	524 M	70.9	89.8
NASNet-A (4 @ 1056)	**5.3 M**	**564 M**	**74.0**	**91.6**
NASNet-B (4 @ 1536)	5.3M	488 M	72.8	91.3
NASNet-C (3 @ 960)	4.9M	558 M	72.5	91.0

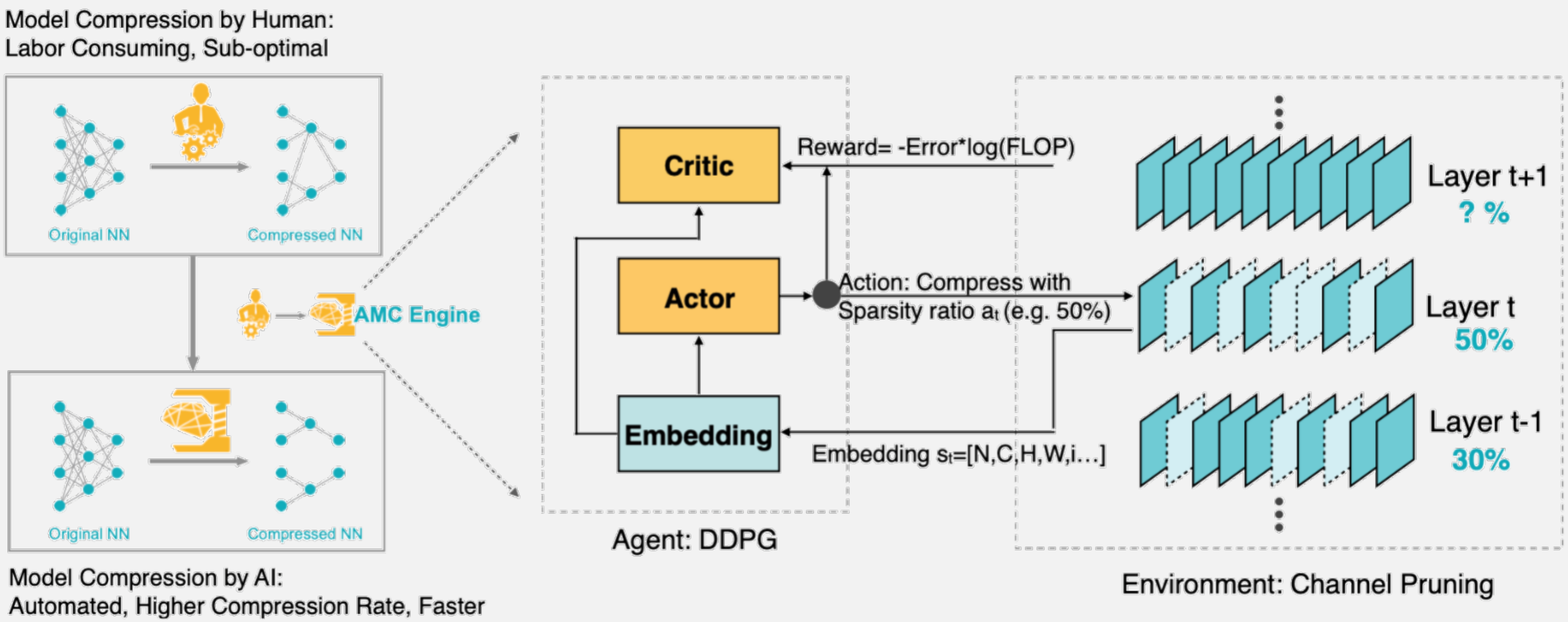
DRL-based architecture search with complexity constraints

Zoph B, Vasudevan V, Shlens J, et al. Learning transferable architectures for scalable image recognition. CVPR. 2018.

DRL for Efficient Network Design

DRL for efficient network design:

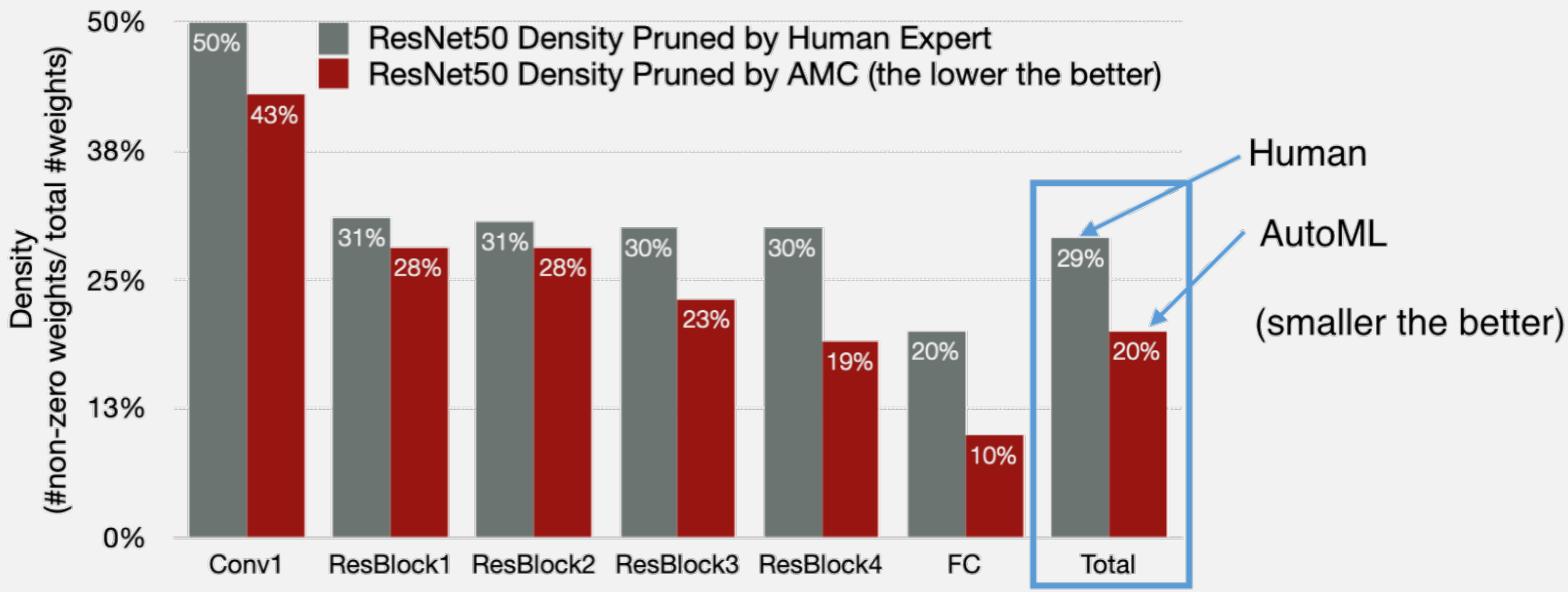
❖ Automated model compression/pruning



He Y, Lin J, Liu Z, et al. AMC: AutoML for model compression and acceleration on mobile devices. ECCV, 2018.

DRL for Efficient Network Design

❖ Automated model compression



DRL for Efficient Network Design

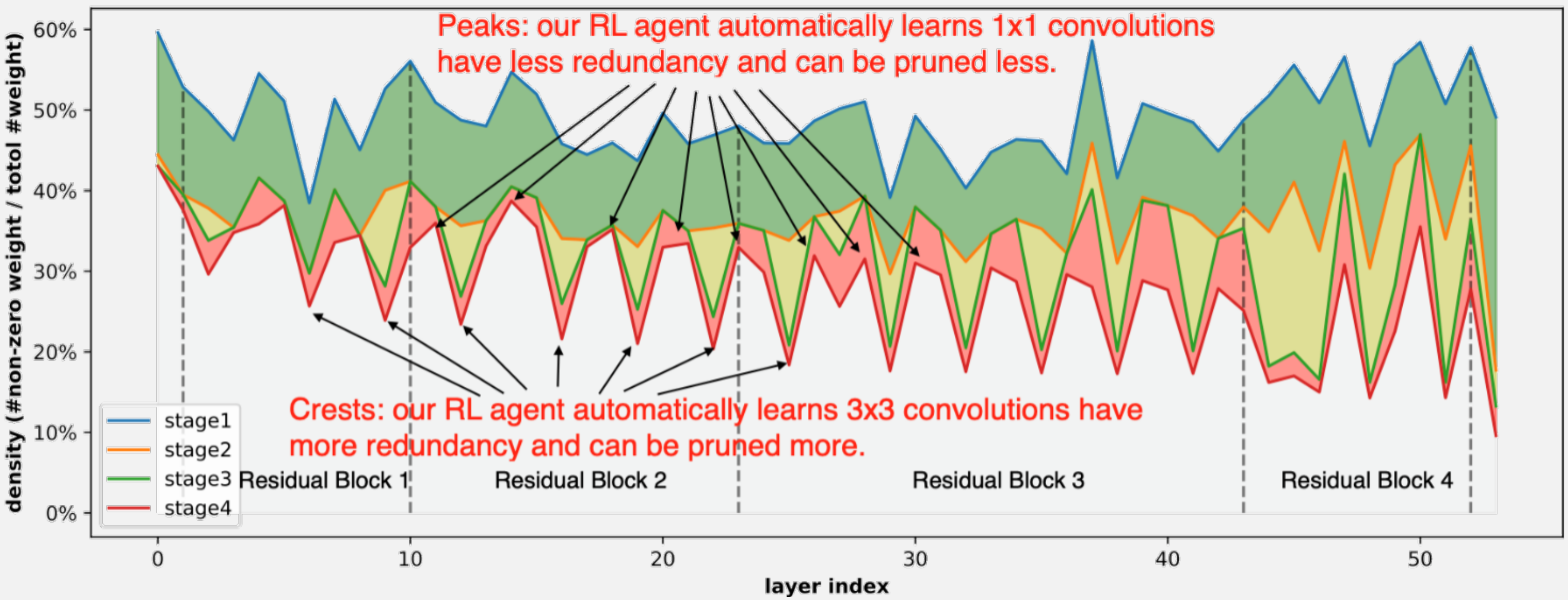
❖ Automated model compression

Model	MAC	Top-1	Top-5	Latency	Speed	Memory
1.0 MobileNet	569M	70.6%	89.5%	**119.0ms**	8.4 fps	20.1MB
AMC (50% MAC)	285M	70.5%	89.3%	**64.4ms**	15.5 fps (1.8x)	14.3MB
AMC (50% Time)	272M	70.2%	89.2%	**59.7ms**	16.8 fps (2.0x)	13.2MB
0.75 MobileNet	325M	68.4%	88.2%	**69.5ms**	14.4 fps (1.7x)	14.8MB

Performance on Mobile Devices

DRL for Efficient Network Design

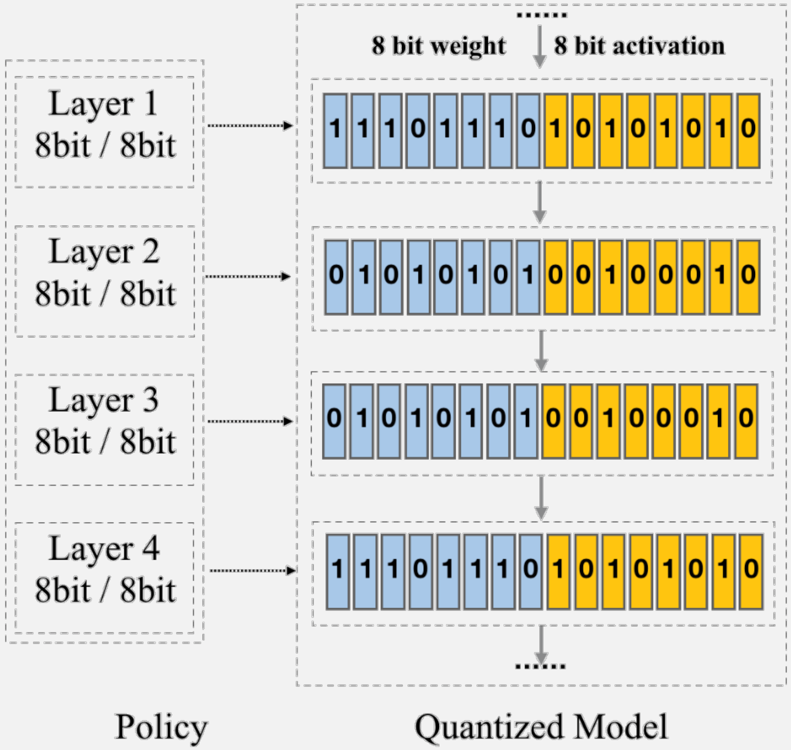
❖ Automated model compression



RL agents are also helpful to find the layers which are more critical

DRL for Efficient Network Design

❖ Automated model quantization with mixed-precision



Conventional quantization method quantize all layers with the same precision

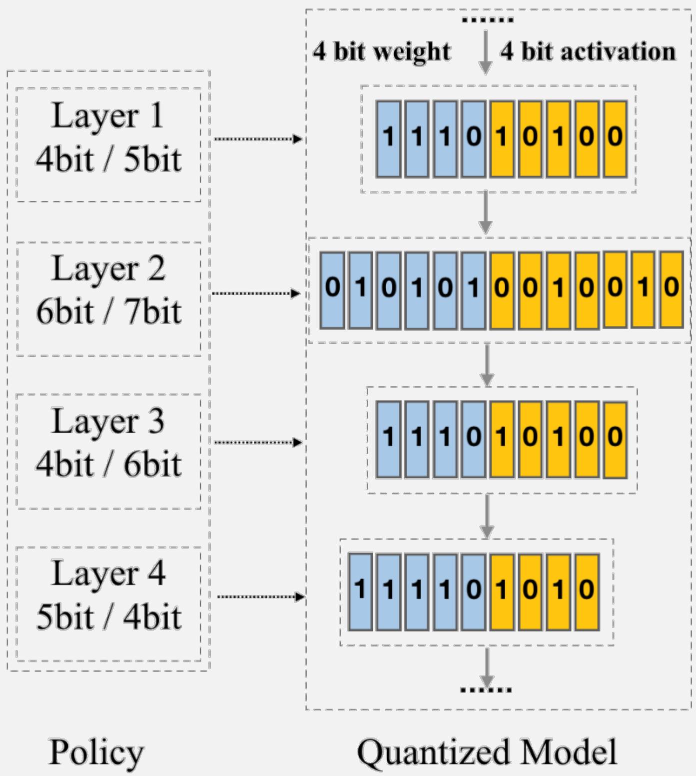
Apple's new A12 chip supports flexible bits for neural network inference

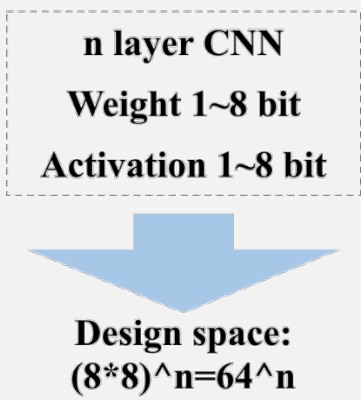
Kuan Wang, Zhijian Liu, Yujun Lin, Ji Lin, Song Han. HAQ: Hardware-aware Automated Quantization with Mixed-precision. CVPR, 2019

DRL for Efficient Network Design

❖ Automated model quantization with mixed-precision

Mixed-precision quantization:

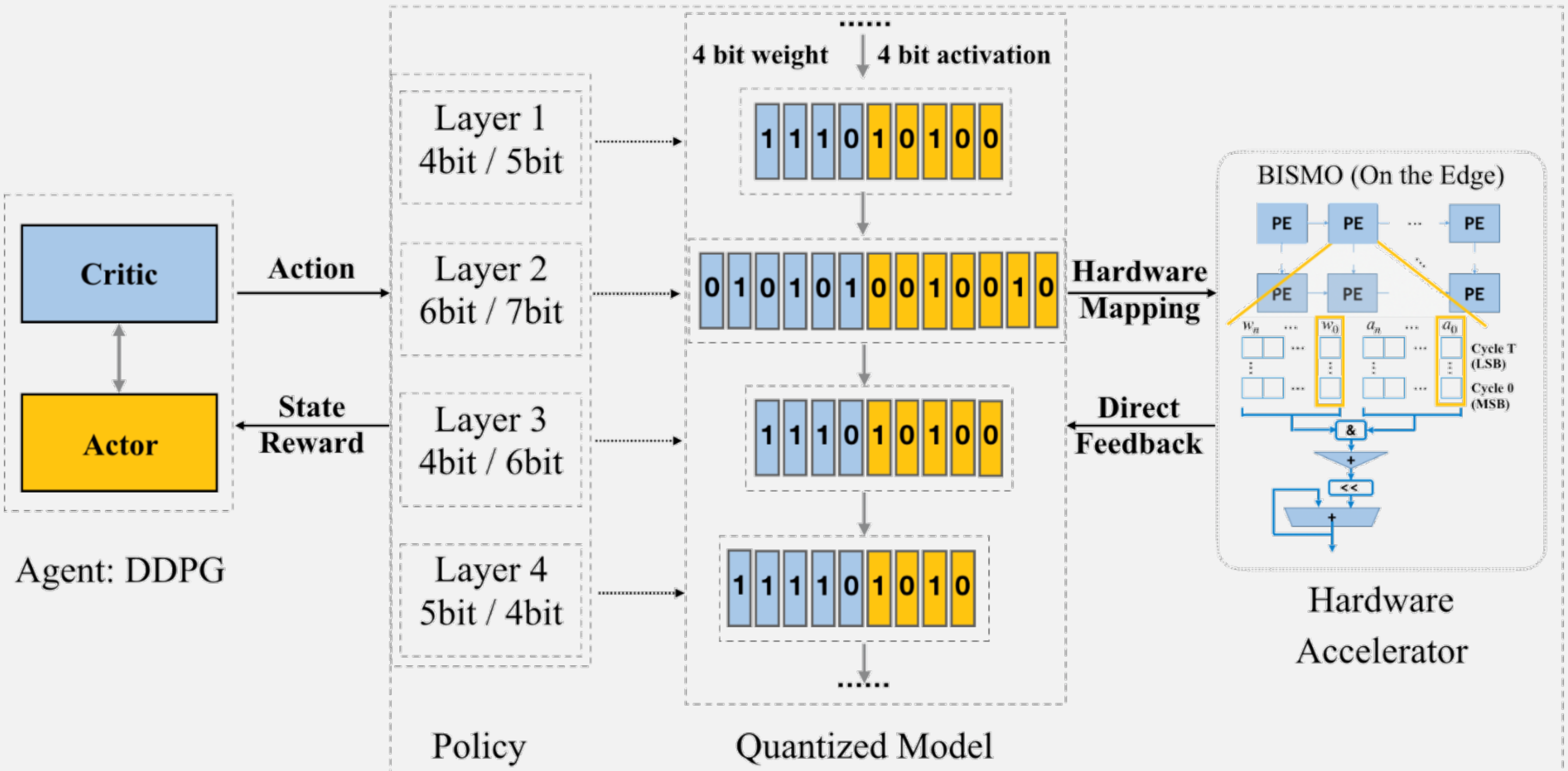




The design space is quiet huge, so learning based method is needed.

