



Imperial College

MICCAI2018
Granada
SPAIN

Automatic View Planning with Multi-scale Deep Reinforcement Learning Agents

Amir Alansary

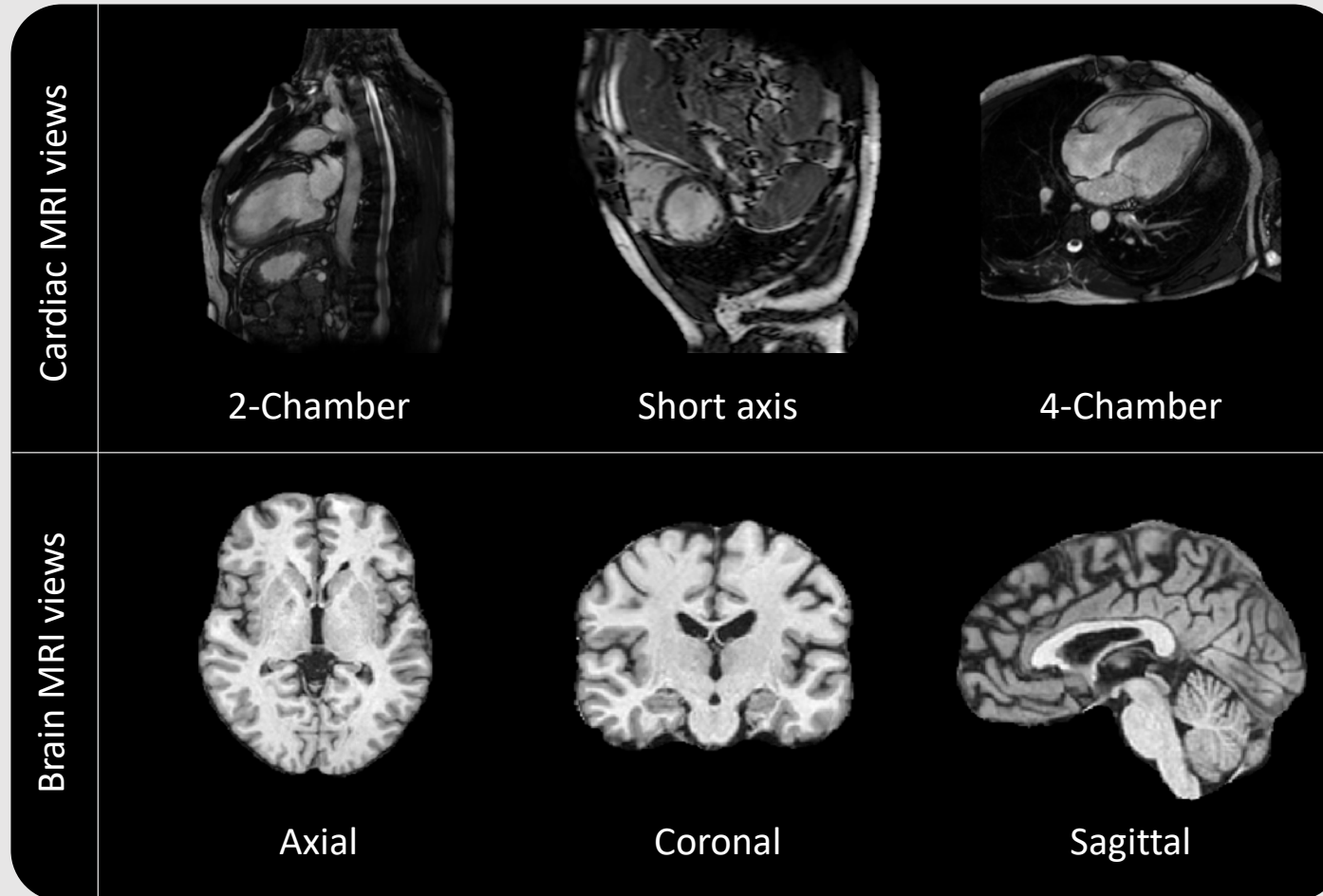
Imperial College London, UK
a.alansary14@imperial.ac.uk

View Planning - Motivation



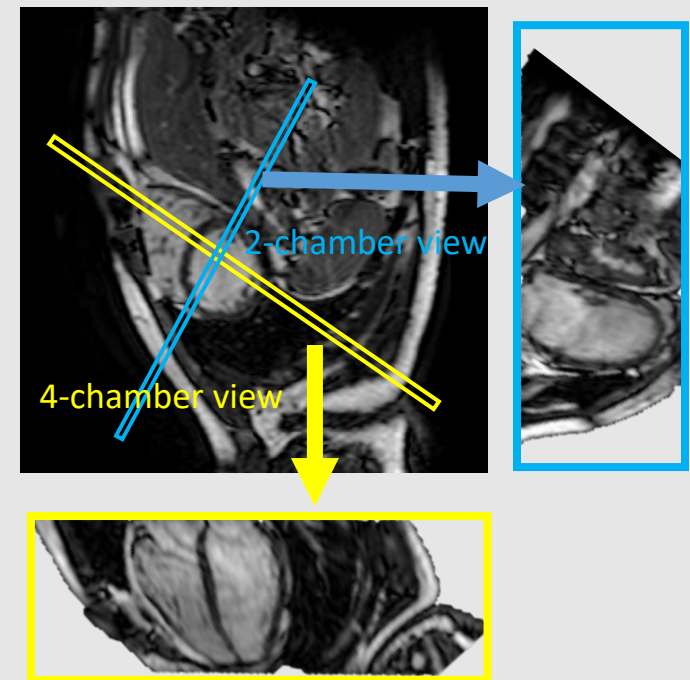
- Standard view planning through a defined anatomy is commonly used in clinical practice to establish comparable metrics
- Obtaining accurate biometric measurements that are comparable across populations is essential for diagnosis and supporting critical decision making
- Standardized views are used to initialize image registration methods, or to evaluate and assess anomalies (*e.g.* mid-sagittal plane in brain and 4-chamber plane in cardiac)