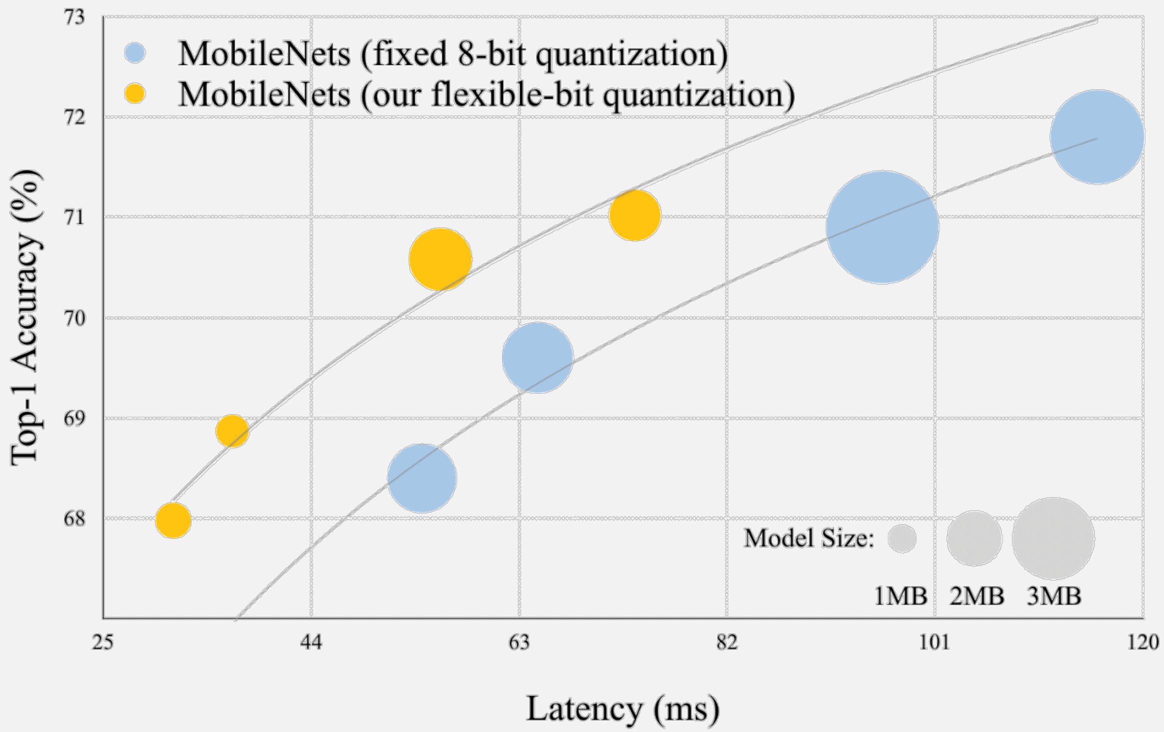
DRL for Efficient Network Design

❖ Automated model quantization with mixed-precision



DRL for Efficient Network Design

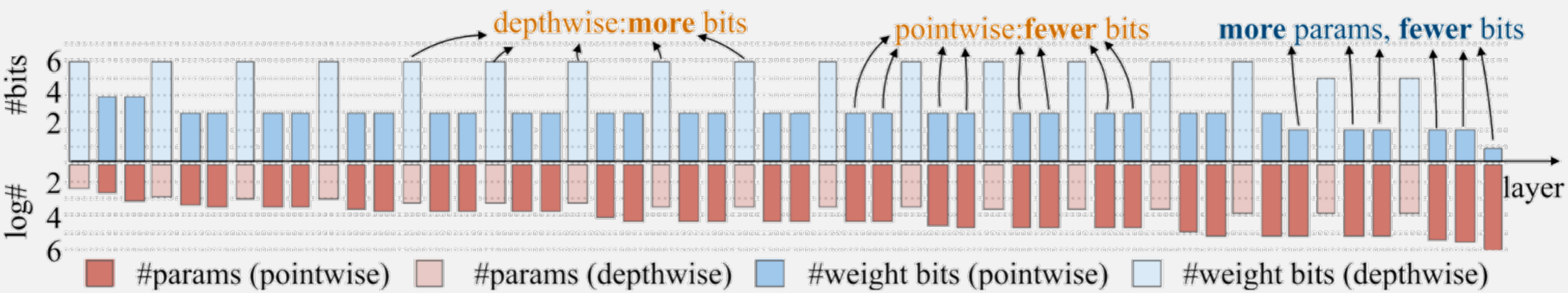
❖ Automated model quantization with mixed-precision



Flexible bit policies for MobileNets are much better than fixed 8bit policy

DRL for Efficient Network Design

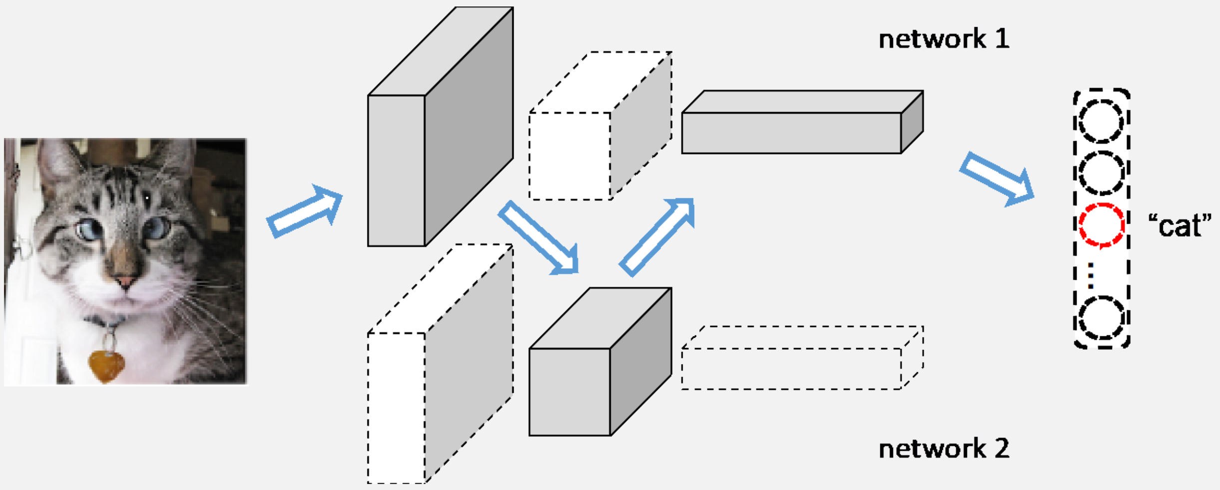
❖ Automated model quantization with mixed-precision

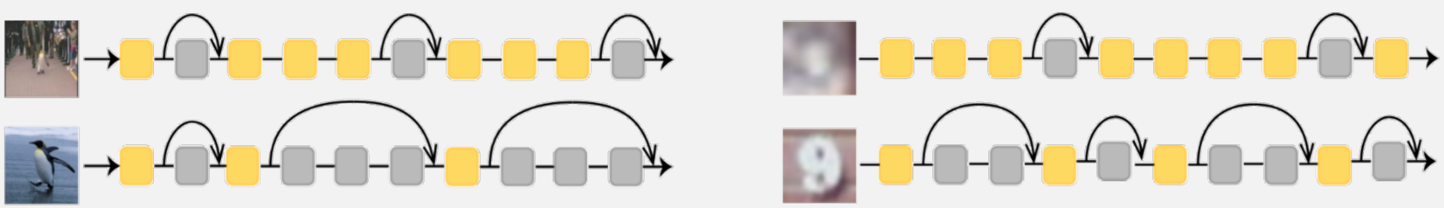


Model size constrained experiments for MobileNet-V2

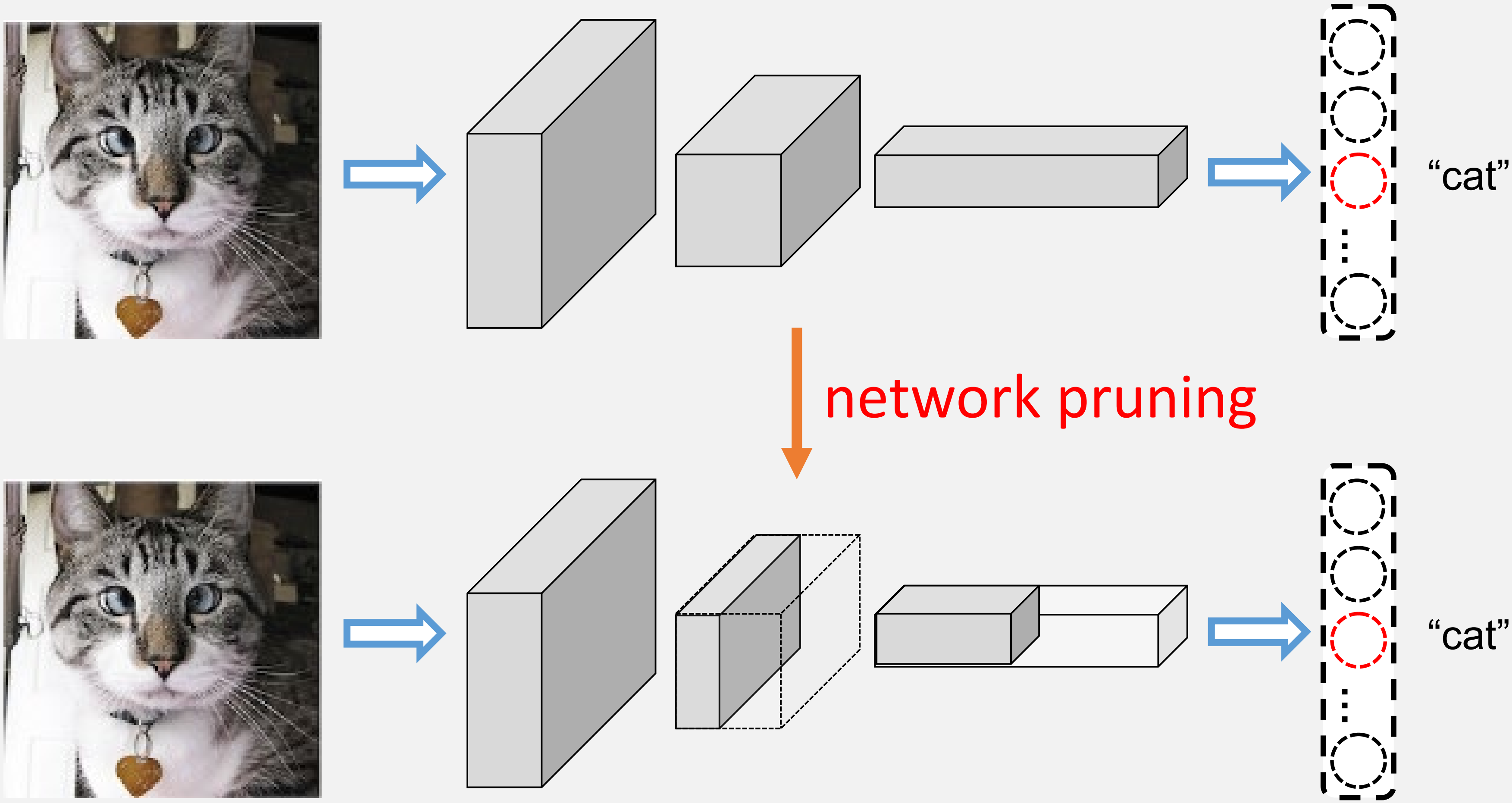
Learning Dynamic Networks with DRL

Dynamic Networks:

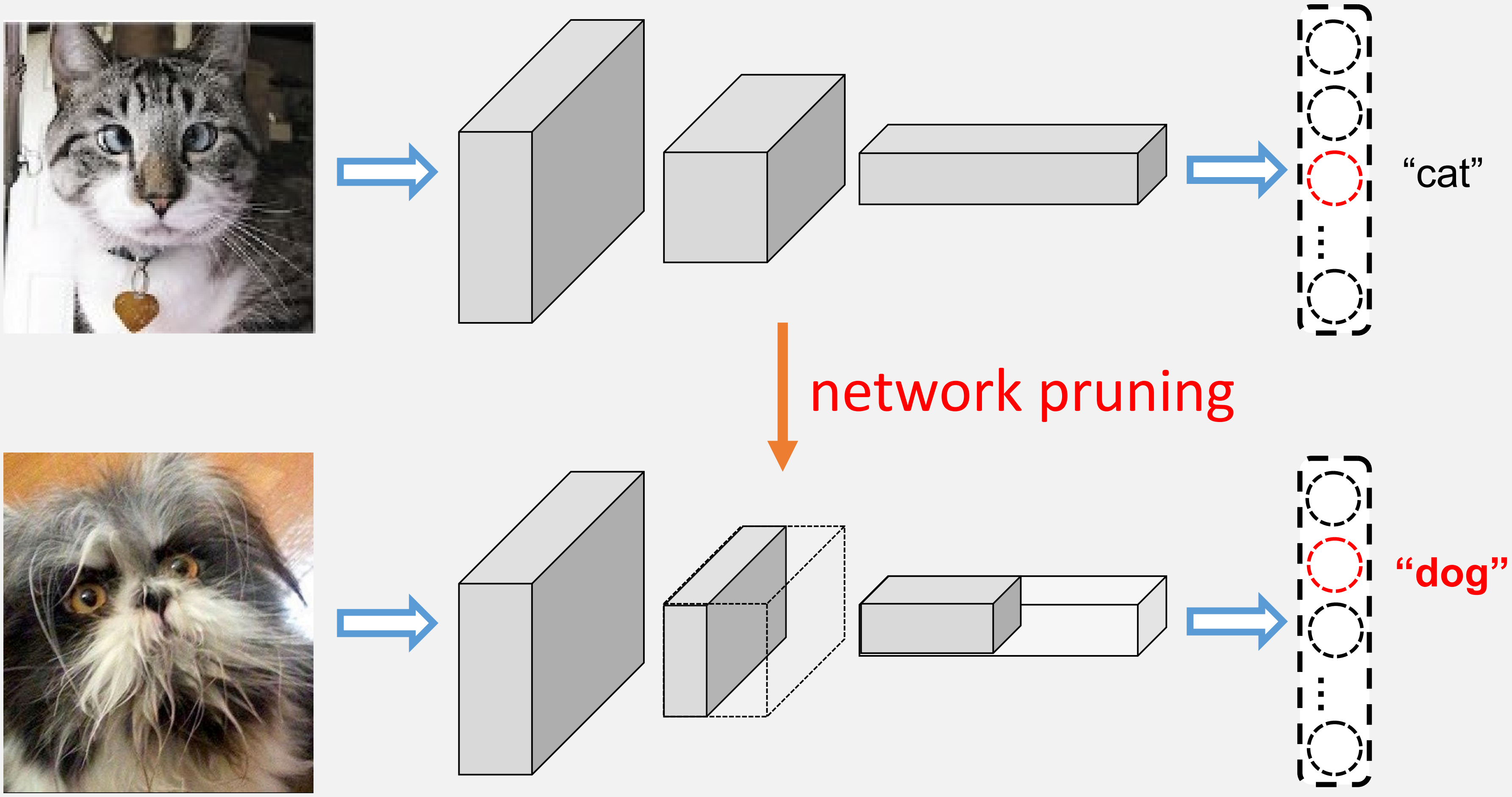




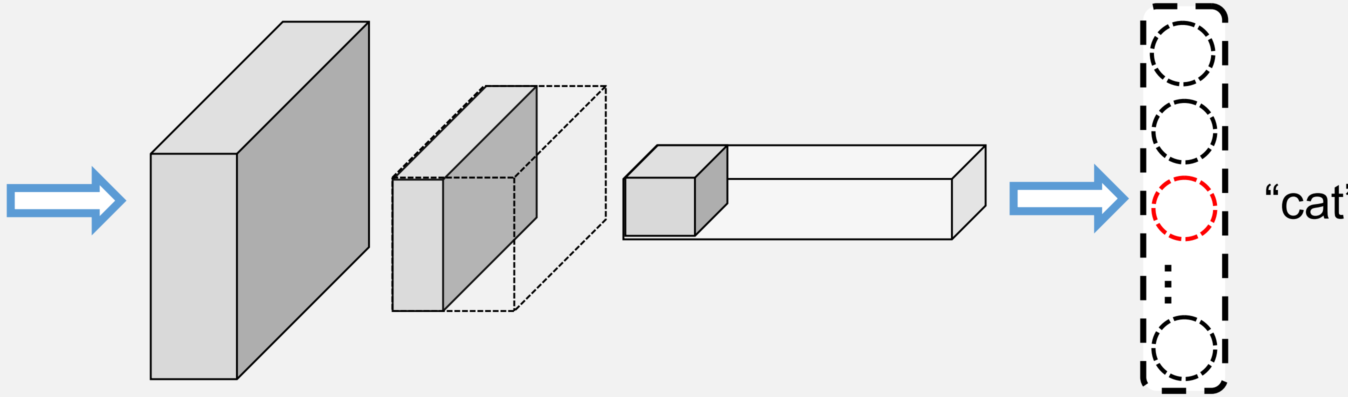
Learning Dynamic Networks with DRL



Learning Dynamic Networks with DRL

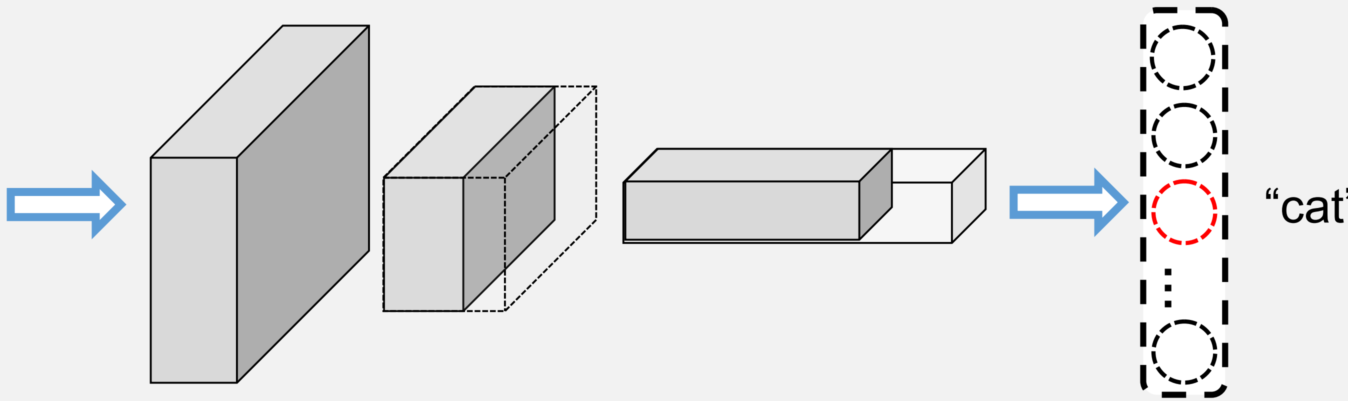


Learning Dynamic Networks with DRL



Easy task, 4x speed-up

Dynamic Network



Difficult task, 1.5x speed-up

Learning Dynamic Networks with DRL

Dynamic Network vs. Fixed Network

- ❖ Dynamic network can adjust complexity conditioned on the inputs
- ❖ Compared to the fixed compressed/quantized network, dynamic network can preserves the full ability of the original network
- ❖ The balance point between accuracy and speed is easily adjustable according to the available resources

Learning Dynamic Networks with DRL

Learning dynamic networks with DRL:

- Input image $F_0 = I$

  State: the set of executed operations x_t and the output features F_t

  Action: the next operation to be executed $a_t: (x_t, F_t) \rightarrow (x_{t+1}, F_{t+1})$

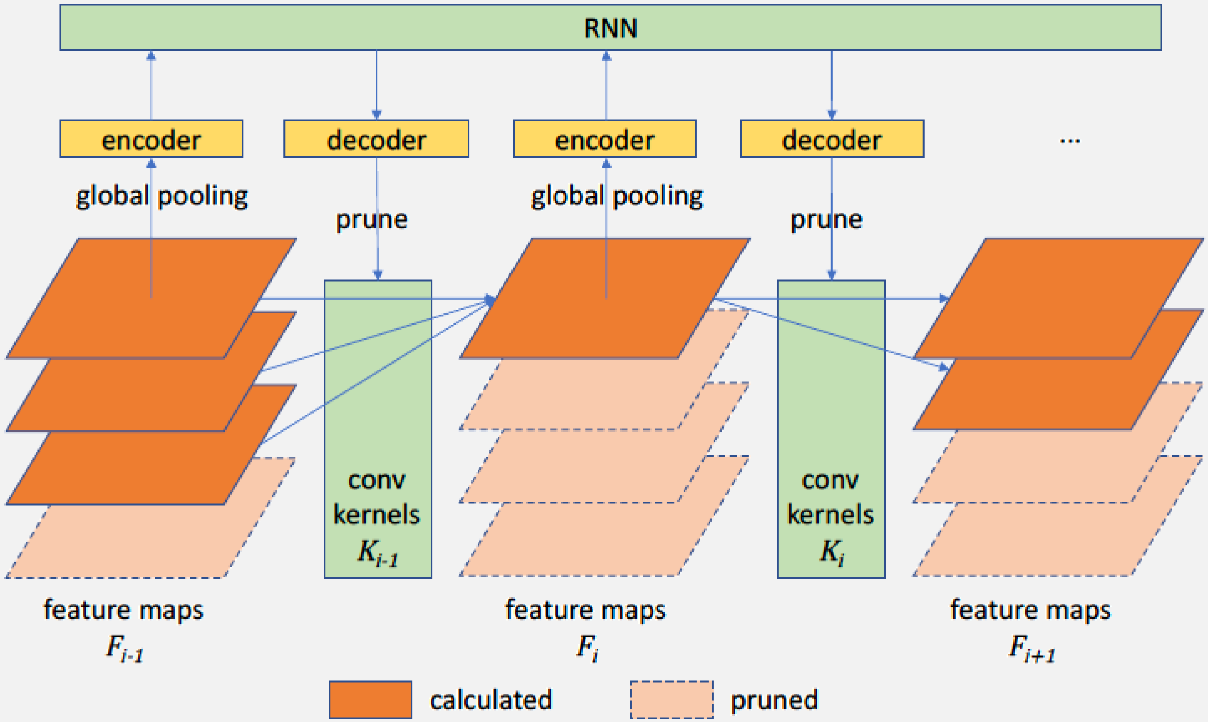
  $\mathrm{p}(\cdot\,|x, a)$:probability over next state(x_t, F_t)

  $q(\cdot\,|x, a)$:probability over rewards $R(x_t, F_t, a_t)$

  Reward: recognition performance (accuracy, CE loss)

Dynamic Pruning for Image Recognition

Neural Runtime Pruning (RNP) framework



Lin, Ji*, **Yongming Rao***, **Jiwen Lu**, and Jie Zhou. Runtime neural pruning. *NeurIPS*. 2017.

Dynamic Pruning for Image Recognition

Approach: Bottom-up Runtime Pruning

- Backbone CNN C with conv layers $C_1, C_2, \dots, C_m$, corresponding kernels $\boldsymbol{K}_1, \boldsymbol{K}_2, \dots, \boldsymbol{K}_m$, #channels n_i, producing feature maps $\boldsymbol{F}_1, \boldsymbol{F}_2, \dots, \boldsymbol{F}_m$, with size $n_i \times H \times W$.
- **Goal**: find and prune the redundant convolutional kernels in $\boldsymbol{K}_{i+1}$, given feature maps $\boldsymbol{F}_i, i = 1, 2, \dots, m-1$, to reduce computation and achieve maximum performance simultaneously.

$$\min_{\mathbf{K}_{i+1}, h} \mathbb{E}_{\mathbf{F}_i}[L_{cls}(\mathrm{conv}(\mathbf{F}_i, \mathbf{K}[h(\mathbf{F}_i)])) + L_{pnt}(h(\mathbf{F}_i))],$$

L_{cls} - classification loss, L_{pnt} - computation penalty.