Examples



Cardiac MRI: 4-Chamber View Planning

1. Localize 3 planes: axial, coronal, and sagittal
2. Acquire an axial stack - above the aortic arch to below the level of the heart
3. Define 2-chamber (2CH) view plane:
 - Perpendicular to axial plane
 - Parallel to interventricular septum (IVS)
 - In the middle of left ventricle (LV)
4. Define pseudo short axis (SA) view plane:
 - Perpendicular to the 2CH view
 - Align with mitral valve
5. Define 4-chamber (4CH) view plane:
 - In SA view:
 - Perpendicular to SA
 - Goes through the center LV and the intersection of interior and inferior of the free wall
 - In 2CH view:
 - Divides the heart along the long axis



Challenges

- Appearance of relevant structures can exhibit large variance according to the positioning of the imaging plane
- Finding planes in an imaging examination through a 3D volume is slow and suffers from inter-observer variability

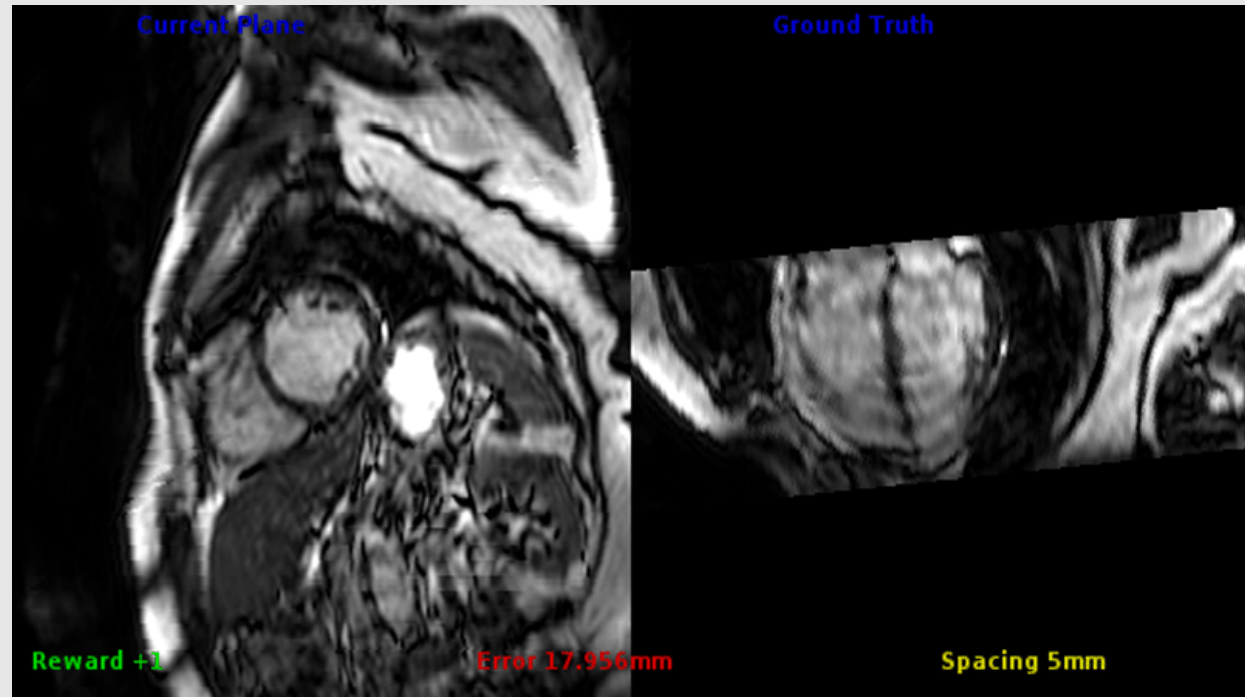
Solution

We propose a novel approach to automate the view planning process by using reinforcement learning (RL), where an agent learns to make comprehensive and sensible decisions by mimicking the view planning process

Reinforcement Learning - Motivation



Mnih et al. 2015



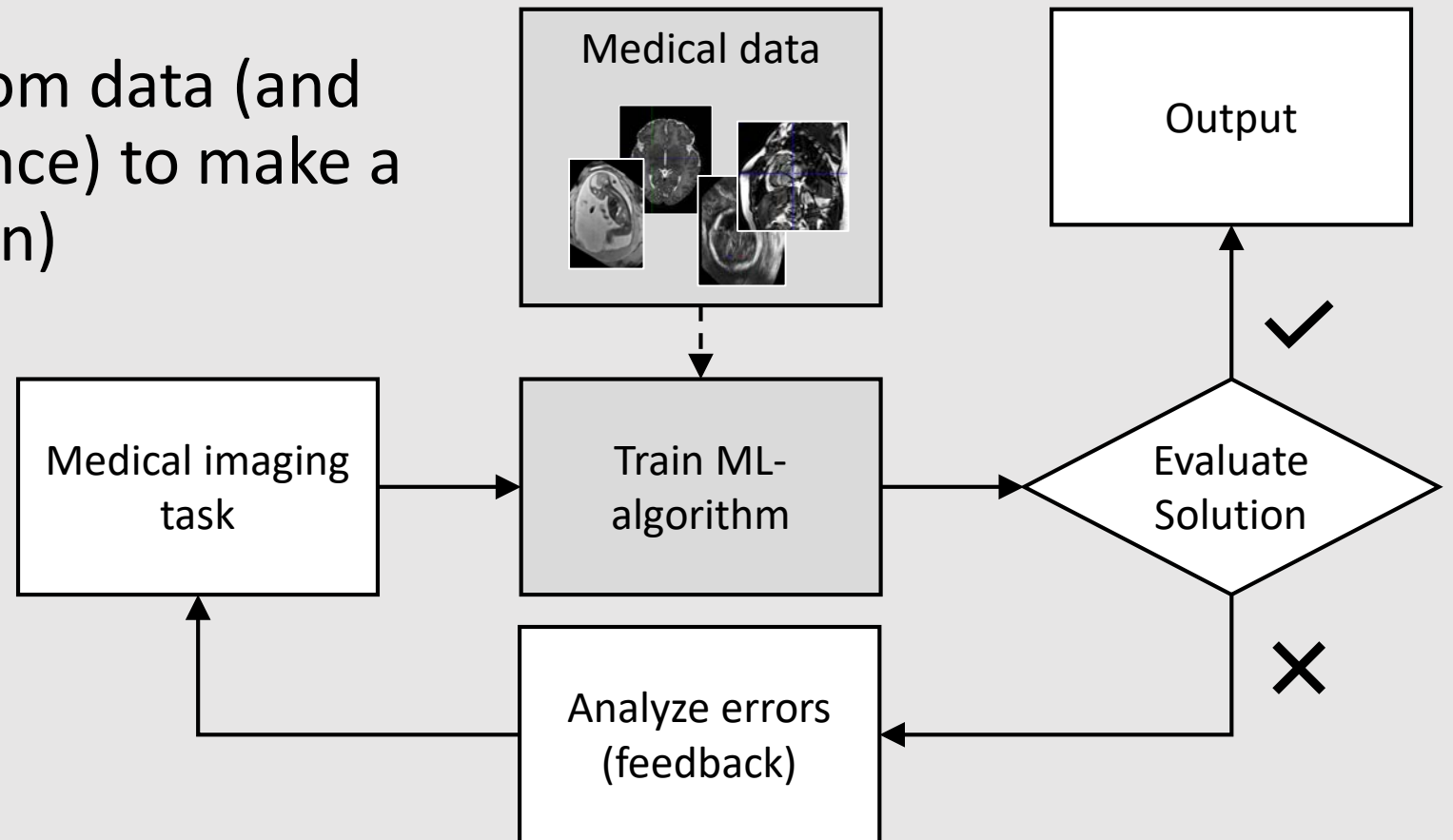
Our agent for view planning

Machine Learning



Automatically learn from data (and improve from experience) to make a decision (take an action)

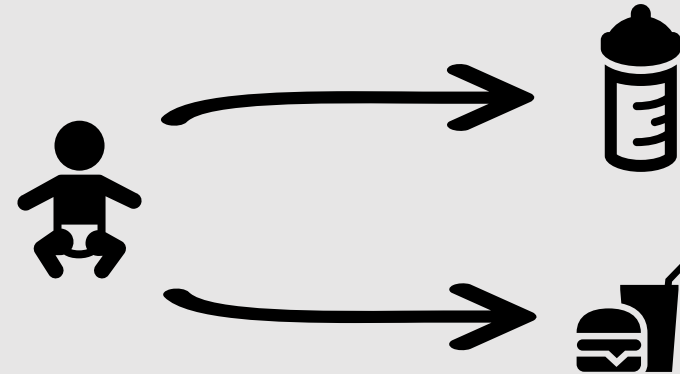
- Unsupervised
- Supervised
- Reinforcement
- ...