Dynamic Pruning for Image Recognition

Approach: Layer-by-layer MDP

- **State**: Given feature map $\boldsymbol{F}_i$, extract dense feature embedding $p_{\boldsymbol{F}_i}$ with global pooling, and use a encoder E, to project into a fixed length embedding $E(p_{\boldsymbol{F}_i})$.
- **Action**: actions for each pruning are defined in an incremental way: taking actions a_i yields calculating the feature map groups $\boldsymbol{F}'_1, \boldsymbol{F}'_2, \ldots, \boldsymbol{F}'_i, i = 1,2, \ldots, k$.
- **Reward**: The reward of each action taken at the t-th step with action a_i is defined as:

$$r_t(a_i) = \begin{cases} -\alpha L_{cls} + (i-1) \times p, & \text{if inference terminates } (t = m-1), \\ (i-1) \times p, & \text{otherwise } (t < m-1) \end{cases}$$

Dynamic Pruning for Image Recognition

RNP model is alternatively optimized:

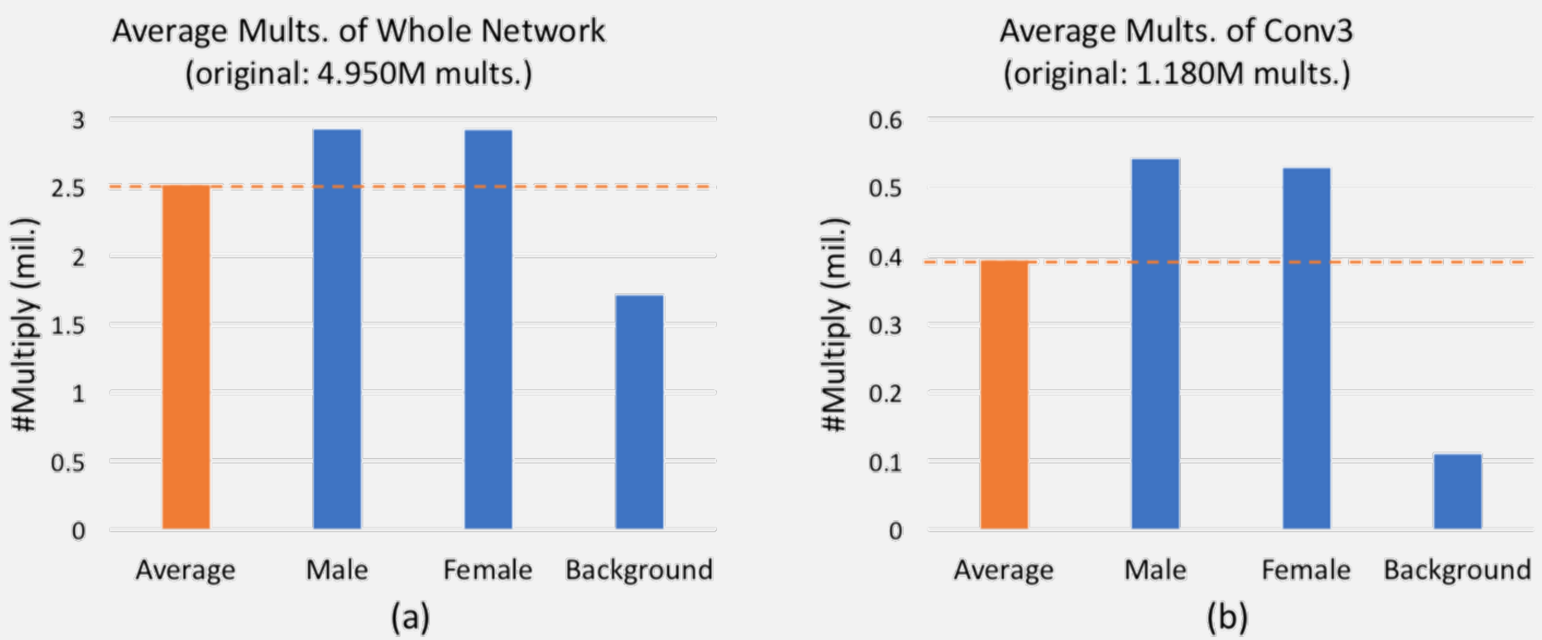
Algorithm 1 Runtime neural pruning for solving optimization problem (1):

Input: training set with labels $\{X\}$
Output: backbone CNN C, decision network D

initialize: train C in normal way or initialize C with pre-trained model
for $i \leftarrow 1, 2, ..., M$ **do**
    // train decision network
    for $j \leftarrow 1, 2, ..., N_1$ **do**
        Sample random minibatch from $\{X\}$
        Forward and sample ϵ-greedy actions $\{s_t, a_t\}$
        Compute corresponding rewards $\{r_t\}$
        Backward Q values for each stage and generate $\nabla_\theta L_{re}$
        Update θ using $\nabla_\theta L_{re}$
    end for
    // fine-tune backbone CNN
    for $k \leftarrow 1, 2, ..., N_2$ **do**
        Sample random minibatch from $\{X\}$
        Forward and calculate L_{cls} after runtime pruning by D
        Backward and generate $\nabla_C L_{cls}$
        Update C using $\nabla_C L_{cls}$
    end for
end for
return C and D

Dynamic Pruning for Image Recognition

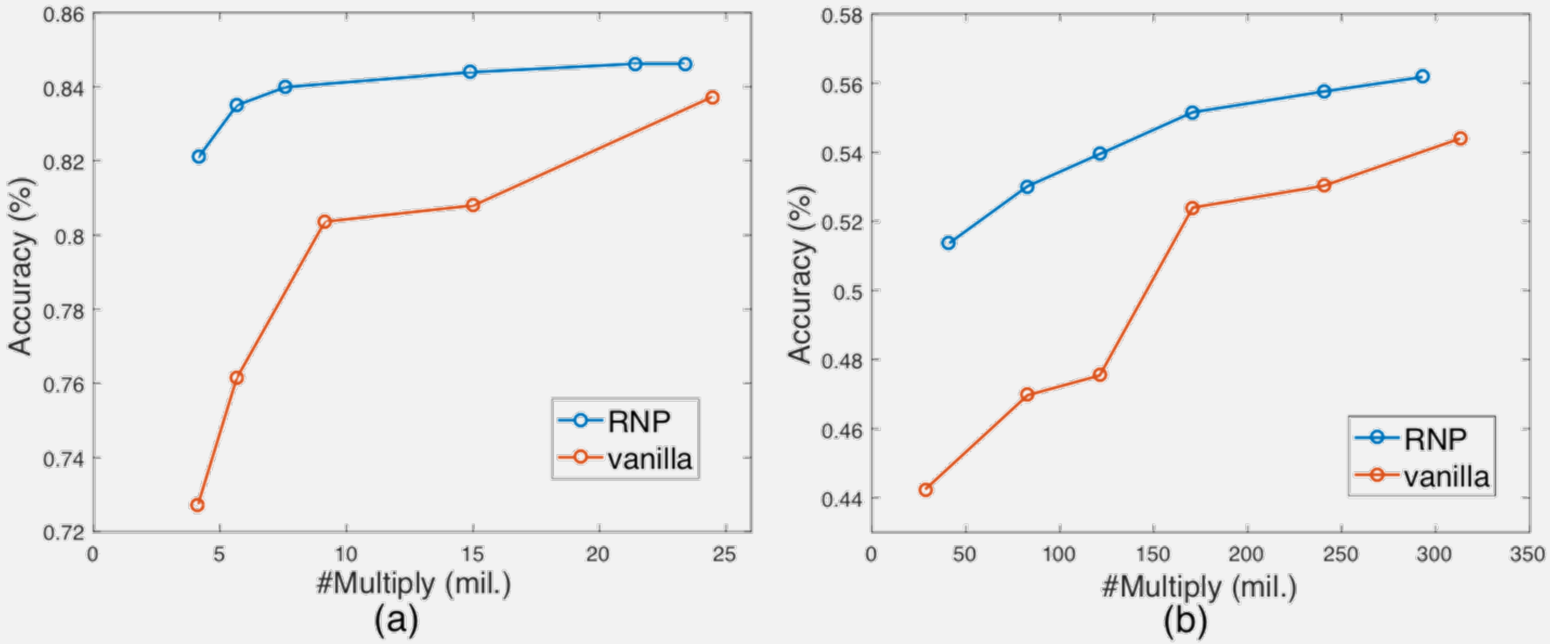
Intuitive 3-class classification experiment on LFW-T



(a) (b)

Dynamic Pruning for Image Recognition

Results on CIFAR10 and CIFAR-100



Dynamic Pruning for Image Recognition

Speed-up (in FLOPs)	3×	4×	5×	10×
Jaderberg *et al.* [26] ([69]'s implementation)	2.3	9.7	29.7	-
Asymmetric [69]	-	3.84	-	-
Filter pruning [36] (our implementation)	3.2	8.6	14.6	-
Taylor expansion [45]	2.3	4.8	-	-
ThiNet [43]	1.98	-	-	7.94
Ours	2.32	3.23	3.58	4.89

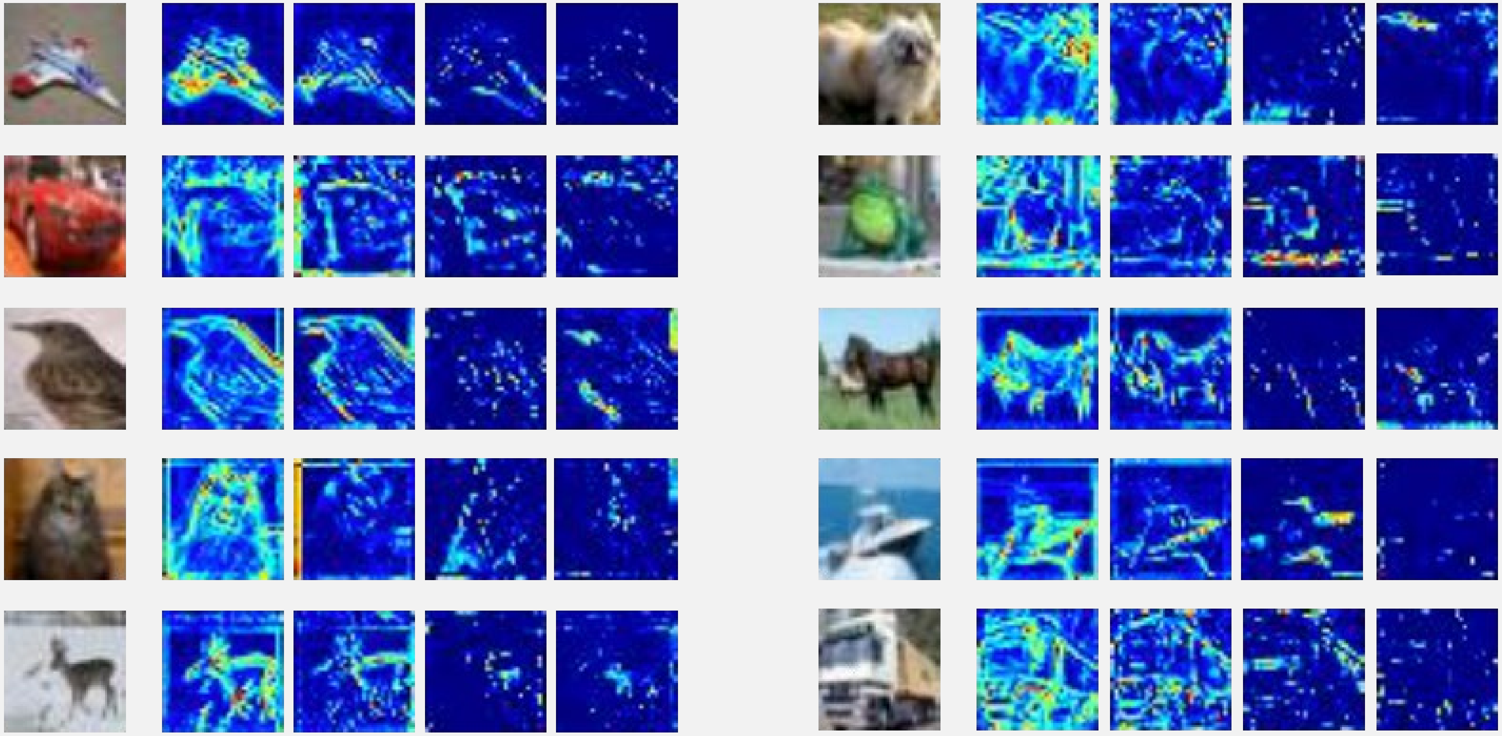
Comparisons of increase of top-5 error on ILSVRC2012-val (%) with recent state-of-the-arts. *(base top-5 error: 10.1%)*

The increase of top-1/top-5 error (%) and GPU inference time (ms) under different theoretical speed-up ratios on the ILSVRC2012-val set.

Speed-up solution	Δtop-1/top-5 err.	Inference time
VGG-16 (1×)	0/0	3.26 (1.0×)
RNP-VGG-16 (3×)	2.98/2.32	1.38 (2.3×)
RNP-VGG-16 (4×)	4.01/3.23	1.07 (3.0×)
RNP-VGG-16 (5×)	4.88/3.58	0.880 (3.7×)
RNP-VGG-16 (10×)	6.12/4.89	0.554 (5.9×)
ResNet-50 (1×)	0/0	2.54 (1.0×)
RNP-ResNet-50 (2×)	2.90/2.14	1.94 (1.31×)
RNP-ResNet-50 (3×)	5.21/3.66	1.68 (1.51×)

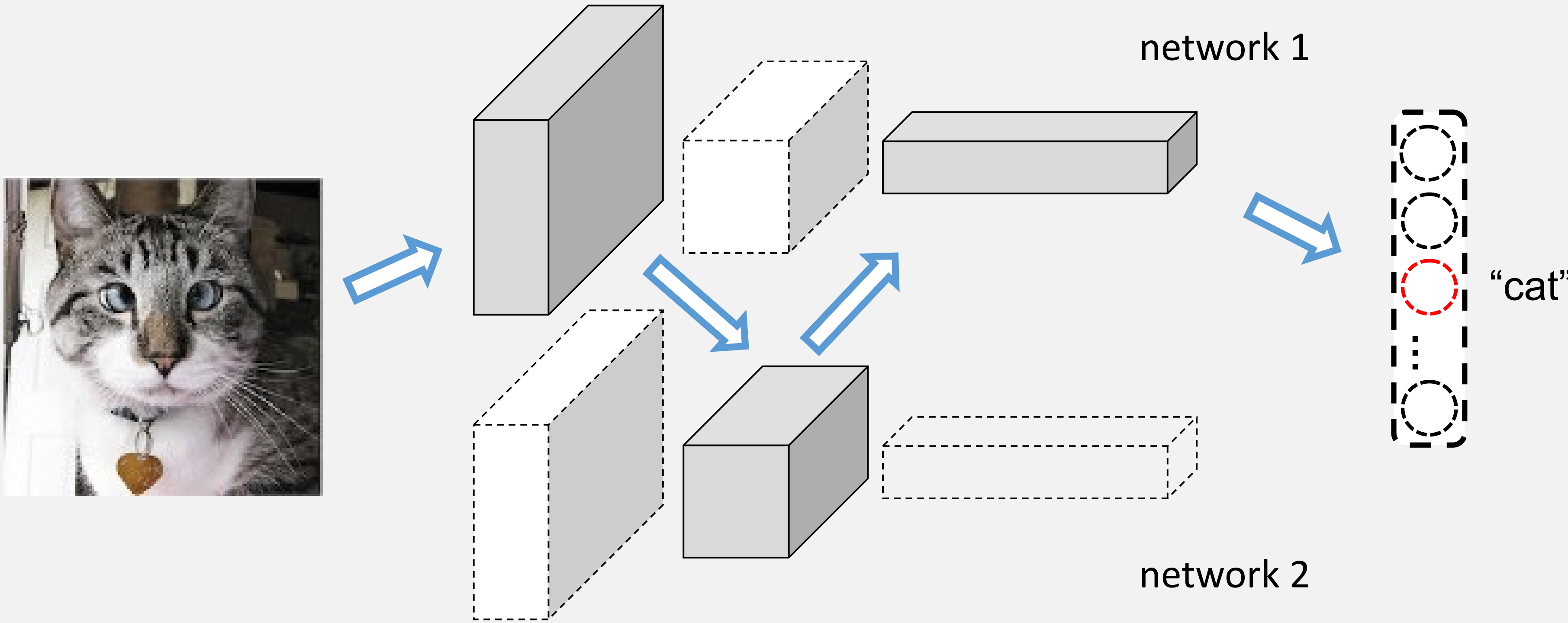
Dynamic Pruning for Image Recognition

Feature map visualization



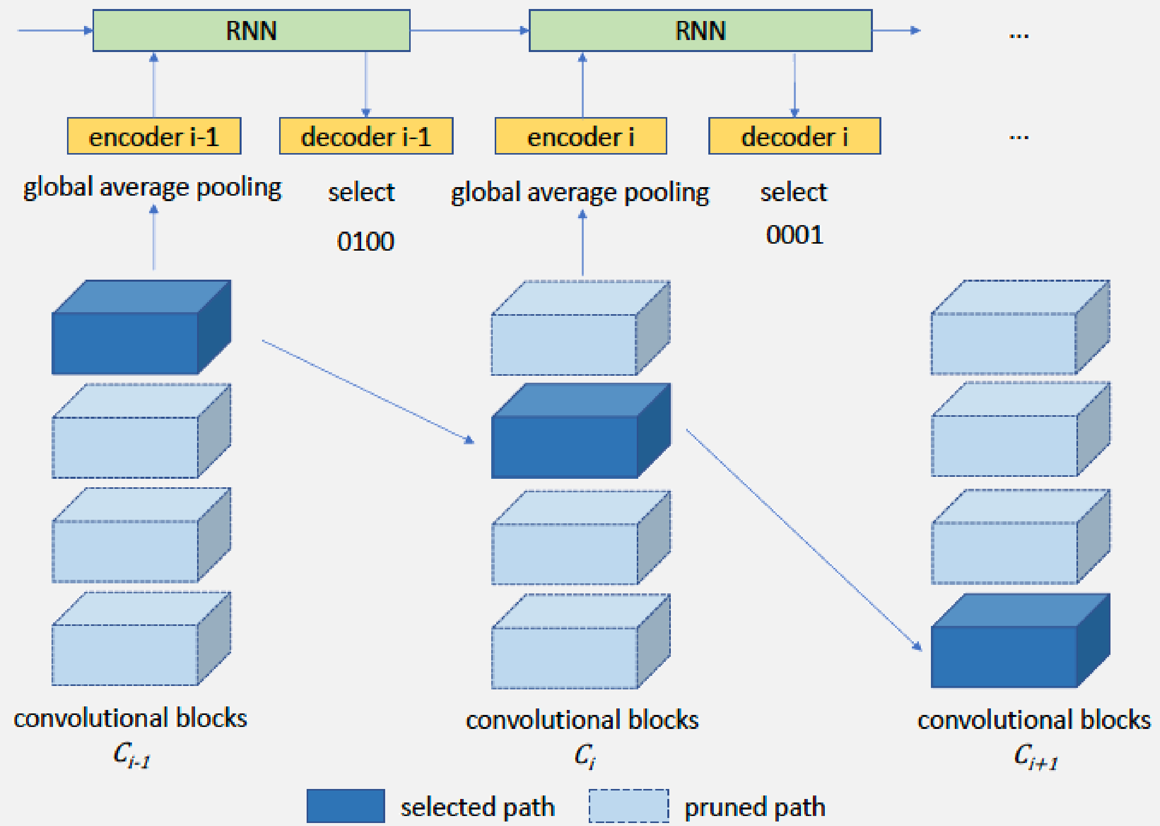
Dynamic Routing in Convolutional Networks

Runtime Network Routing aims at learning to selects an optimal path inside the network during inference conditioned on the input image



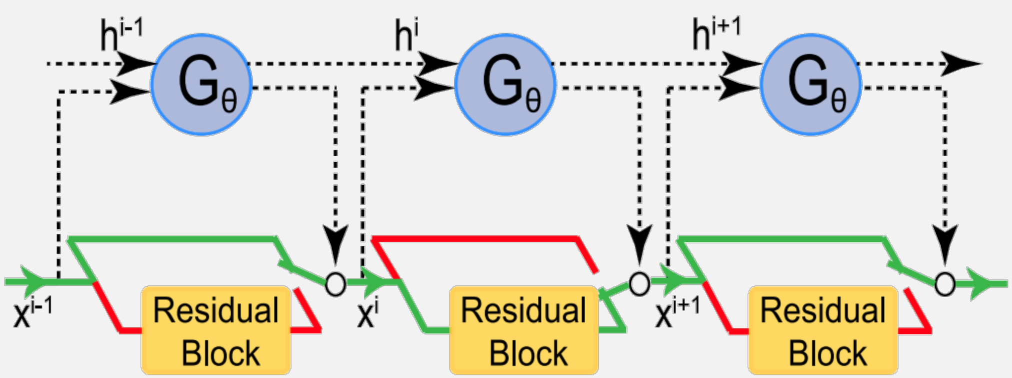
Yongming Rao, Jiwen Lu, Ji Lin, and Jie Zhou. "Runtime Network Routing for Efficient Image Classification." *T-PAMI,* 2019.