Unsupervised Learning



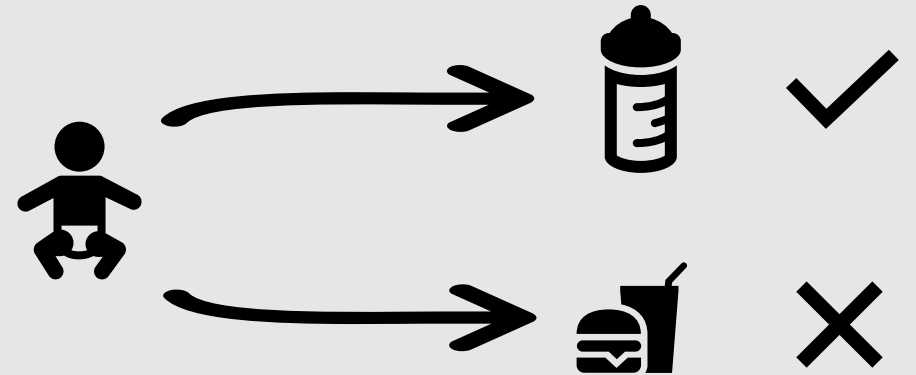
Explores data and draws inferences from datasets to describe hidden structures from unlabeled data



Supervised Learning



Learning from a training set of labeled examples provided by a knowledgeable external supervisor

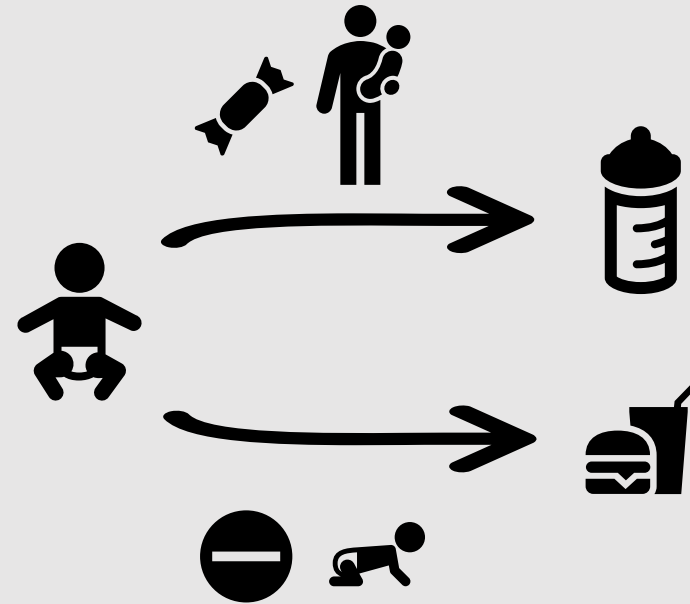


Reinforcement Learning

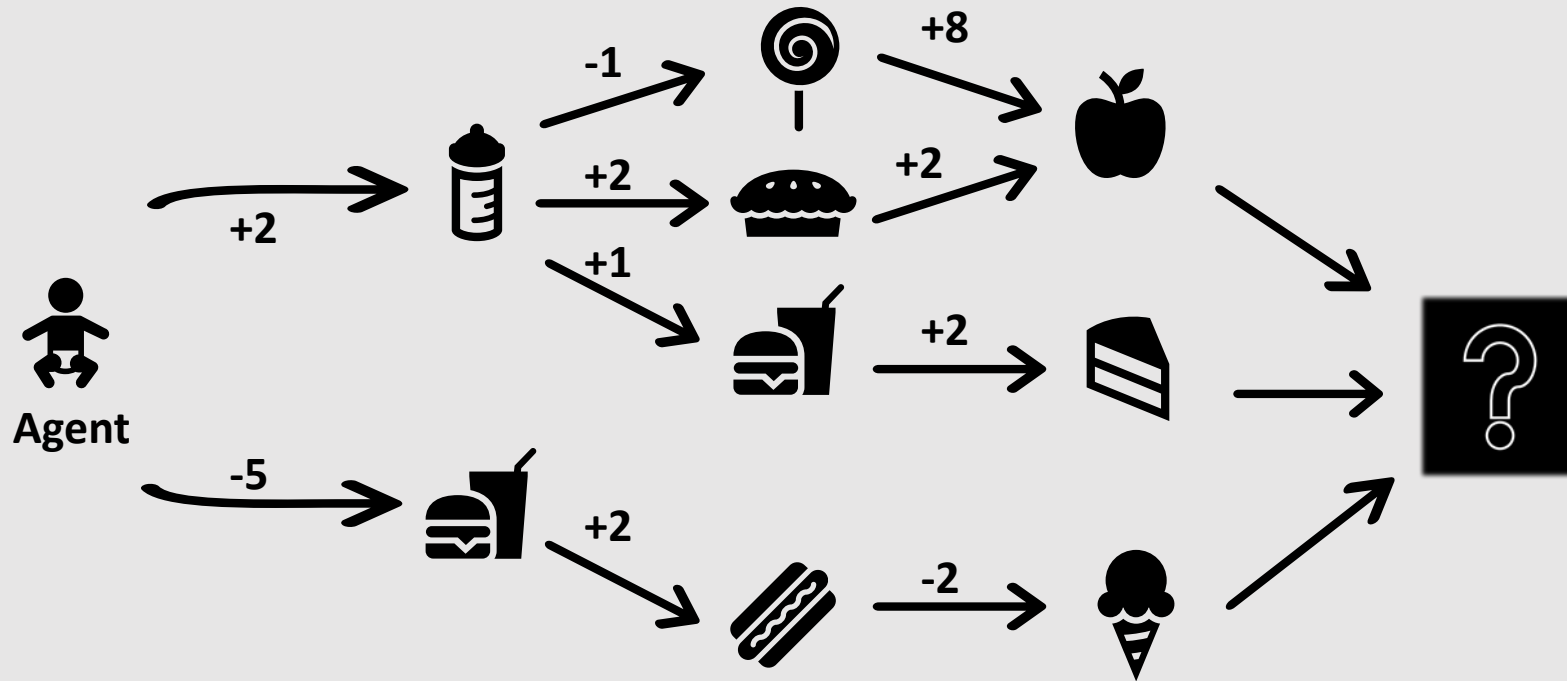


Computational approach to learn by interacting with an environment

- Single decision must be made
 - Multiple actions
 - Each action has a reward associated with it
- Goal is to maximize reward
 - Pick an action with the highest reward



Reinforcement Learning



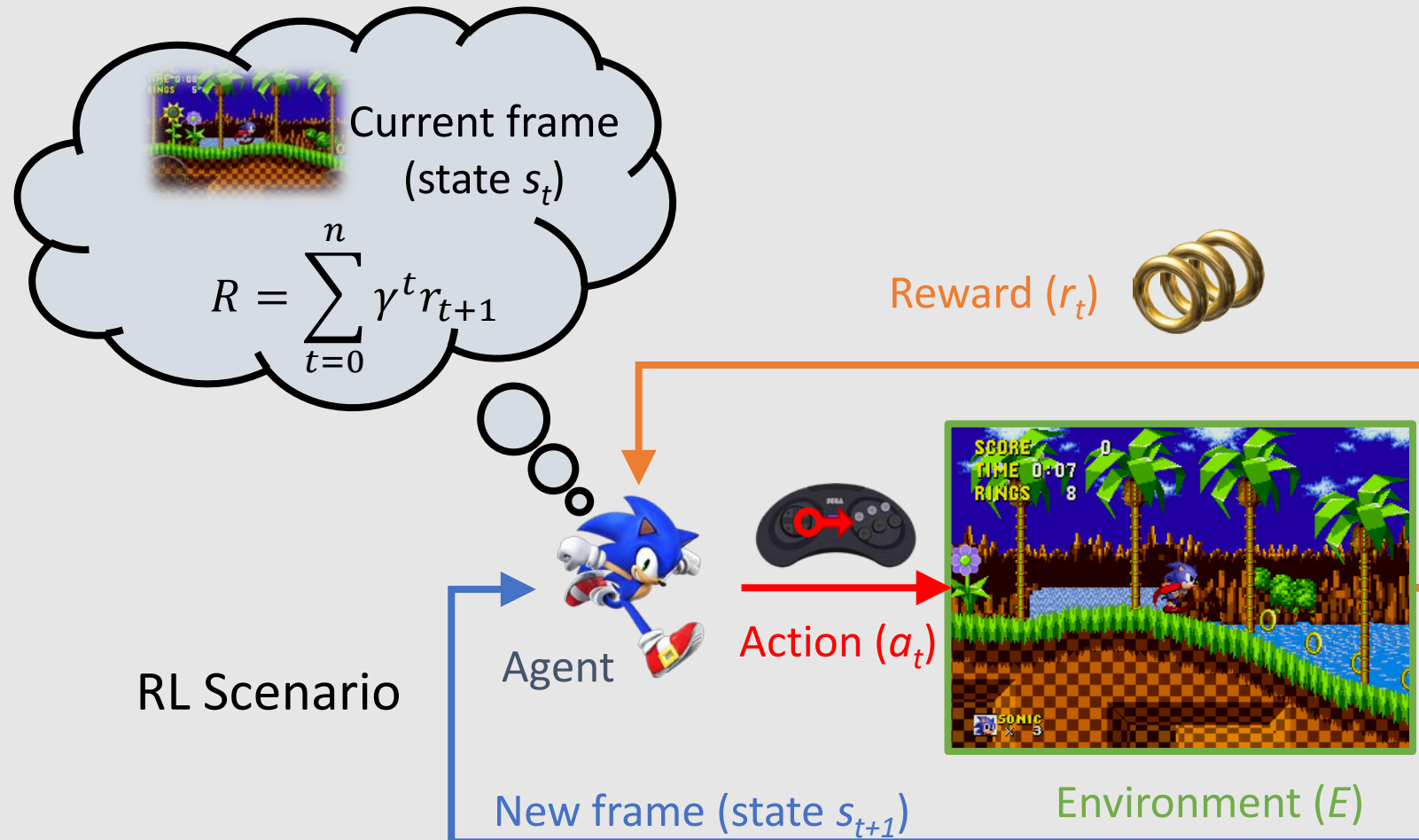
Sequential decision making

Exploitation-Exploration Dilemma

- The goal of the agent is to obtain a lot of reward
 - **Exploitation** - prefer actions that it has tried in the past and found to be effective in producing reward
 - **Exploration** - discover such actions, it has to try actions that it has not selected before
- Simple exploration methods are the most practical:
 - **ϵ -greedy** - the agent chooses an action uniformly at random with probability $(1 - \epsilon)$
 - **ϵ -soft** - similar to ϵ -greedy but the probability is divided by number of actions
 - **Softmax**
 - ...