Dynamic Routing in Convolutional Networks



Dynamic Skipping in Convolutional Networks

SkipNet learns to skip convolutional layers on a per-input basis



recurrent module is used to learn cross-layer dependency

Wang X, Yu F, Dou Z Y, et al. Skipnet: Learning dynamic routing in convolutional networks. ECCV, 2018.

Dynamic Skipping in Convolutional Networks

Hybrid RL algorithm to learn policies and backbone CNN ***simultaneously***

$$\begin{aligned}\nabla_\theta \mathcal{J}(\theta) &= \mathbb{E}_{\mathbf{x}} \nabla_\theta \sum_{\mathbf{g}} p_\theta(\mathbf{g}|\mathbf{x}) L_\theta(\mathbf{g}, \mathbf{x}) \\ &= \mathbb{E}_{\mathbf{x}} \sum_{\mathbf{g}} p_\theta(\mathbf{g}|\mathbf{x}) \nabla_\theta \mathcal{L} + \mathbb{E}_{\mathbf{x}} \sum_{\mathbf{g}} p_\theta(\mathbf{g}|\mathbf{x}) \nabla_\theta \log p_\theta(\mathbf{g}|\mathbf{x}) L_\theta(\mathbf{g}, \mathbf{x}) \\ &= \mathbb{E}_{\mathbf{x}} \mathbb{E}_{\mathbf{g}} \nabla_\theta \mathcal{L} - \mathbb{E}_{\mathbf{x}} \mathbb{E}_{\mathbf{g}} \sum_{i=1}^{N} \nabla_\theta \log p_\theta(g_i|\mathbf{x}) r_i .\end{aligned}$$

Algorithm 1: Hybrid Learning Algorithm (HRL+SP)

Input: A set of images $\mathbf{x}$ and labels $\mathbf{y}$

Output: Trained SkipNet

1. Supervised pre-training (Sec. 3.3)

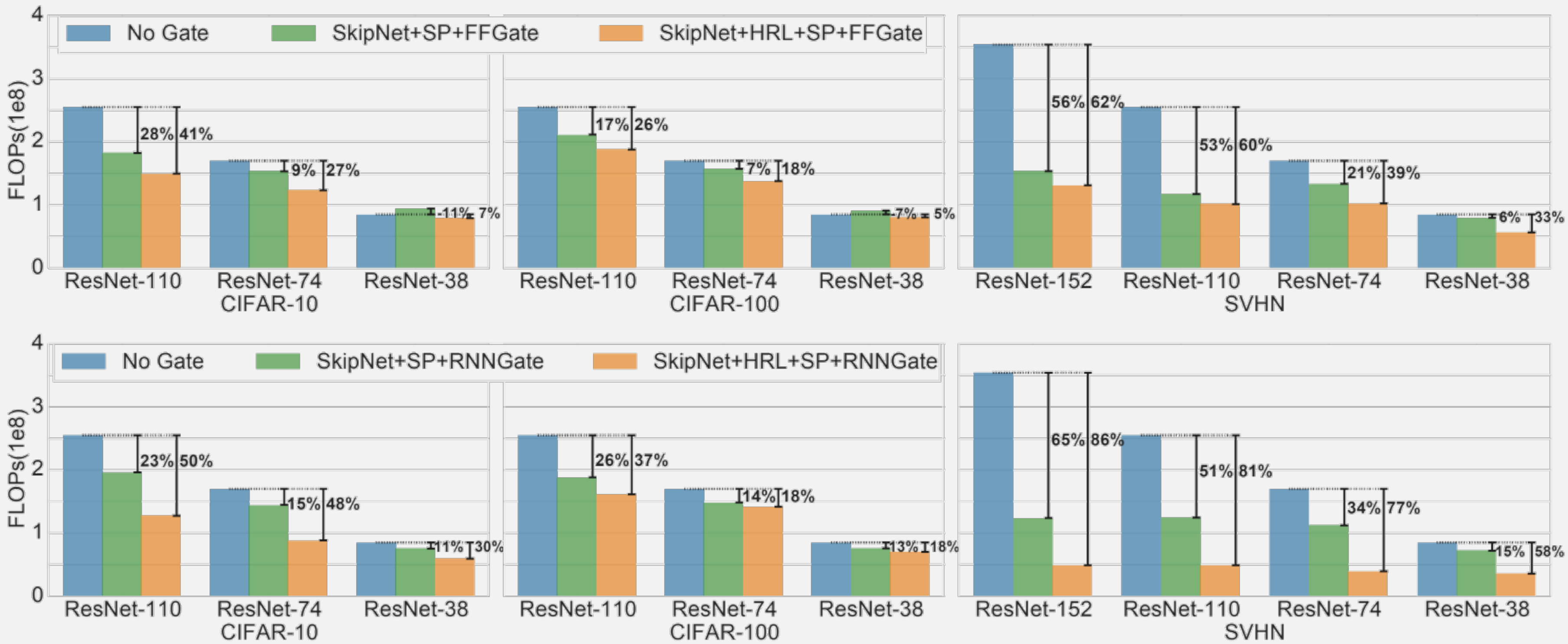
$\theta_{SP} \leftarrow \text{SGD}(L_{\text{Cross-Entropy}}, \text{SkipNet-}G_{\text{relax}}(\mathbf{x}))$

2. Hybrid reinforcement learning (Sec. 3.2)

Initialize θ_{HRL+SP} with θ_{SP}

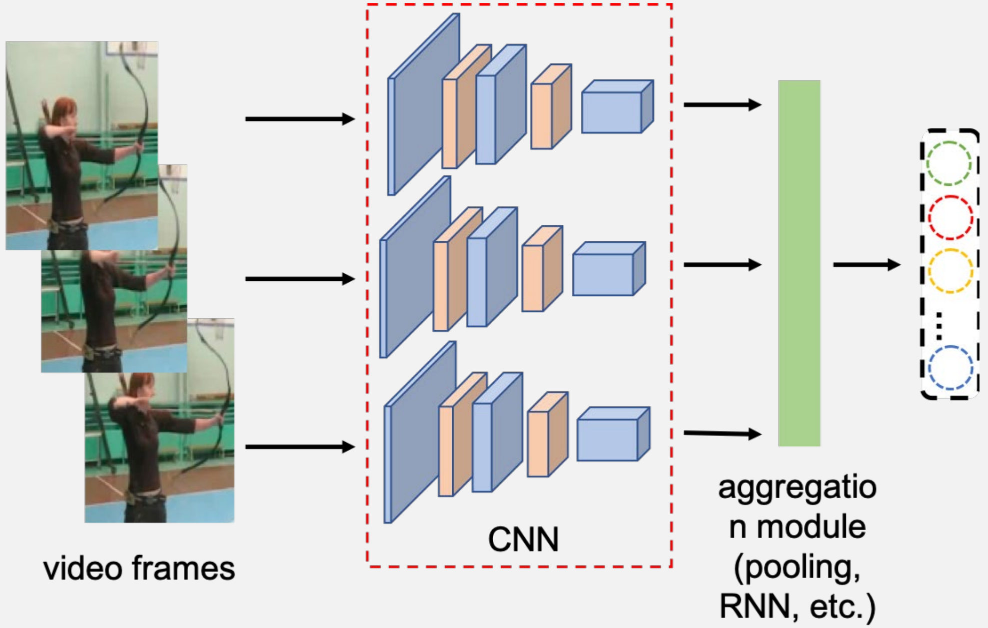
$\theta_{HRL+SP} \leftarrow \text{REINFORCE}(\mathcal{J}, \text{SkipNet-}G(\mathbf{x}))$

Dynamic Skipping in Convolutional Networks



Dynamic Networks for Video Understanding

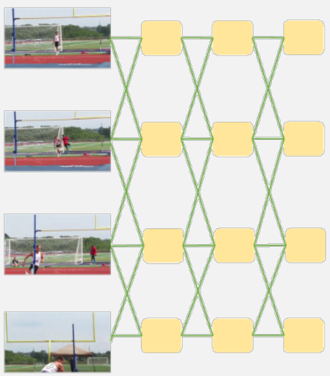
Neural Networks for Video Classification



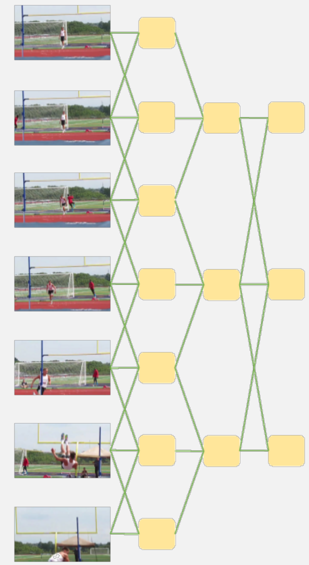
- The cost of video recognition model is ***linear*** to the number of input frames, which has become the crucial factor in determining the overall computation
- ***T times*** computational cost compared to image classification.

Dynamic Networks for Video Understanding

Two existing strategies to reduce temporal computational cost:



uniformly sample frames

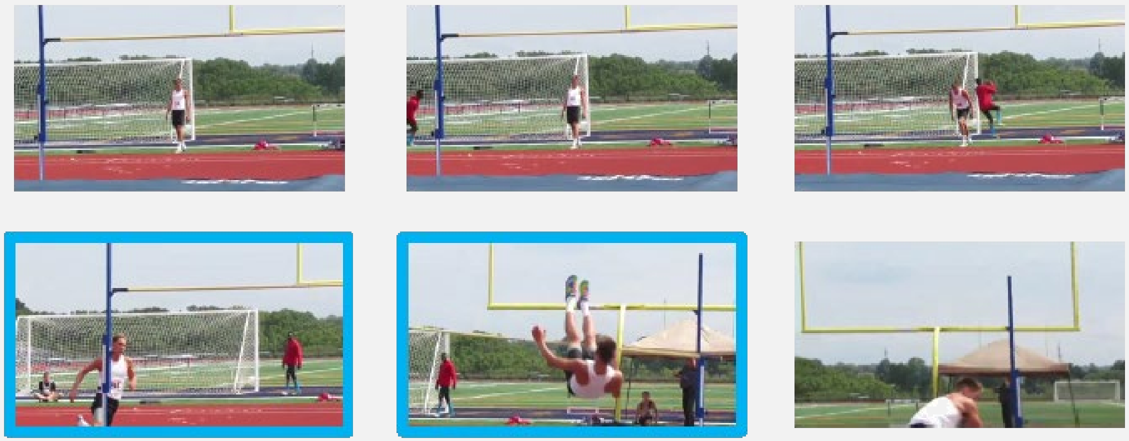


perform pooling on feature maps along temporal dimension

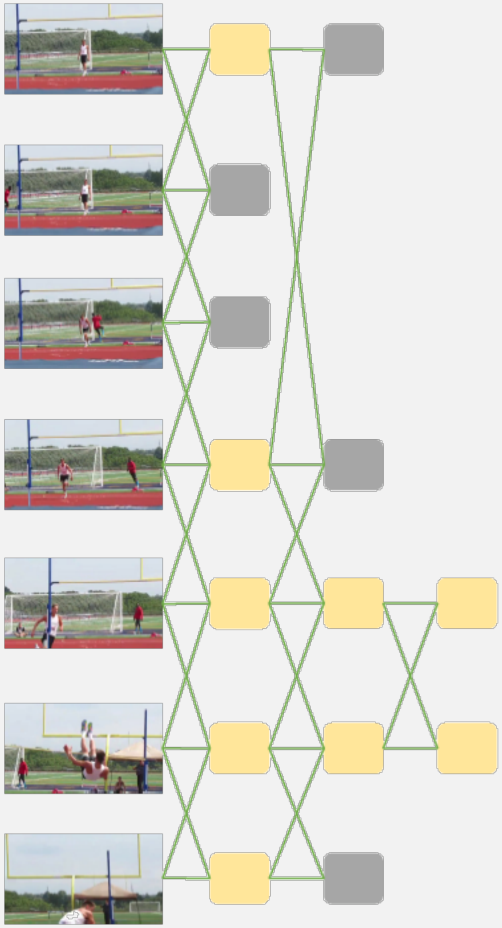
Both strategies assume:

- ❖ frames inside a video are of equal importance
- ❖ all videos are of equal importance

Dynamic Networks for Video Understanding

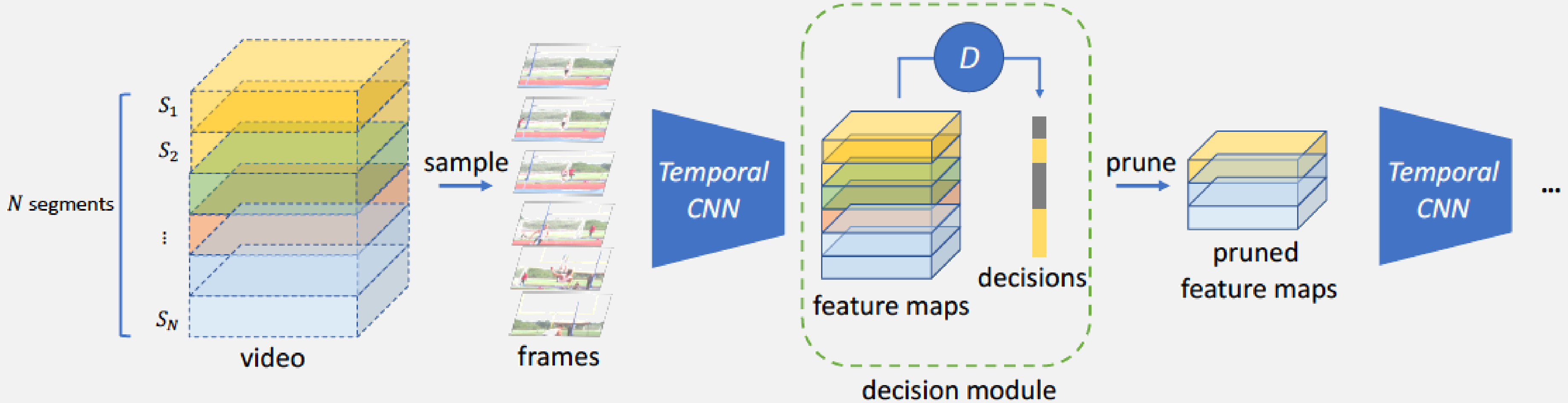


A subset of most informative frames (in blue boxes) is sufficient to understand this video. Therefore, video frames should be pruned ***non-uniformly*** and ***dynamically***



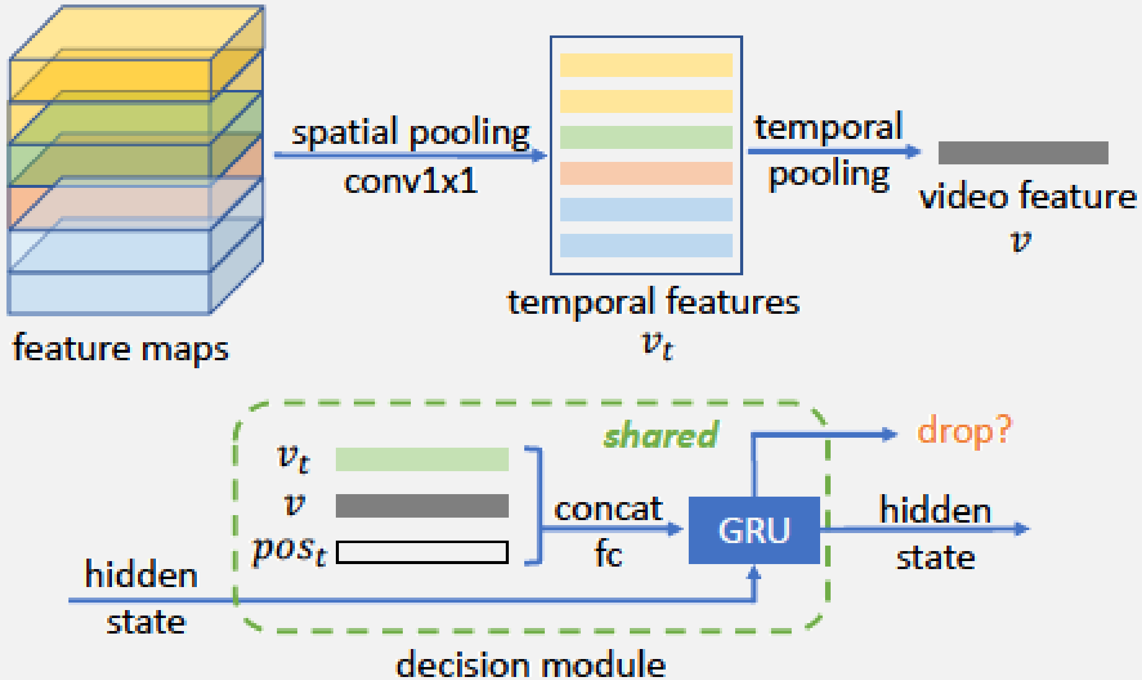
Dynamic Networks for Video Understanding

Dynamic Progressive Pruning proposes to insert a decision module at the beginning of each stage to selectively prune less informative frames conditioned on feature maps produced by previous layer.



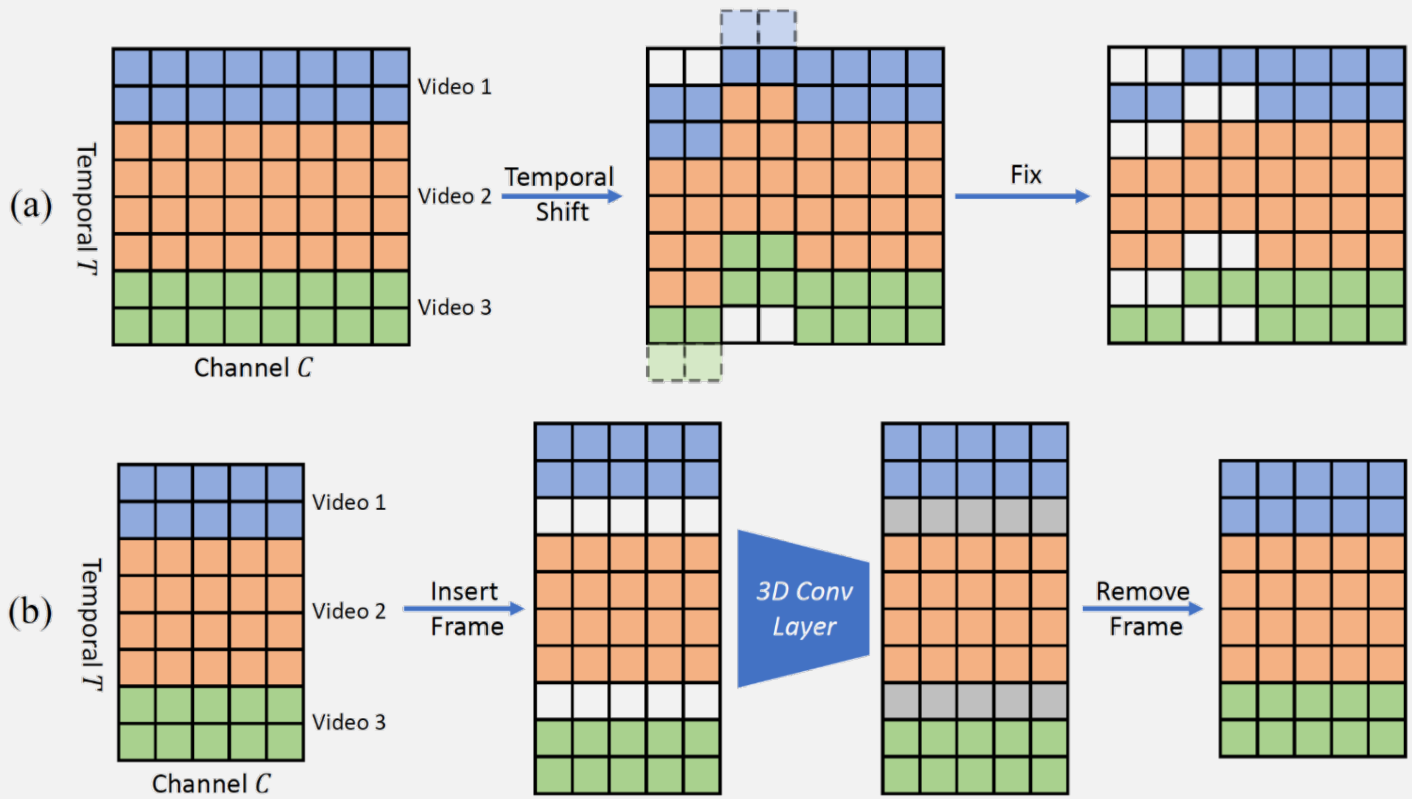
Yongming Rao, Ji Lin, **Jiwen Lu**, Jie Zhou. Dynamic Progressive Pruning for Efficient Video Classification. 2019.