Note exploration and exploitation dilemma does not arise in supervised and unsupervised learning

Reinforcement Learning



Some RL Terminologies



State

- Whatever information is available to the agent about its environment

Terminal state

- The final state where no more available actions, followed by a reset to a standard starting state or to a sample from a standard distribution of starting states

Episode

- Complete play from the initial to final state $(s_0, a_0, r_0), (s_1, a_1, r_1), \dots, (s_n, a_n, r_n)$

Cumulative Reward

- The discounted sum of rewards accumulated throughout an episode

$$R = \sum_{t=0}^n \gamma^t r_{t+1}$$

RL Main Elements



Policy π

- The agent's strategy to choose an action at each state
- **Optimal Policy π^*** is the theoretical policy that maximizes the expectation of cumulative rewards

Reward signal

- Specifies what's good and what's bad in an immediate sense

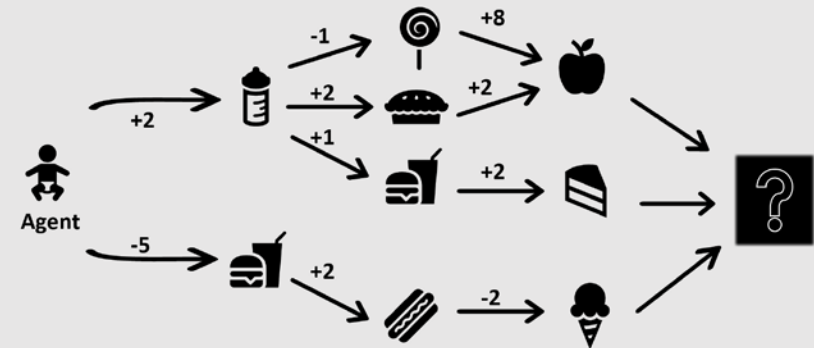
Value function

- The total amount of reward an agent can expect to accumulate over the future

RL Solution



- Approximates iteratively the optimal value function when the whole MDP is unknown by sampling states and actions from the MDP, and learning from experience
 - Certainty equivalence
 - Temporal difference (TD)
 - State-action-reward-state-action (SARSA)
 - Q-learning
 - ...



Reinforcement learning

Learning what to do (how to map situations to action) -> so as to maximize sum of numerical rewards seen over the learner's lifetime (**Policy π : S->A**)

Value Functions



- A value function is defined as a prediction of the expected, accumulated, discounted, future reward in order to measure how good each state or state-action is
- **State-action value function:** Estimates a value of each action a in each state s under policy π

$$Q^{\pi}(s, a) = E[R|s, a, \pi]$$

- Optimal policy $*$ achieves the best expected return from *any* initial state

$$Q^*(s, a) = \max_{\pi} Q^{\pi}(s, a)$$

Deep Q-Networks (DQN) Mnih 2013



- DQN is an implementation of a standard Q-learning algorithm with function approximation using a CNN

$$Q^\pi(s, a) \approx Q^\pi(s, a; \theta)$$

- Objective function: MSE in Q-values

$$L(\theta) = E_{s,r,a,s' \sim D} \left[\left(r + \gamma \max_{a'} Q(s', a'; \theta^-) - Q(s, a; \theta) \right)^2 \right]$$

- Optimize end-to-end by SGD, using $\frac{\delta L(\theta)}{\delta \theta}$

From David Silver lectures on RL

RL in Medical Imaging Analysis



Image Segmentation

- RL for image thresholding and segmentation

Shokri, M. et al. (2003)
Sahba, F. et al. (2006)

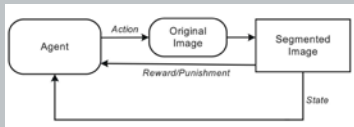
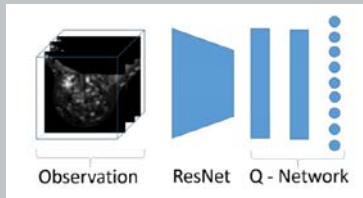


Image Localization

- Deep RL for Active Breast Lesion Detection from DCE-MRI

Maicas, G. et al. (2017)



Landmark Detection

- Artificial agent for anatomical landmark detection in medical images

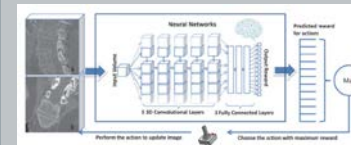
Ghesu, FC. et al. (2016, 2017)
Alansary, A. (2018)



Image Registration

- Artificial Agent for Robust Image Registration (rigid, non-rigid, 2D/3D)

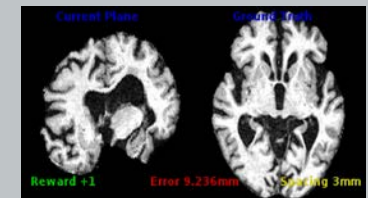
Liao, R. et al. (2017)
Krebs J. et al. (2017)
Miao, S. et al. (2017)



View Planning


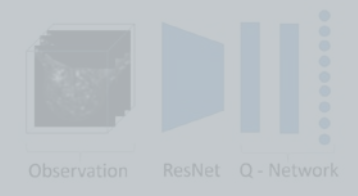


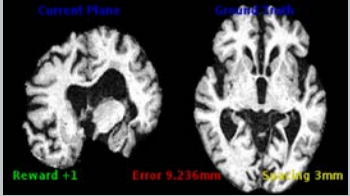
- Automatic view planning using deep RL agents

Alansary, A. (2018)



RL in Medical Imaging Analysis



Image Segmentation	Image Localization	Landmark Detection	Image Registration	View Planning
<ul style="list-style-type: none">• RL for image thresholding and segmentation <p>Shokri, M. et al. (2003) Sahba, F. et al. (2006)</p> 	<ul style="list-style-type: none">• Deep RL for Active Breast Lesion Detection from DCE-MRI <p>Maicas, G. et al. (2017)</p> 	<ul style="list-style-type: none">• Artificial agent for anatomical landmark detection in medical images <p>Ghesu, FC. et al. (2016, 2017) Alansary, A. (2018)</p> 	<ul style="list-style-type: none">• Artificial Agent for Robust Image Registration (rigid, non-rigid, 2D/3D) <p>Liao, R. et al. (2017) Krebs J. et al. (2017) Miao, S. et al. (2017)</p> 	<ul style="list-style-type: none">• Automatic view planning using deep RL agents <p>Alansary, A. (2018)</p> 

RL Agents for View Planning



Sequential decision process, where our RL-agent learns to navigate in an environment by sampling new planes towards the target plane using discrete action-steps

States:

Sampled 3D planes [$\mathbf{a} \cdot \mathbf{x} + \mathbf{b} \cdot \mathbf{y} + \mathbf{c} \cdot \mathbf{z} + \mathbf{d} = 0$] from the input 3D image scan

Action space:

At every step, the agent selects an action to update plane parameters for the next plane

$$\{\pm a_{\theta_x}, \pm a_{\theta_y}, \pm a_{\theta_z}, \pm a_d\}$$

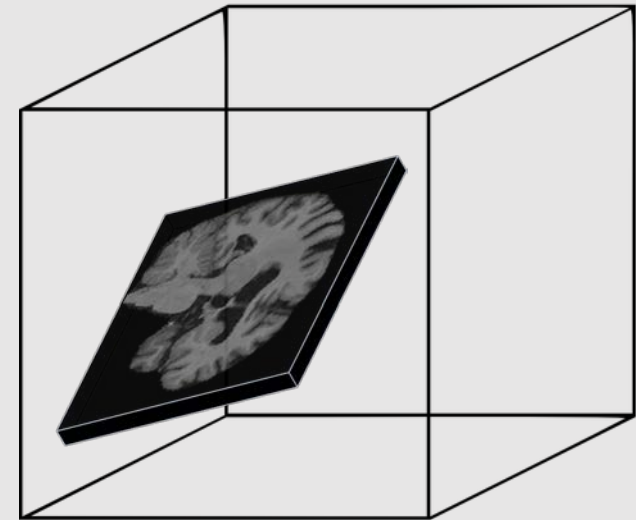


Image Scan

Reward Function (R)



- Designing good empirical reward functions R is often difficult as RL agents can easily under/overfit the specified reward and thereby produce undesirable or unexpected results
- R should be proportional to the improvement that the agent makes to detect the target plane after selecting a particular action

$$R = \text{sgn}(D(P_{i-1}, P_t) - D(P_i, P_t))$$

D Euclidean distance between plane parameters

P_i current plane at step i

P_t target ground truth plane

Terminal State



Training:

- Distance between current estimated and ground truth parameters are less than T_θ

Testing:

1. Extra trigger action that terminates
 - + Modifies the environment by marking target plane
 - Increases the complexity of the task to be learned by increasing the action space size.
 2. Oscillation property ^[1]
 - + No added complexity to the action space
 - The correct target plane is unmarked in the environment
- The terminating state based on the corresponding lower Q-value, when the agent oscillates
 - Q-values are lower when the agent is closer to the target point and higher when it is far
 - Intuitively, it encourages awarding higher Q-values to actions for far states from target



[1] Martin Riedmiller “Reinforcement learning without an explicit terminal state.” Neural Networks Proceedings, 1998.