Dynamic Networks for Video Understanding



recurrent decision module

Dynamic Networks for Video Understanding



dynamic video batching method for state-of-the-art TSM model and 3D convolution

Dynamic Networks for Video Understanding

Kinetics

Method	Backbone	#Frame	FLOPs	#Param	Top-1	Top-5
TSN [39]	BNInception	25	50G	24.3M	69.1	88.7
TSN [39] (our impl.)	ResNet-50	8	33G	24.3M	68.8	88.3
ECO [47]	BNInception + 3D ResNet-18	8	32G	47.5M	67.8	-
ECO Lite [47]	BNInception + 3D ResNet-18	16	47G	37.5M	64.4	-
TSM-8f [21]	ResNet-50	8	33G	24.3M	70.6	89.5
TSM-16f P [21]	ResNet-50	16	39G	24.3M	70.9	89.7
Ours (2×)	ResNet-50	16	35G	24.5M	**71.7**	**90.2**
TSN [39] ([45]'s impl.)	BNInception	8	16G	10.7M	63.3	-
TSN [39] (our impl.)	ResNet-50	5	21G	24.3M	67.9	87.6
TRN-Multiscale [45]	BNInception	8	16G	18.3M	63.2	-
ECO [47]	BNInception + 3D ResNet-18	4	16G	47.5M	66.2	-
TSM-4f [21]	ResNet-50	4	17G	24.3M	68.2	87.9
TSM-16f P [21]	ResNet-50	16	20G	24.3M	67.2	87.5
TSM-8f P [21]	ResNet-50	8	19G	24.3M	68.4	88.0
Ours (3.3×)	ResNet-50	16	19G	24.5M	**69.8**	**89.1**

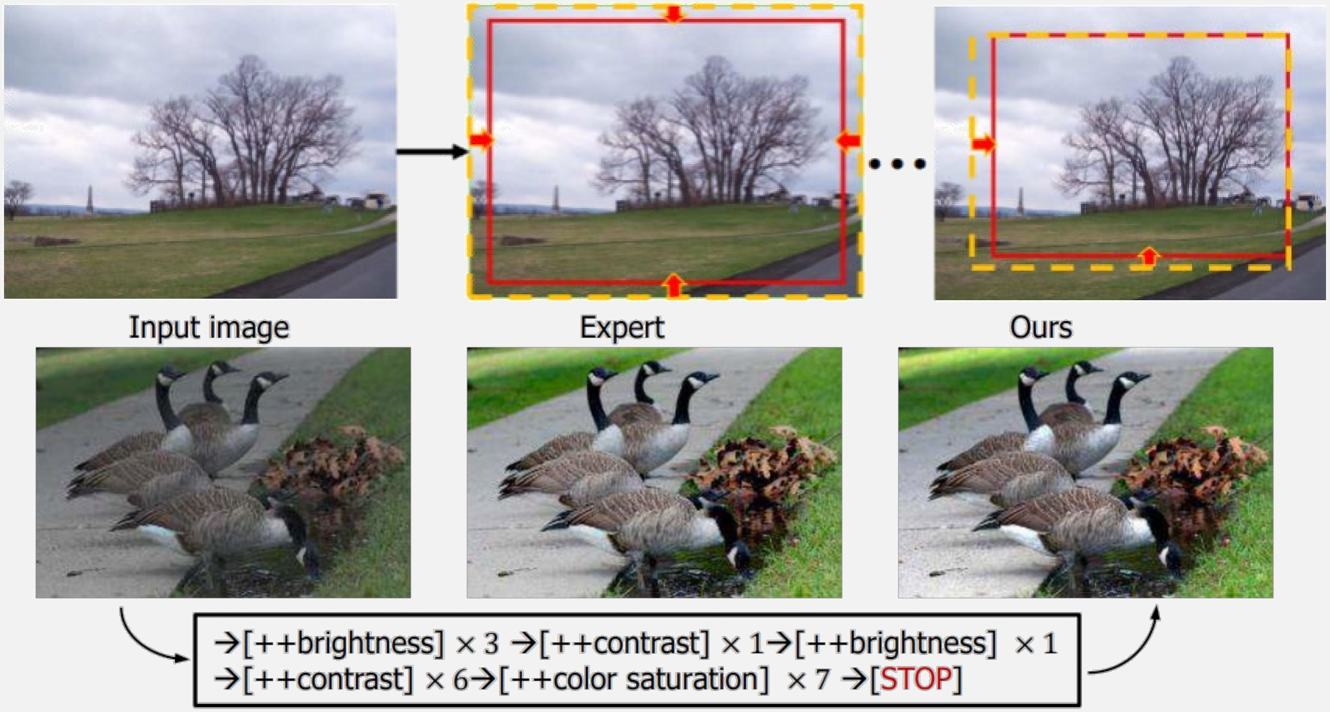
UCF-101 & HMDB-51

Method	Backbone	#Frame	FLOPs	#Param	UCF-101	HMDB-51
TSN [39] ([45]'s impl.)	BNInception	8	16G	10.7M	82.69	-
TRN-Multiscale [45]	BNInception	8	16G	18.3M	83.8	-
ECO [47]	BNInception + 3D ResNet-18	4	16G	47.5M	87.4	58.1
TSM-4f [21]	ResNet-50	4	17G	24.3M	92.1	66.6
Ours (2×)	ResNet-50	8	17G	24.5M	**93.2**	**67.8**

Part 4:
DRL for Image Editing & Understanding

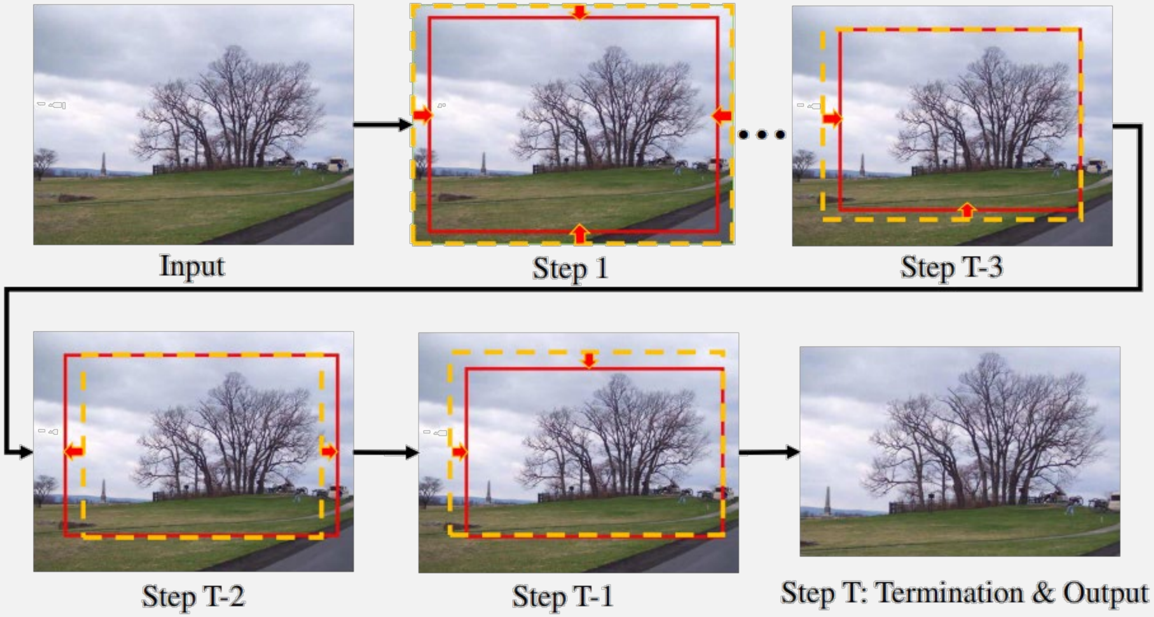
DRL for Image Editing

- Image Cropping & Alignment
- Image Super-resolution & Enhancement

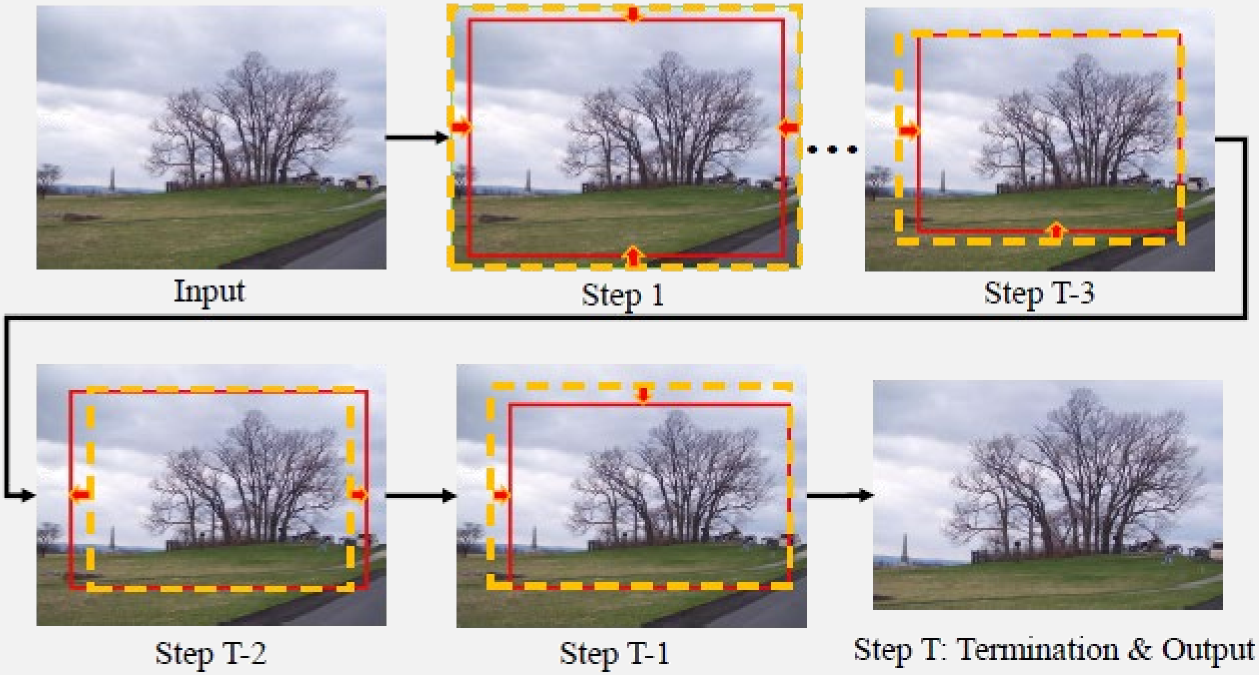


DRL for Image Editing

- Original Image I_0
  State: current image I_t and editing history $x_t = (x_{t-1}, I_t)$
  Action: operation on image $a_t: I_t \rightarrow I_{t+1}$, a $\in A$
  $p(\cdot\,|x,a)$:probability over next state x_{t+1}
  $q(\cdot\,|x,a)$:probability over rewards $R(x_t,a_t)$



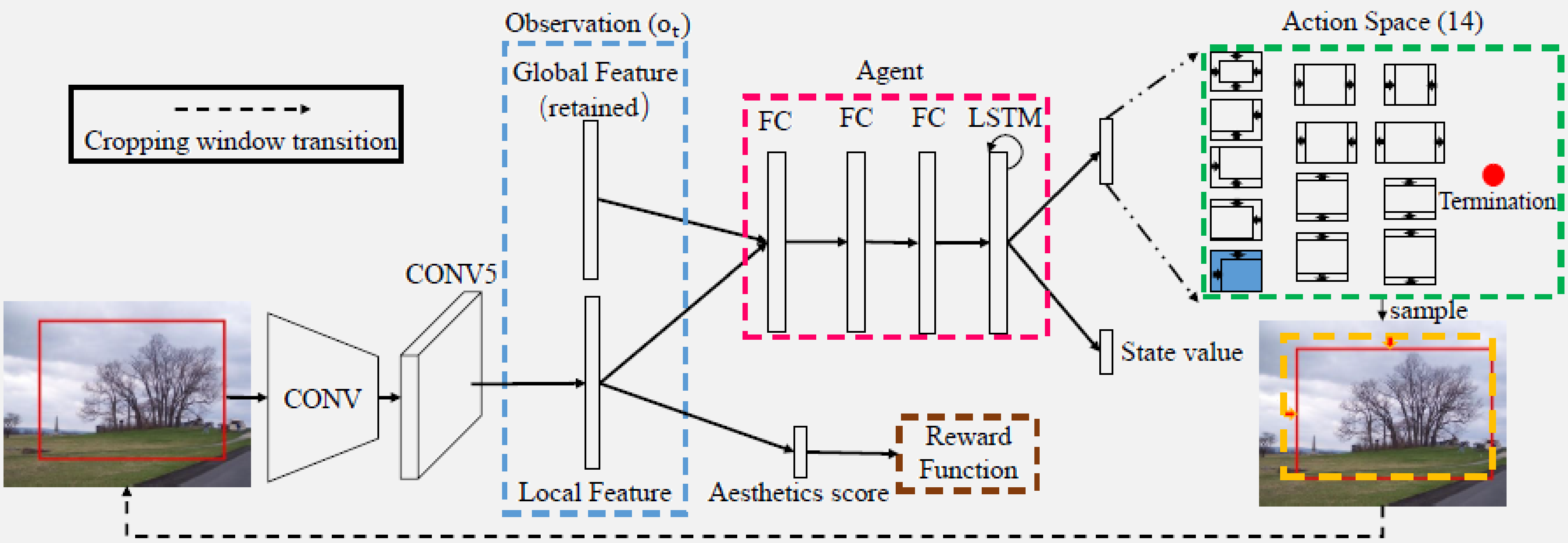
DRL for Image Cropping



Li, Debang, et al. "A2-RL: aesthetics aware reinforcement learning for image cropping." *CVPR*. 2018.

DRL for Image Cropping

$$r'_t = sign(s_{aes}(I_{t+1}) - s_{aes}(I_t)) - 0.001 * (t+1)$$



DRL for Image Cropping

Method	Annotation I		Annotation II		Annotation III	
	Avg IoU	Avg Disp Error	Avg IoU	Avg Disp Error	Avg IoU	Avg Disp Error
eDN [27]	0.4636	0.1578	0.4399	0.1651	0.4370	0.1659
RankSVM+DeCAF$_7$ [4]	0.6643	0.092	0.6556	0.095	0.6439	0.099
LearnChange [29]	0.7487	0.0667	0.7288	0.0720	0.7322	0.0719
VFN+SW [5]	0.7401	0.0693	0.7187	0.0762	0.7132	0.0772
A2-RL w/o *nr*	0.6841	0.0852	0.6733	0.0895	0.6687	0.0895
A2-RL w/o LSTM	0.7855	0.0569	0.7847	0.0578	0.7711	0.0578
A2-RL(Ours)	**0.8019**	**0.0524**	**0.7961**	**0.0535**	**0.7902**	**0.0535**

Table 2. Cropping accuracy on CUHK Image Cropping Dataset [29]. The best results are highlighted in bold.

Method	Avg IoU	Avg Disp Error
eDN [27]	0.4857	0.1372
RankSVM+DeCAF$_7$ [4]	0.6019	0.1060
VFN+SW [5]	0.6328	0.0982
A2-RL w/o *nr*	0.5720	0.1178
A2-RL w/o LSTM	0.6310	0.1014
A2-RL(Ours)	**0.6633**	**0.0892**

Table 1. Cropping accuracy on Flickr Cropping Dataset [4]. The best results are highlighted in bold.

Method	Top-1 Max IoU
Fang *et al.* [9]	0.6998
Kao *et al.* [11]	0.7500
A2-RL w/o *nr*	0.7089
A2-RL w/o LSTM	0.7960
A2-RL(Ours)	**0.8204**

Table 3. Cropping accuracy on Human Cropping Dataset [9]. The best results are highlighted in bold.

DRL for Color Enhancement

→[++brightness] × 3 →[++contrast] × 1→[++brightness] × 1
→[++contrast] × 6→[++color saturation] × 7 →[STOP]

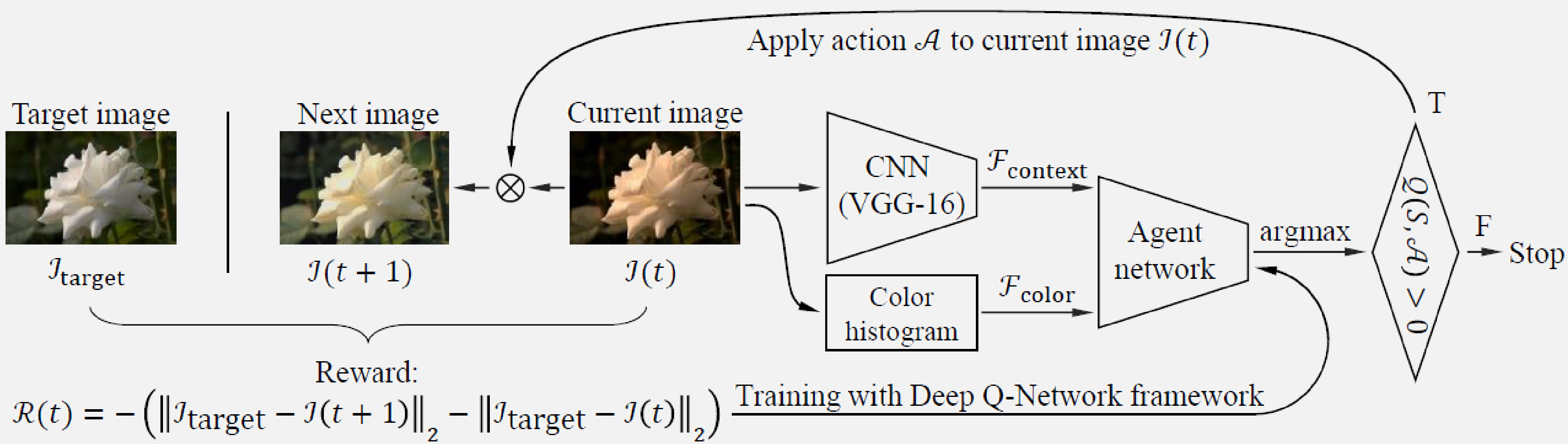
Park, J., Lee, J. Y., Yoo, D., & So Kweon, I. Distort-and-recover: Color enhancement using deep reinforcement learning. CVPR, 2018.

DRL for Color Enhancement

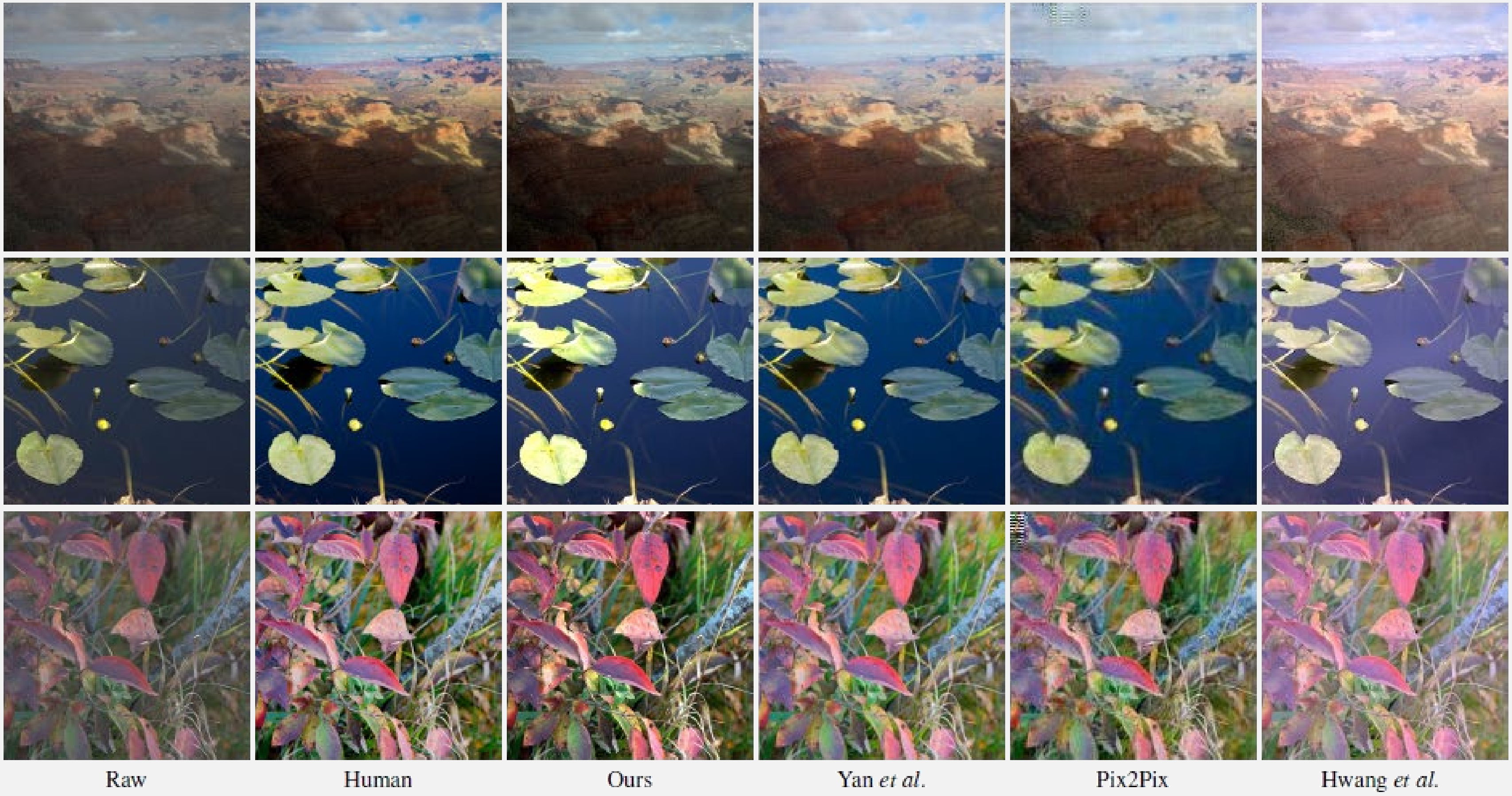
$$\mathcal{R}(t) = \|\mathcal{I}_{\text{target}} - \mathcal{I}(t-1)\|^2 - \|\mathcal{I}_{\text{target}} - \mathcal{I}(t)\|^2$$

$$\mathcal{Q}(\mathcal{S}(t), \mathcal{A}) = E\left[r(t) + \gamma \cdot r(t+1) + \gamma^2 \cdot r(t+2) + \cdots\right]$$

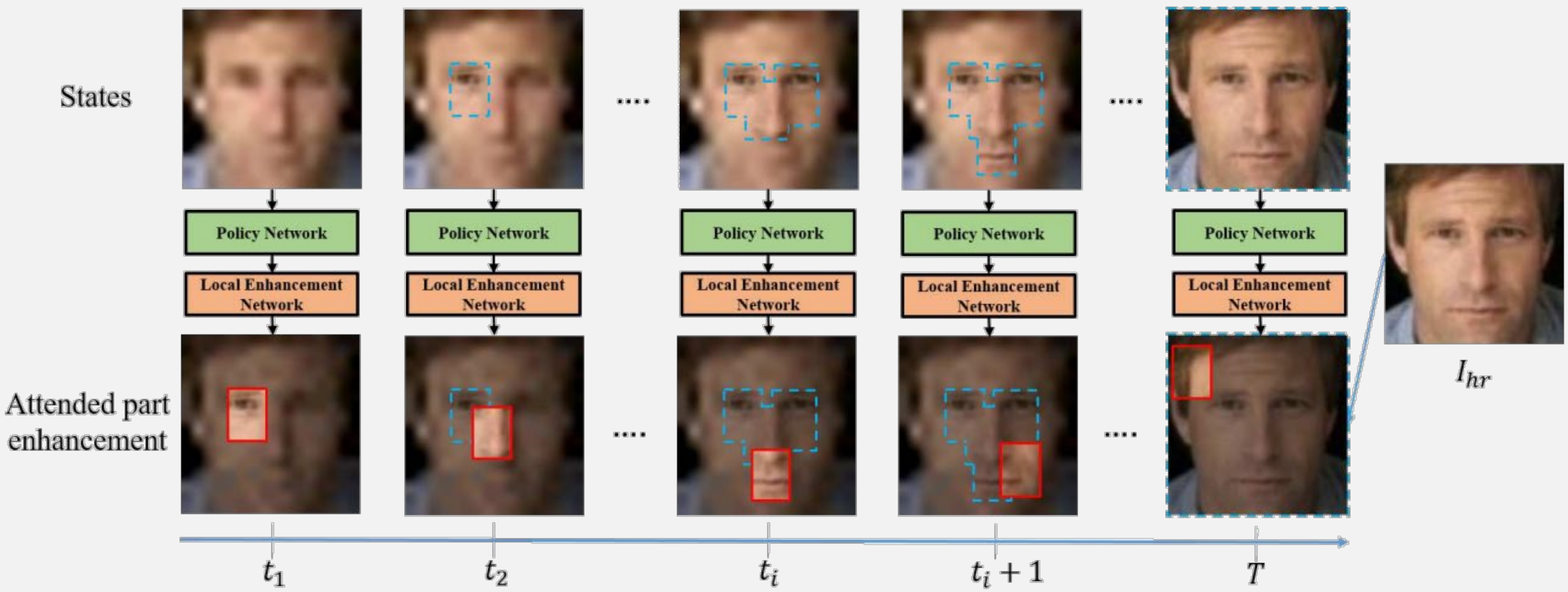
#	Action description	#	Action description
1	$\downarrow$ contrast ($\times 0.95$)	2	$\uparrow$ contrast ($\times 1.05$)
3	$\downarrow$ color saturation ($\times 0.95$)	4	$\uparrow$ color saturation ($\times 1.05$)
5	$\downarrow$ brightness ($\times 0.95$)	6	$\uparrow$ brightness ($\times 1.05$)
7	$\downarrow$ red and green ($\times 0.95$)	8	$\uparrow$ red and green ($\times 1.05$)
9	$\downarrow$ green and blue ($\times 0.95$)	10	$\uparrow$ green and blue ($\times 1.05$)
11	$\downarrow$ red and blue ($\times 0.95$)	12	$\uparrow$ red and blue ($\times 1.05$)



DRL for Color Enhancement



DRL for Face Hallucination

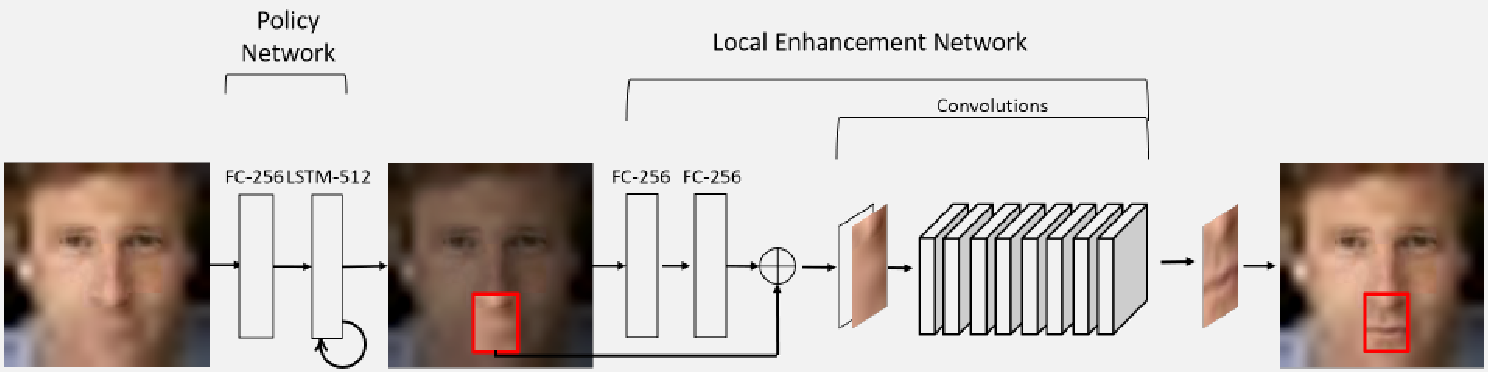


Cao, Q., Lin, L., Shi, Y., Liang, X., & Li, G. Attention-aware face hallucination via deep reinforcement learning. CVPR, 2017.

DRL for Face Hallucination

❖ Recurrent Policy Network

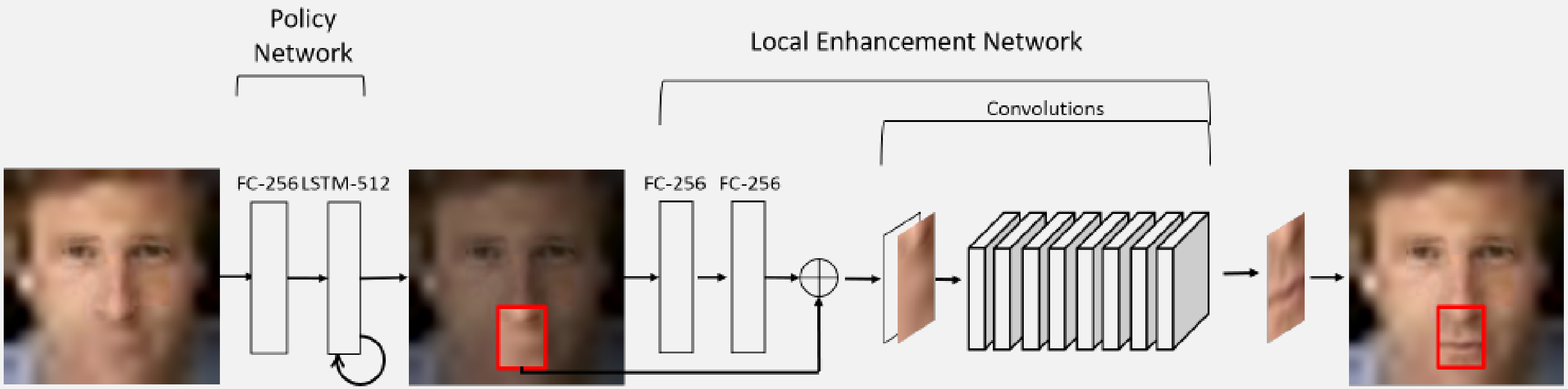
- State:
  1) the enhanced hallucinated face image I_t from previous step
  2) the latent variable h_t obtained by forwarding the encoded history action vector h_{t-1} into the LSTM layer
- Action: selecting one region from all possible locations
- Reward: $r_t = \begin{cases} 0 & t < T \\ -L_{\theta_\pi} & t = T \end{cases}$ $L_{\theta_\pi} = E_{p(I;\pi)}[\|I_{hr} - I_T\|_2], \ \ \gamma = 1$



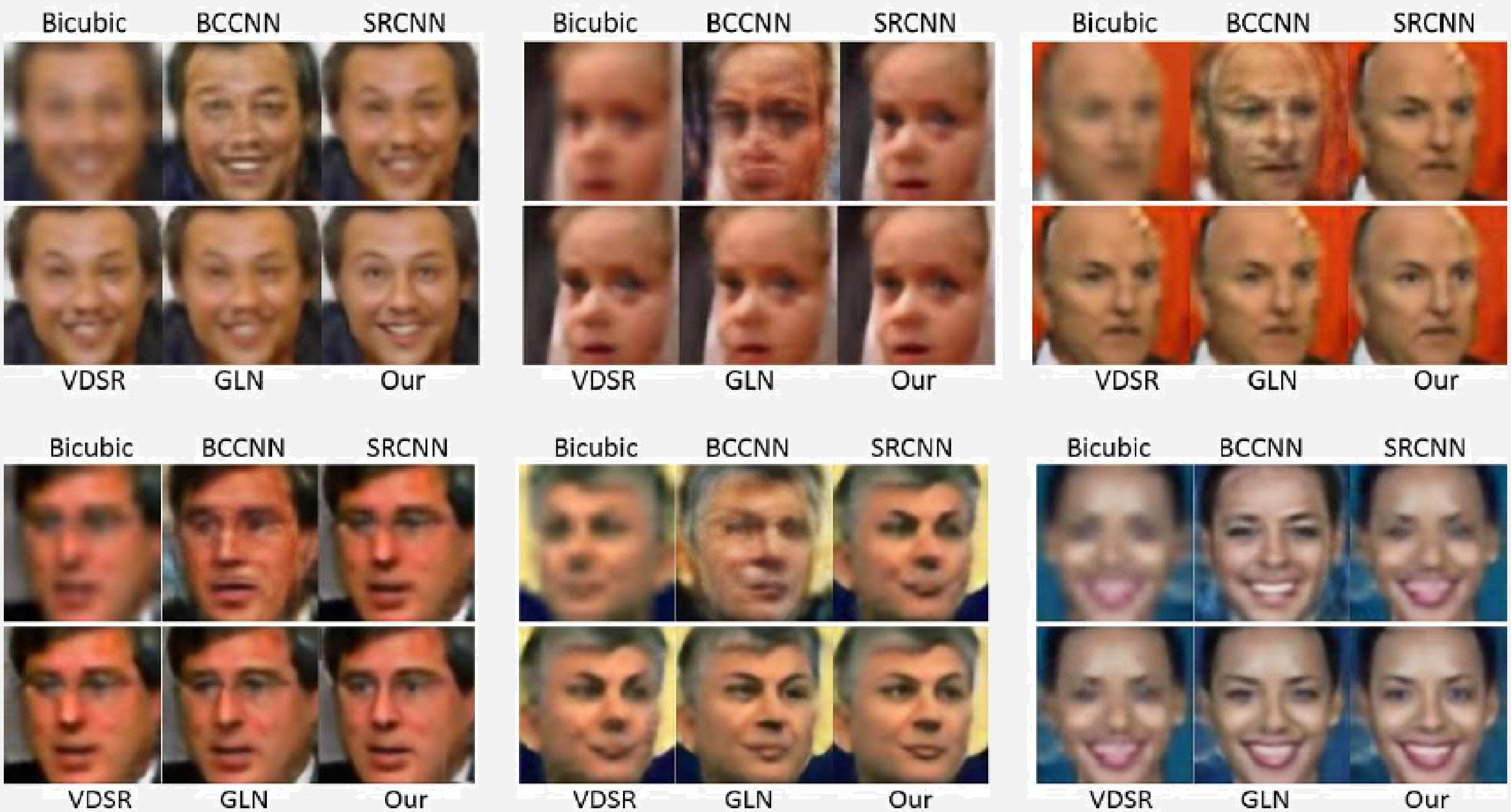
DRL for Face Hallucination

❖ Local Enhancement Network

- up-sample the image I_{lr} to the same size as high-resolution image I_{hr} with Bicubic method.
- generates a residual map



DRL for Face Hallucination

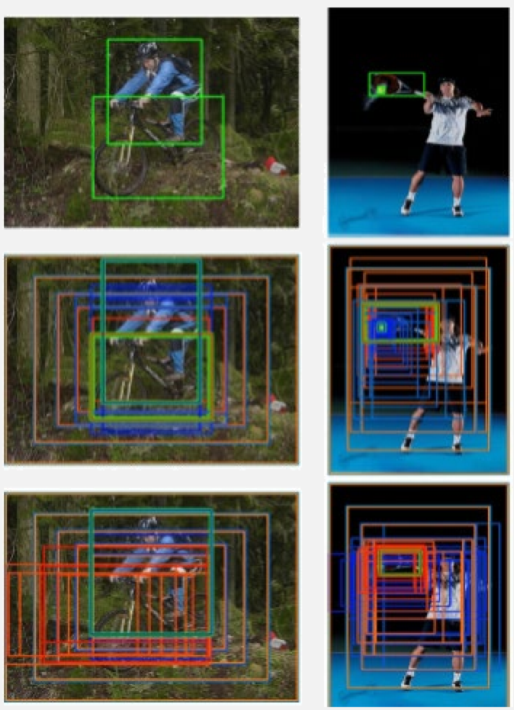


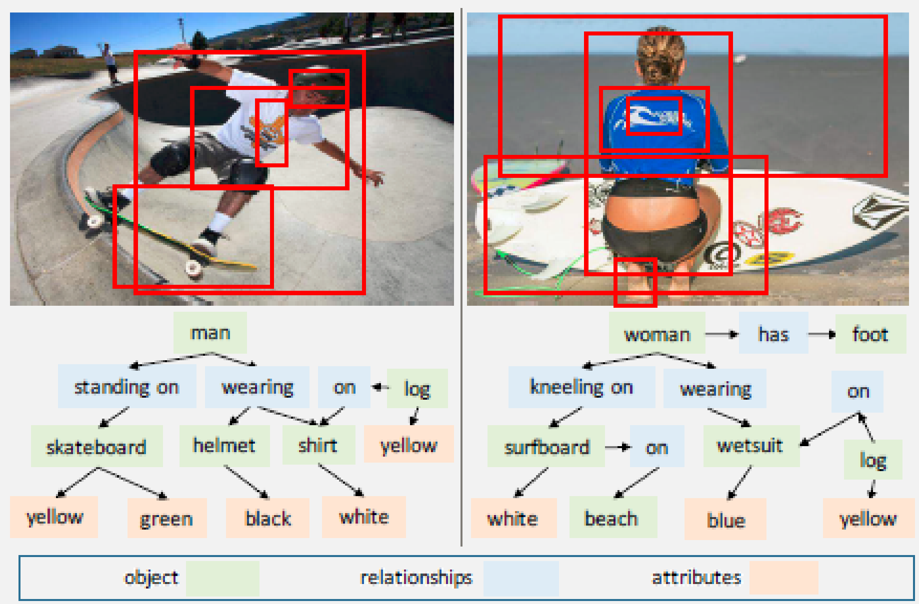
清華大學 Tsinghua University

i-VisionGroup

DRL for Image Understanding

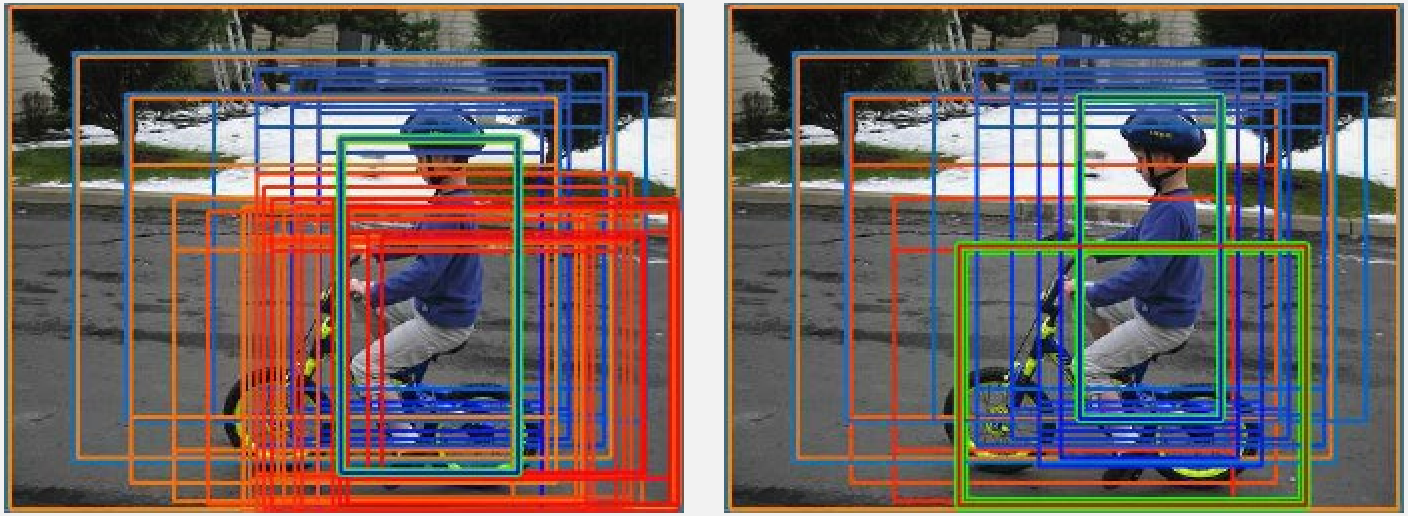
- DRL for Joint Object Search
- DRL for Global Optimized Object Detection
- DRL for Visual Relationship Detection





DRL for Joint Object Search

Collaborative deep reinforcement learning for joint object search



(a) Single agent detection

200 iterations

(b) Joint agent detection

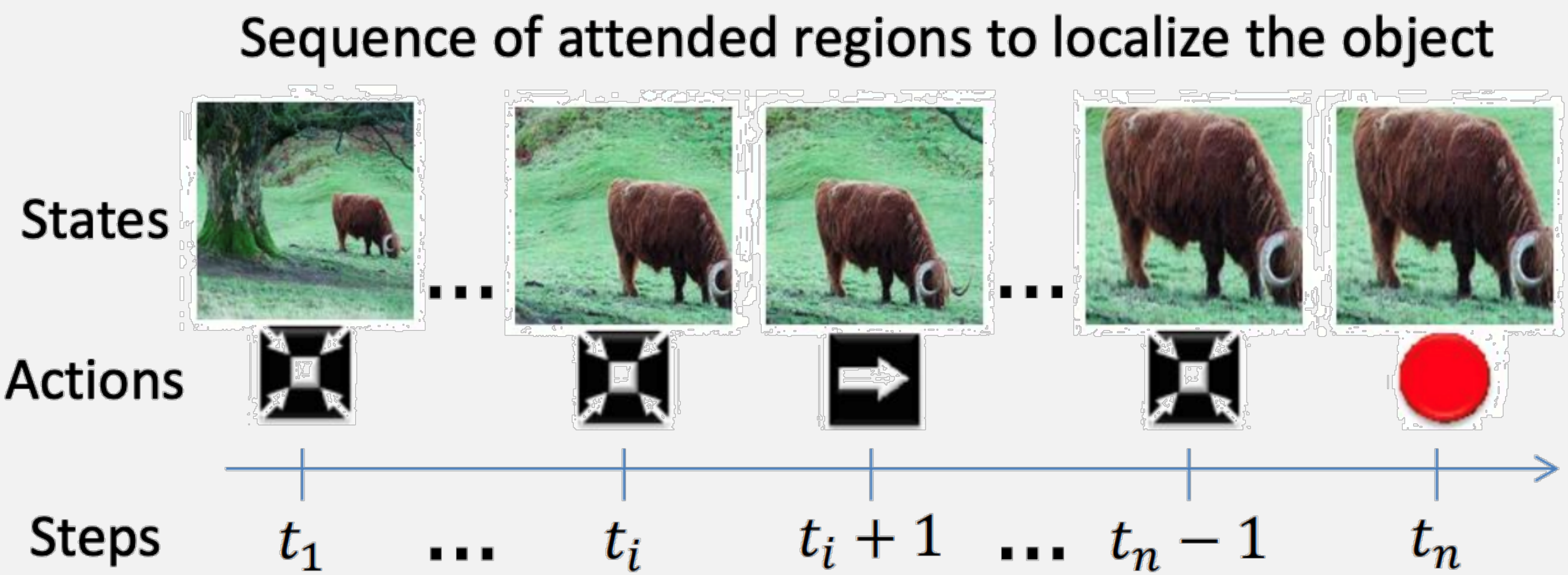
15 iterations

Kong, Xiangyu, Bo Xin, Yizhou Wang, and Gang Hua. "Collaborative deep reinforcement learning for joint object search." CVPR. 2017.

DRL for Joint Object Search

- Single Agent RL Object Localization:

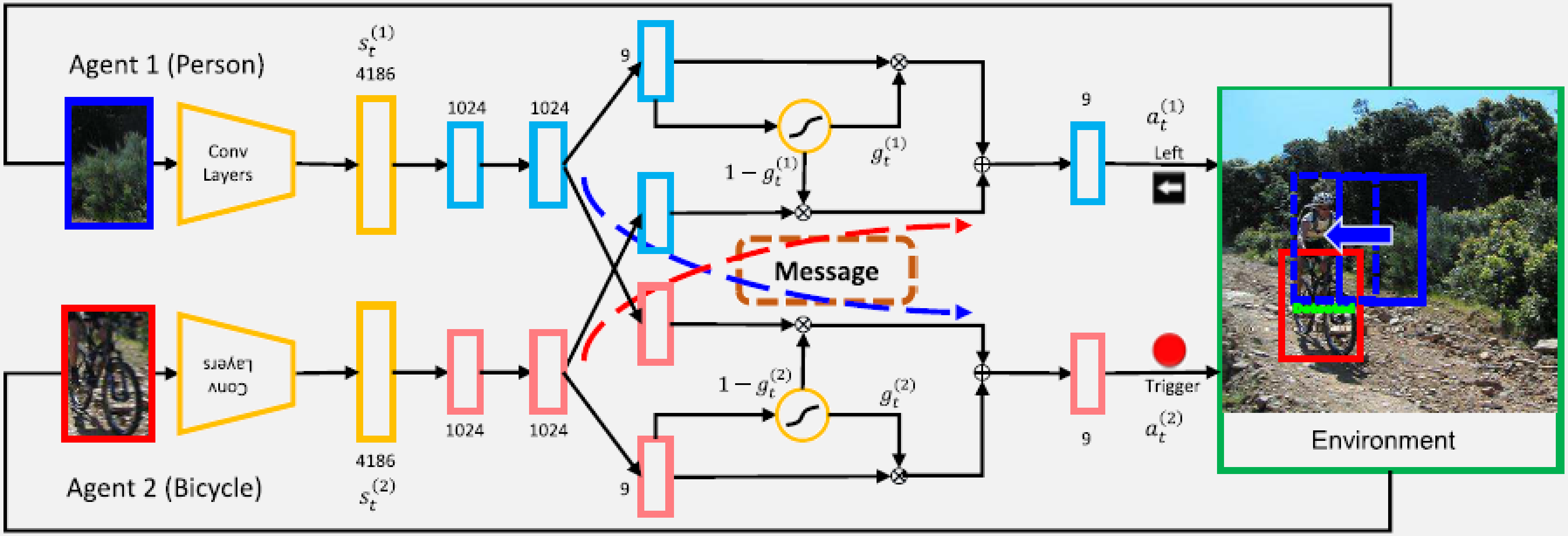
$$R(a, s \to s') = sign(IoU(b', g) - IoU(b, g))$$



DRL for Joint Object Search

- Collaborative RL for Joint Object Localization:

$$Q := Q^{(i)}(a^{(i)}, m^{(i)}, s^{(i)}, m^{(-i)}, \theta_a^{(i)}, \theta_m^{(i)})$$



DRL for Joint Object Search

GT

Joint Agent

Single Agent

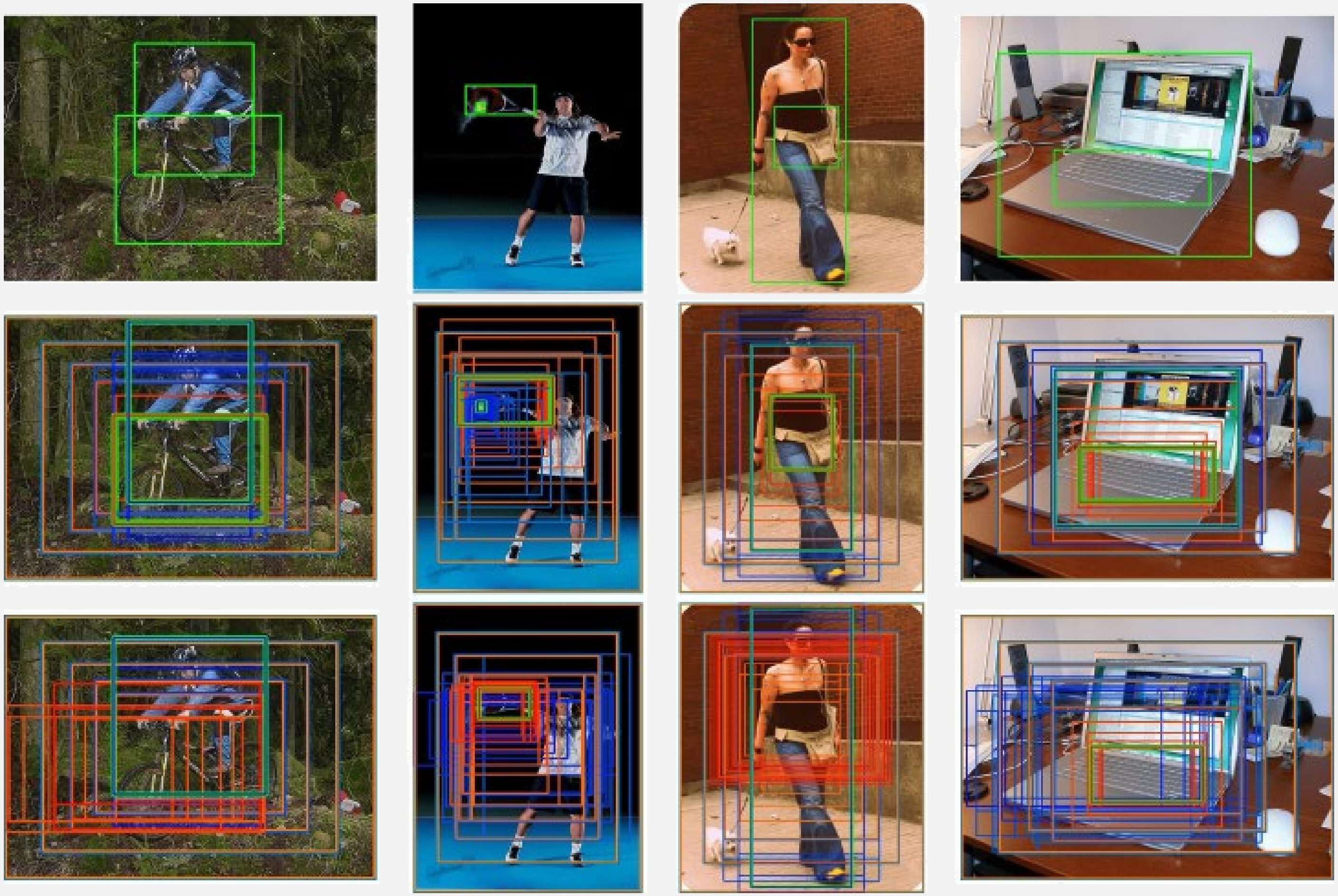
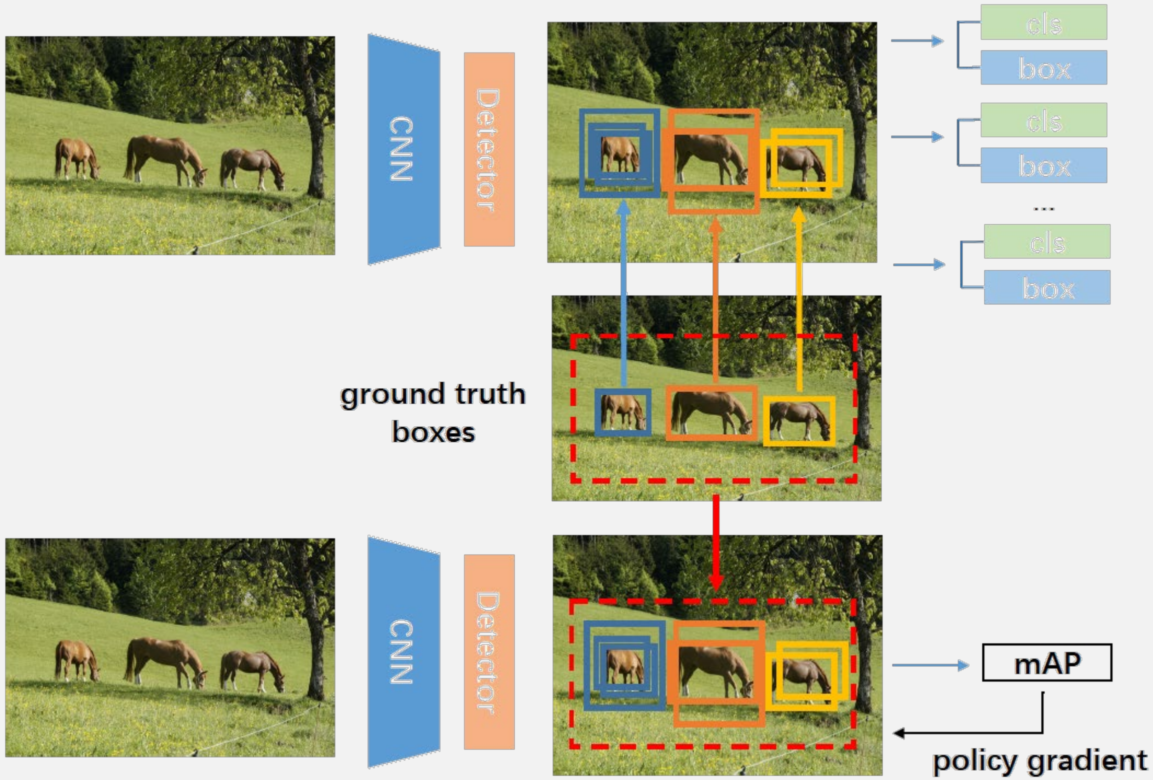


Figure 4. Joint agent detection (mid) compared with single agent detection (bottom). The bounding box trajectories are indicated by gradual color change with blue and red each for one detector. Successful detections are highlighted in bold green.

DRL for Object Detection

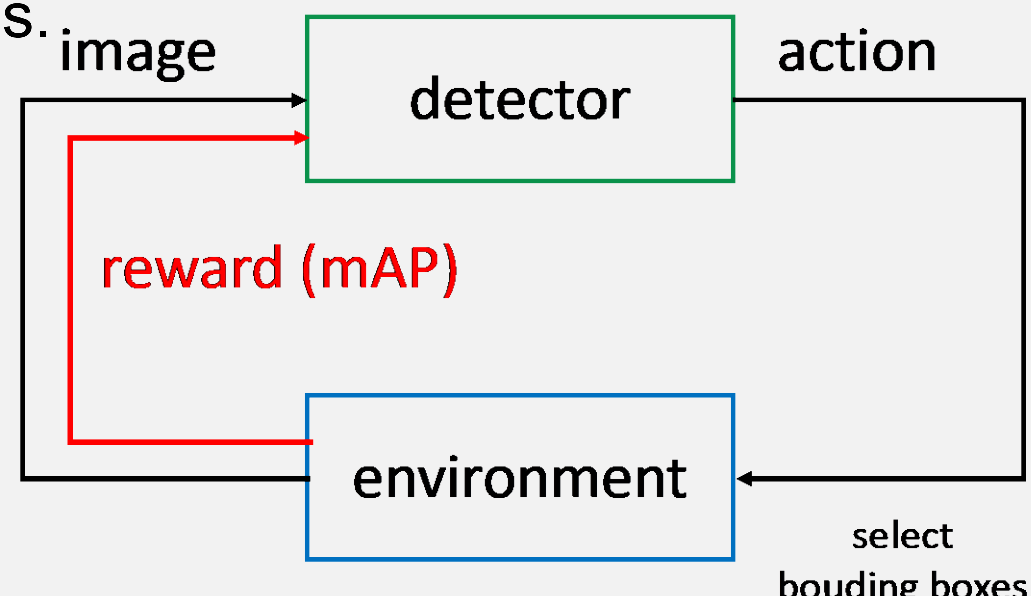
Learning Globally Optimized Object Detector via Policy Gradient



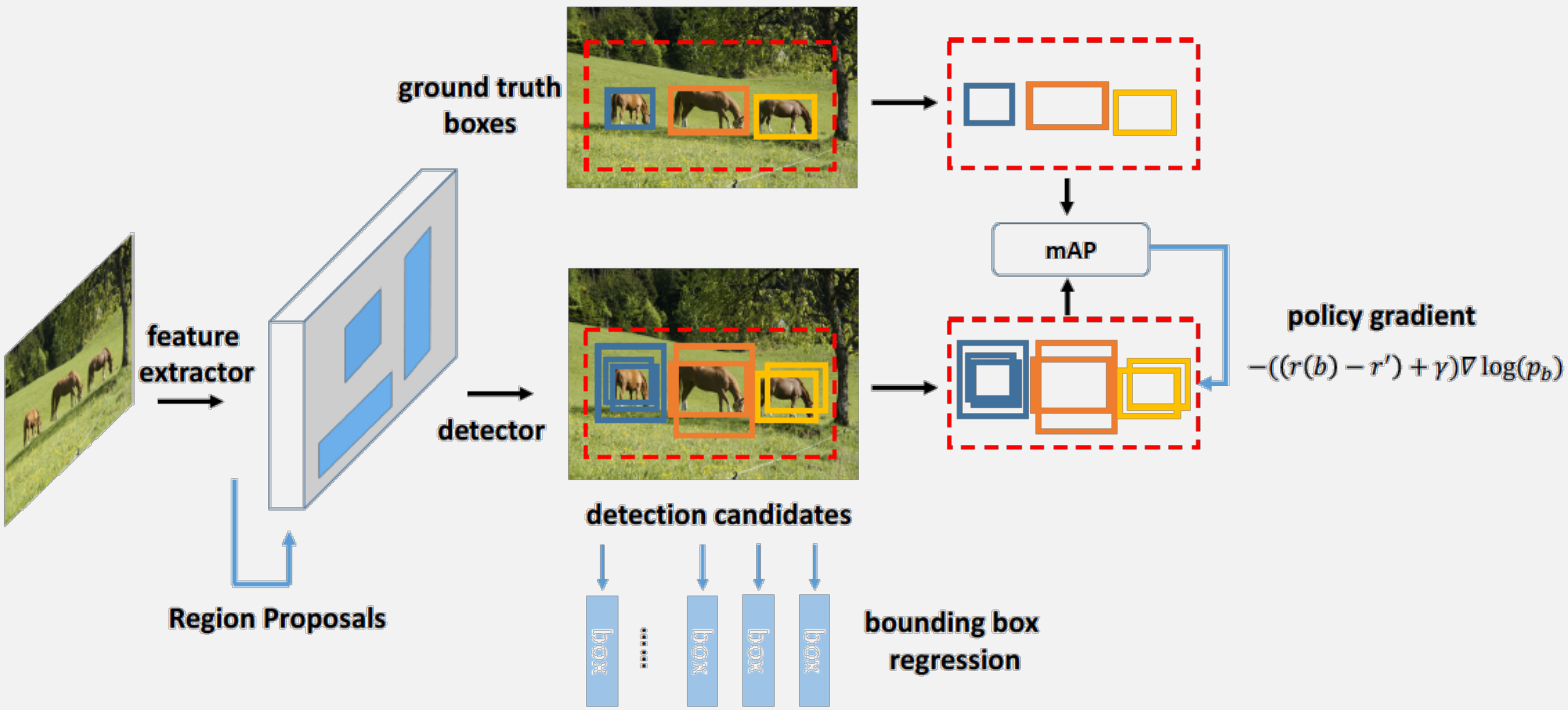
Yongming Rao, Dahua Lin, **Jiwen Lu**, and Jie Zhou. "Learning globally optimized object detector via policy gradient." CVPR. 2018.

DRL for Object Detection

- **Agent:** detector interacts with an external environment
- **Reward:** mAP between detection candidates and ground truth boxes
- The aim of the agent is to get maximum possible mAP and learn a good policy to select bounding boxes from candidates.



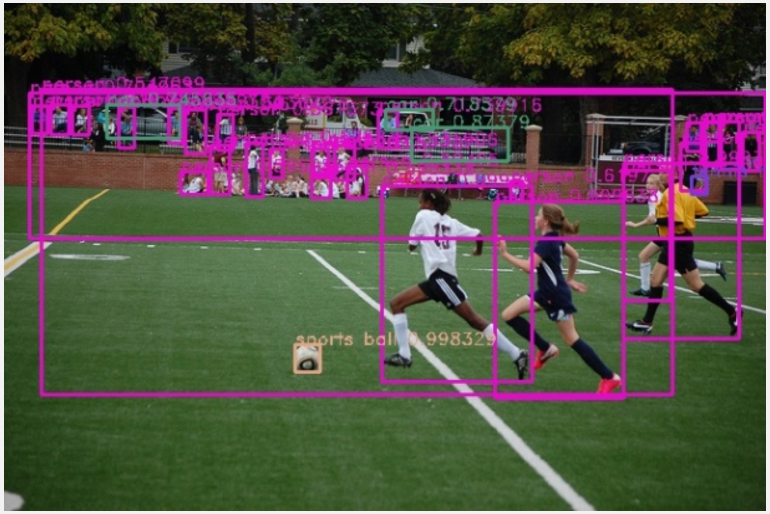
DRL for Object Detection

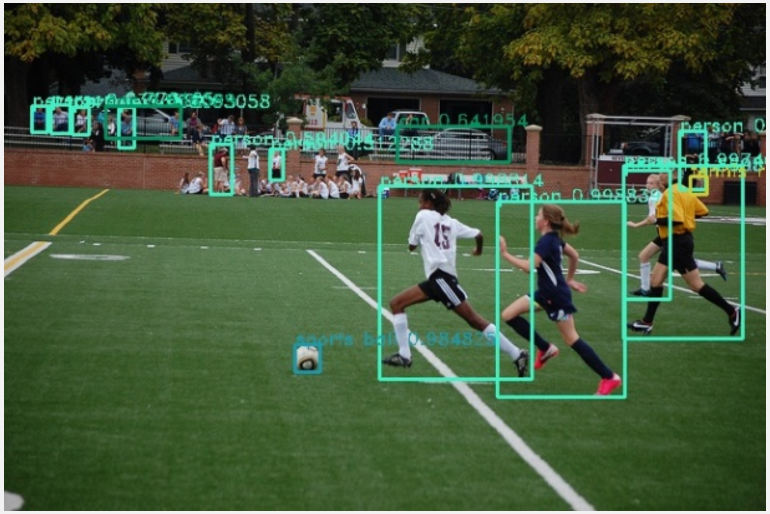


DRL for Object Detection

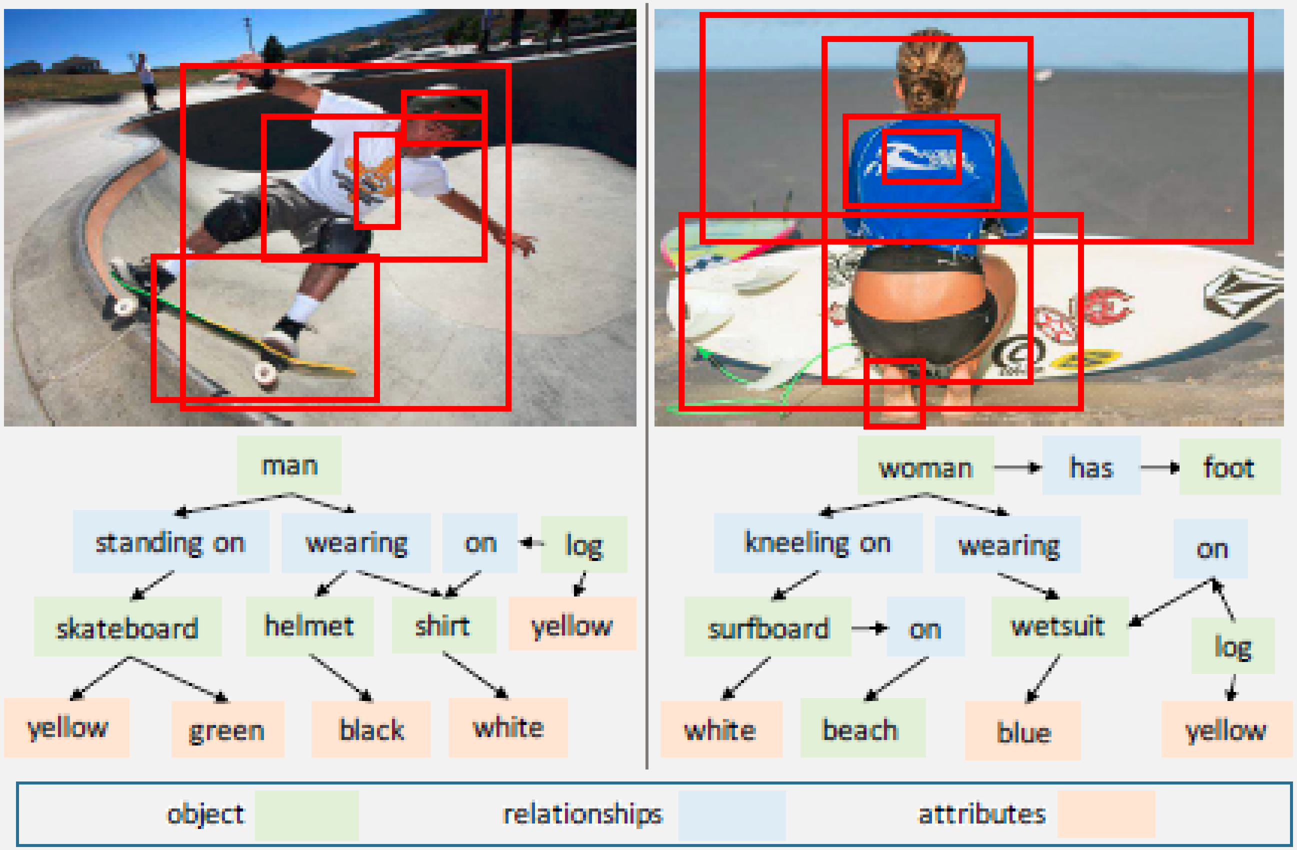
Results on COCO:

Detection model	training method	greedy NMS	soft NMS	mAP	mAP_{50}	mAP_{75}	mAP_S	mAP_M	mAP_L
Faster R-CNN	standard	✓		36.3	57.3	38.8	17.7	42.4	51.4
Faster R-CNN	standard		✓	36.9	57.2	40.1	18.0	42.7	52.1
Faster R-CNN	OHEM	✓		36.9	57.3	40.2	17.7	42.7	52.4
Faster R-CNN	ours ($\gamma = 0$)	✓		37.6	60.0	40.2	19.6	42.6	52.0
Faster R-CNN	ours ($\gamma = 1$)	✓		38.3	60.6	40.9	20.7	43.2	52.6
Faster R-CNN	ours ($\gamma = 1$)		✓	**38.5**	**60.8**	**41.3**	**20.9**	**43.4**	**52.7**
Faster R-CNN with FPN	standard	✓		37.7	58.5	40.8	19.3	41.7	**52.3**
Faster R-CNN with FPN	ours ($\gamma = 1$)	✓		**39.5**	**60.2**	**43.3**	**22.7**	**44.1**	51.9





DRL for Visual Relationship Detection

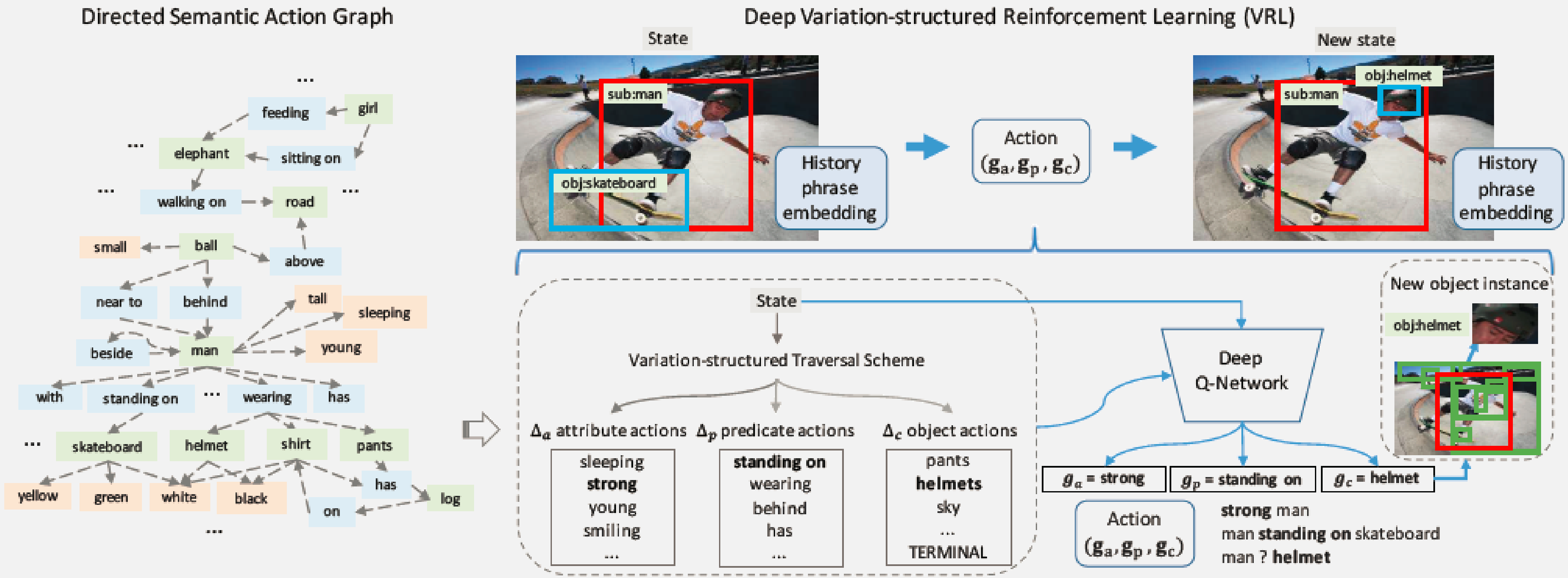


Liang, Xiaodan, Lisa Lee, and Eric P. Xing. "Deep variation-structured reinforcement learning for visual relationship and attribute detection." CVPR, 2017.

DRL for Visual Relationship Detection

- Directed Semantic Action Graph

G = (V, E) is a directed semantic graph to organize all possible object nouns, attributes, and relationships into a compact and semantically meaningful representation.

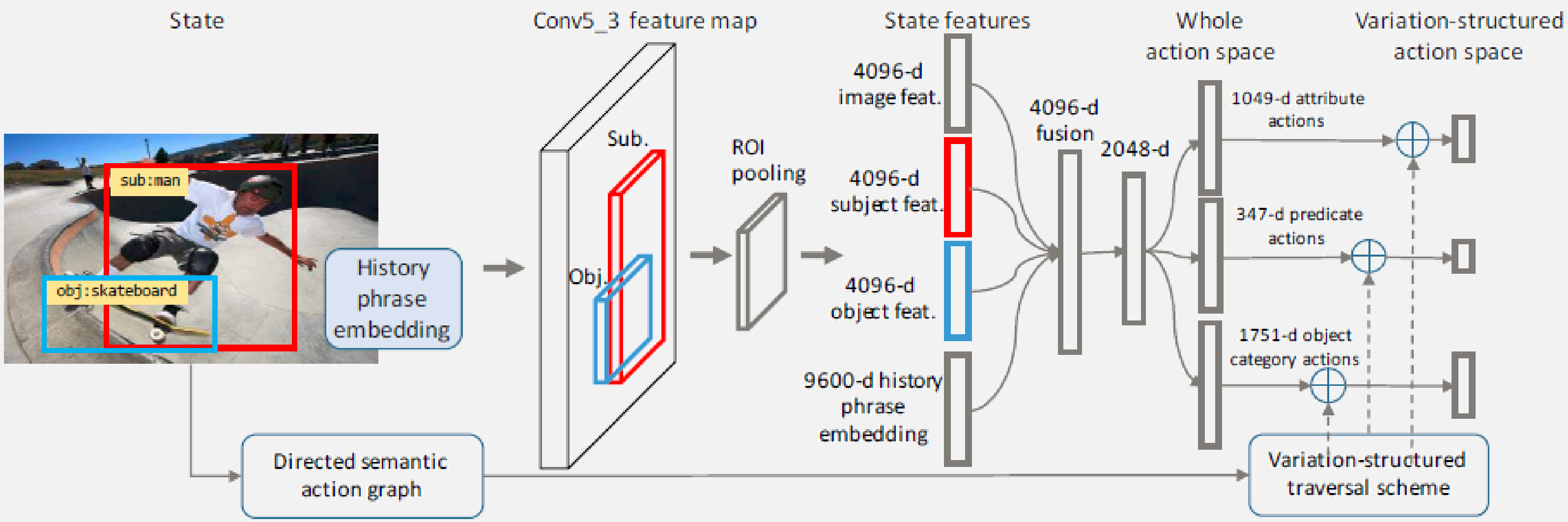


DRL for Visual Relationship Detection

Variation structured RL

- Rewards:

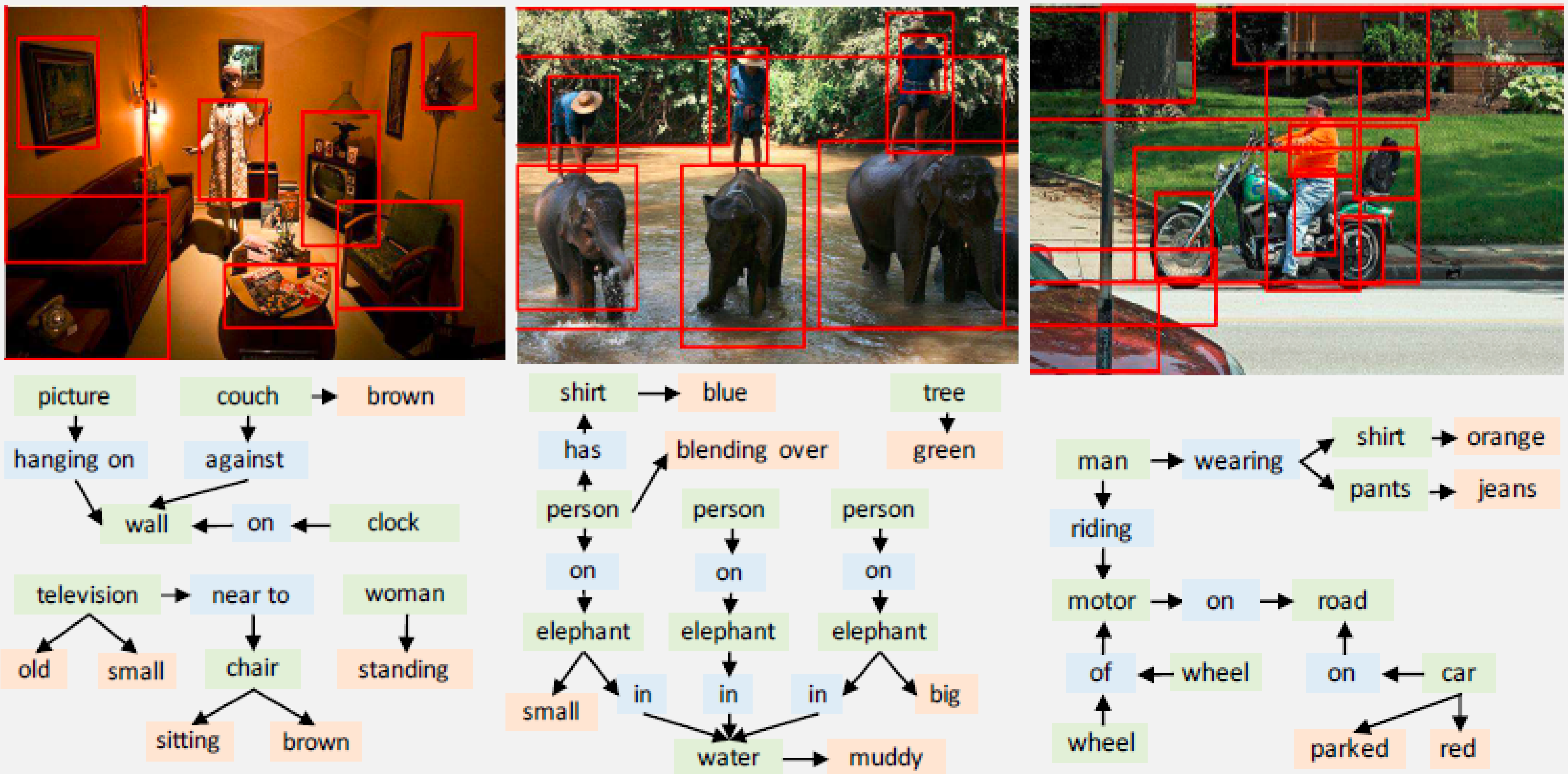
$$\begin{cases} R_a(f, g_a) = \pm 1, \\ R_p(f, g_p) = \pm 1 \\ R_c(f, g_c) = +5/-1 \end{cases}$$



DRL for Visual Relationship Detection

- $L_{\theta_a} = \left(R_a + \gamma \max_{g'_a} Q\left(f', g'_a; \theta_a^{(t)-}\right) - Q\left(f, g_a; \theta_a^{(t)}\right)\right)^2$
- $L_{\theta_p} = \left(R_p + \gamma \max_{g'_p} Q\left(f', g'_p; \theta_p^{(t)-}\right) - Q\left(f, g_p; \theta_p^{(t)}\right)\right)^2$
- $L_{\theta_a} = \left(R_c + \gamma \max_{g'_c} Q\left(f', g'_c; \theta_c^{(t)-}\right) - Q\left(f, g_c; \theta_c^{(t)}\right)\right)^2$

DRL for Visual Relationship Detection



Part 5: Conclusion and Future Directions

Summary

- Deep reinforcement learning has been developed as one of the basic techniques in machine learning and successfully applied to a wide range of computer vision tasks (showing state-of-the-art performance).
- We overview the trend of deep reinforcement learning techniques and discuss how they are employed to boost the performance of various computer vision tasks (solve various problems in computer vision).
- We briefly introduce the basic concept of deep reinforcement learning and show the key challenges in different computer vision tasks.
- We present several applications of deep reinforcement learning in different fields of computer vision.

Future Directions

☐ Inverse-RL:

- To learn from experts without designed rewards

☐ Multi-agent:

- Interaction and communication
- Competition and cooperation

☐ Robotic vision

- Visual grasping
- Visual navigation

References

- Caicedo, Juan C., and Svetlana Lazebnik. "Active object localization with deep reinforcement learning." Proceedings of the IEEE International Conference on Computer Vision. 2015.
- **Jiwen Lu**, Junlin Hu, and Jie Zhou, Deep Metric Learning for Visual Understanding, *IEEE Signal Processing Magazine*, vol. 34, no. 6, pp. 76-84, 2017.
- **Jiwen Lu**, Venice Erin Liong, and Jie Zhou, Simultaneous Local Binary Feature Learning and Encoding for Homogeneous and Heterogeneous Face Recognition, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017, accepted.
- Hao Liu, **Jiwen Lu**, Jianjiang Feng, and Jie Zhou, Two-Stream Transformer Networks for Video-based Face Alignment, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017, accepted.
- Yueqi Duan**, Jiwen Lu**, Jianjiang Feng, and Jie Zhou, Context-Aware Local Binary Feature Learning for Face Recognition, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017, accepted.
- Junlin Hu, **Jiwen Lu**, and Yap-Peng Tan, Sharable and Individual Multi-View Metric Learning, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017, accepted.
- **Jiwen Lu**, Venice Erin Liong, Xiuzhuang Zhou, and Jie Zhou, Learning Compact Binary Face Descriptor for Face Recognition, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 10, pp. 2041-2056, 2015.
- **Jiwen Lu**, Xiuzhuang Zhou, Yap-Peng Tan, Yuanyuan Shang, and Jie Zhou, Neighborhood Repulsed Metric Learning for Kinship Verification, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 2, pp. 331-345, 2014.

References

- **Jiwen Lu**, Yap-Peng Tan, and Gang Wang, Discriminative Multimanifold Analysis for Face Recognition from a Single Training Sample per Person, *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 35, no. 1, pp. 39-51, 2013.
- Supancic III, James Steven, and Deva Ramanan. "Tracking as Online Decision-Making: Learning a Policy from Streaming Videos with Reinforcement Learning." ICCV. 2017.
- Das, Abhishek, et al. "Learning Cooperative Visual Dialog Agents with Deep Reinforcement Learning." Computer Vision (ICCV), 2017 IEEE International Conference on. IEEE, 2017.
- Yongming Rao, **Jiwen Lu**, and Jie Zhou. "Attention-aware deep reinforcement learning for video face recognition." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017.
- Liu, Fangyu, Shuaipeng Li, Liqiang Zhang, Chenghu Zhou, Rongtian Ye, Yuebin Wang, and **Jiwen Lu**. "3DCNN-DQN-RNN: a deep reinforcement learning framework for semantic parsing of large-scale 3D point clouds." In IEEE Int. Conf. on Computer Vision (ICCV), pp. 5679-5688. 2017. Ren, Zhou, et al. "Deep Reinforcement Learning-Based Image Captioning with Embedding Reward." *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017.
- Cao, Qingxing, Liang Lin, Yukai Shi, Xiaodan Liang, and Guanbin Li. "Attention-Aware Face Hallucination via Deep Reinforcement Learning." In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pp. 1656-1664. IEEE, 2017
- Liang, Xiaodan, Lisa Lee, and Eric P. Xing. "Deep variation-structured reinforcement learning for visual relationship and attribute detection." *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*. IEEE, 2017.

References

- Yun, S., Choi, J., Yoo, Y., Yun, K., & Choi, J. Y. (2017, July). Action-Decision Networks for Visual Tracking with Deep Reinforcement Learning. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on* (pp. 1349-1358). IEEE.
- Wang, Xin, Wenhu Chen, Jiawei Wu, Yuan-Fang Wang, and William Yang Wang. "Video captioning via hierarchical reinforcement learning." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4213-4222. 2018.
- Tang, Yansong, Yi Tian, **Jiwen Lu**, Peiyang Li, and Jie Zhou. "Deep Progressive Reinforcement Learning for Skeleton-Based Action Recognition." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5323-5332. 2018.
- Lan, Shuyue, Rameswar Panda, Qi Zhu, and Amit K. Roy-Chowdhury. "FFNet: Video Fast-Forwarding via Reinforcement Learning." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6771-6780. 2018.
- Pirinen, Aleksis, and Cristian Sminchisescu. "Deep Reinforcement Learning of Region Proposal Networks for Object Detection." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6945-6954. 2018.
- Li, Debang, Huikai Wu, Junge Zhang, and Kaiqi Huang. "A2-RL: Aesthetics Aware Reinforcement Learning for Image Cropping." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8193-8201. 2018.
- Duan, Yueqi, Ziwei Wang, **Jiwen Lu**, Xudong Lin, and Jie Zhou. "GraphBit: Bitwise Interaction Mining via Deep Reinforcement Learning." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8270-8279. 2018.

References

- Kong, Xiangyu, Bo Xin, Yizhou Wang, and Gang Hua. "Collaborative deep reinforcement learning for joint object search." In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017.
- Han, Junwei, Le Yang, Dingwen Zhang, Xiaojun Chang, and Xiaodan Liang. "Reinforcement Cutting-Agent Learning for Video Object Segmentation." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9080-9089. 2018.
- Yongming Rao, Dahua Lin, **Jiwen Lu**, and Jie Zhou. "Learning Globally Optimized Object Detector via Policy Gradient." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6190-6198. 2018.
- Chen, Lei, **Jiwen Lu**, Zhanjie Song, and Jie Zhou. "Part-Activated Deep Reinforcement Learning for Action Prediction." In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 421-436. 2018.
- Guo, Minghao, **Jiwen Lu**, and Jie Zhou. "Dual-Agent Deep Reinforcement Learning for Deformable Face Tracking." In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 768-783. 2018.
- **Liangliang Ren**, Xin Yuan, **Jiwen Lu**, Ming Yang, and Jie Zhou. "Deep Reinforcement Learning with Iterative Shift for Visual Tracking." In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 684-700. 2018.
- **Liangliang Ren**, **Jiwen Lu**, Zifeng Wang, Qi Tian, and Jie Zhou. "Collaborative Deep Reinforcement Learning for Multi-object Tracking." In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 586-602. 2018.

References

- Liang, Xiaodan, Tairui Wang, Luona Yang, and Eric Xing. "CIRL: Controllable imitative reinforcement learning for vision-based self-driving." *arXiv preprint arXiv:1807.03776* 1 (2018).
- Ramakrishnan, Santhosh K., and Kristen Grauman. "Sidekick Policy Learning for Active Visual Exploration." In *European Conference on Computer Vision*, pp. 424-442. Springer, Cham, 2018.
- Yuan, Xin, **Liangliang Ren**, **Jiwen Lu**, and Jie Zhou. "Relaxation-Free Deep Hashing via Policy Gradient." In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 134-150. 2018.
- Rhinehart, Nicholas, Kris M. Kitani, and Paul Vernaza. "R2P2: A reparameterized pushforward policy for diverse, precise generative path forecasting." In *The European Conference on Computer Vision (ECCV)*. 2018.
- Chen, Boyu, Dong Wang, Peixia Li, Shuang Wang, and Huchuan Lu. "Real-Time 'Actor-Critic'Tracking." In *European Conference on Computer Vision*, pp. 328-345. Springer, Cham, 2018.
- Rhinehart, Nicholas, and Kris Kitani. "First-Person Activity Forecasting from Video with Online Inverse Reinforcement Learning." IEEE transactions on pattern analysis and machine intelligence (2018).
- Liu, Feng, et al. "Inverse Visual Question Answering: A New Benchmark and VQA Diagnosis Tool." IEEE transactions on pattern analysis and machine intelligence (2018).
- Zhang, Xiaowei, et al. "Too Far to See? Not Really!—Pedestrian Detection With Scale-Aware Localization Policy." IEEE transactions on image processing 27.8 (2018): 3703-3715.
- **Yongming Rao**, **Jiwen Lu**, Ji Lin, and Jie Zhou. "Runtime Network Routing for Efficient Image Classification." *IEEE transactions on pattern analysis and machine intelligence*(2018).
- Dong, Xingping, Jianbing Shen, Wenguan Wang, Yu Liu, Ling Shao, and Fatih Porikli. "Hyperparameter Optimization for Tracking With Continuous Deep Q-Learning." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 518-527. 2018.