Multi-scale Agent



Motivation

Capture spatial relations within a global neighborhood

Challenge

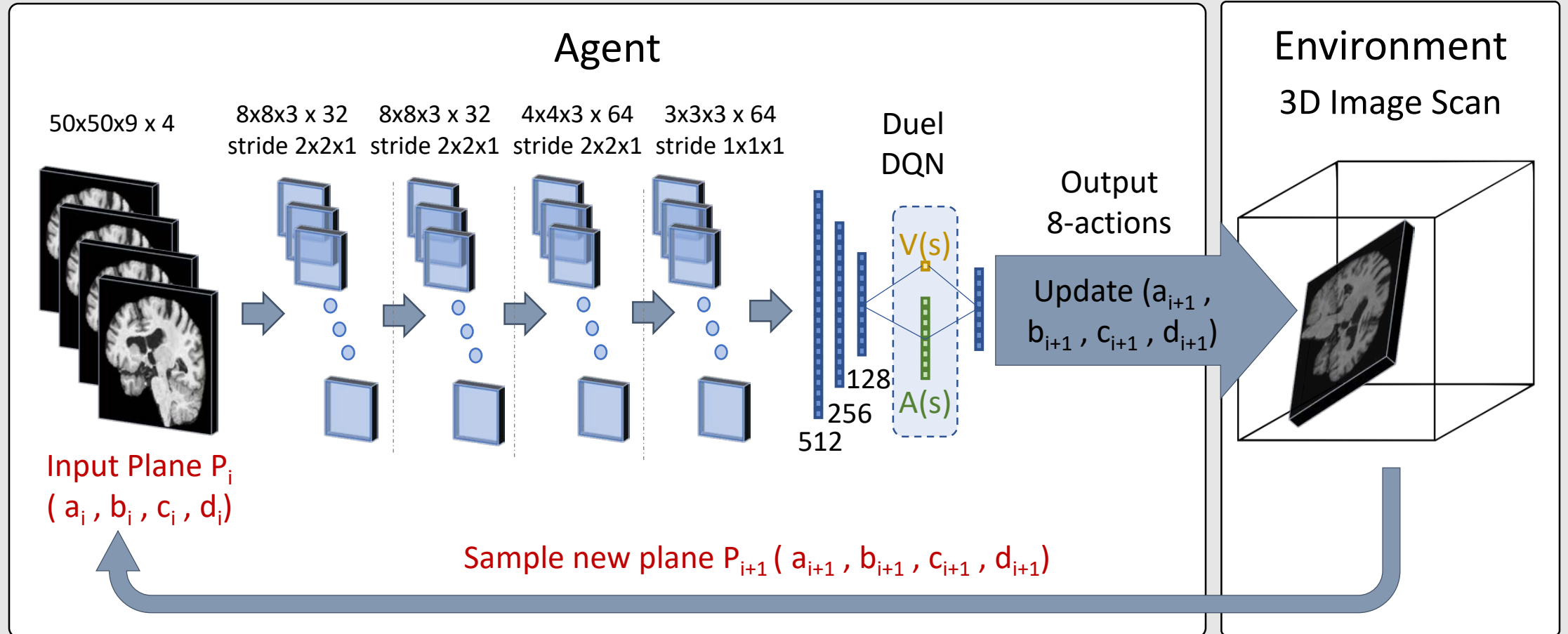
Increasing the network's field of view requires larger memory and higher computational complexity

Solution

- + Multi-scale agent strategy (coarse-to-fine fashion) [Ghesu et al 2017]
 - **Coarser levels** enables the agent to see more structural information
 - **Finer scales** provides more precise adjustments for the final estimation
- + Hierarchical action steps
 - **Larger steps** speed convergence towards the target plane
 - **Smaller steps** fine tune the final estimation of plane parameters



Proposed Pipeline



Improvements on DQN



We experimentally evaluate two recent state-of-the-art variants of the standard DQN

- **Double DQN (DDQN)** H. Van Hasselt 2015

Removes upward bias caused by maximum approximated action value

- Current Q-net θ is used to select actions
- Older target Q-net θ^- is used to evaluate actions

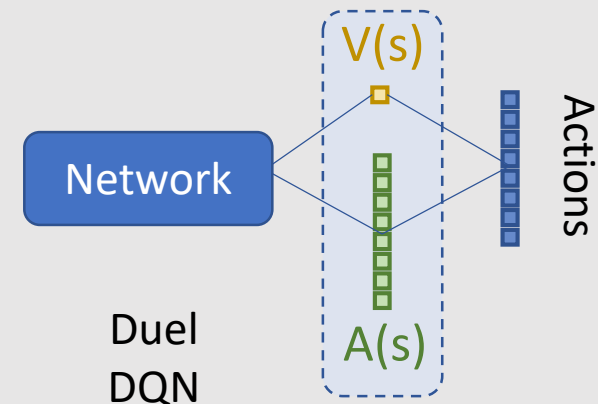
$$L(\theta) = E_{s,r,a,s' \sim D} \left[\left(r + \gamma \max_{a'} Q(s', Q(s', a'; \theta), \theta^-) - Q(s, a; \theta) \right)^2 \right]$$

- **Dueling DQN** Z. Wang 2015

Split Q-net into two channels:

- Action-independent value function $V(s)$
- Action-dependent advantage function $A(s,a)$

$$Q^\pi(s, a) = A^\pi(s, a) + V^\pi(s)$$



Experiments



Experiments:

- Total 12 different experiments
 - 4 different DQN-based methods
 - 3 target planes from 2 different dataset

Evaluation:

- Two metrics:
 - The distance between anatomical landmarks and the detected planes
 - The orientation error by calculating the angle between normal vectors of the detected and target plan

Training



1. Select a random point
2. Define an initial plane using the normal vector from center of the image to the random point
3. Define the origin of the new plane by projecting the center of the input image
4. Sample a plane of size (50,50,9) voxels around the plane origin

Experimental details

- Initial $a_{\theta_x}, a_{\theta_y}, a_{\theta_z} = 8$ and $a_d = 4$
- Every new scale $a_{\theta_x}, a_{\theta_y}, a_{\theta_z}$ decrease by a factor of 2 and a_d decrease 1 unit
- 3-levels of scale with spacing from 3 to 1 mm are used for the brain experiments,
- 4-levels of scale from 5 to 2 mm for the cardiac experiment.

Experiment I – Cardiac MRI



4-Chamber views, commonly used to assess cardiac anomalies

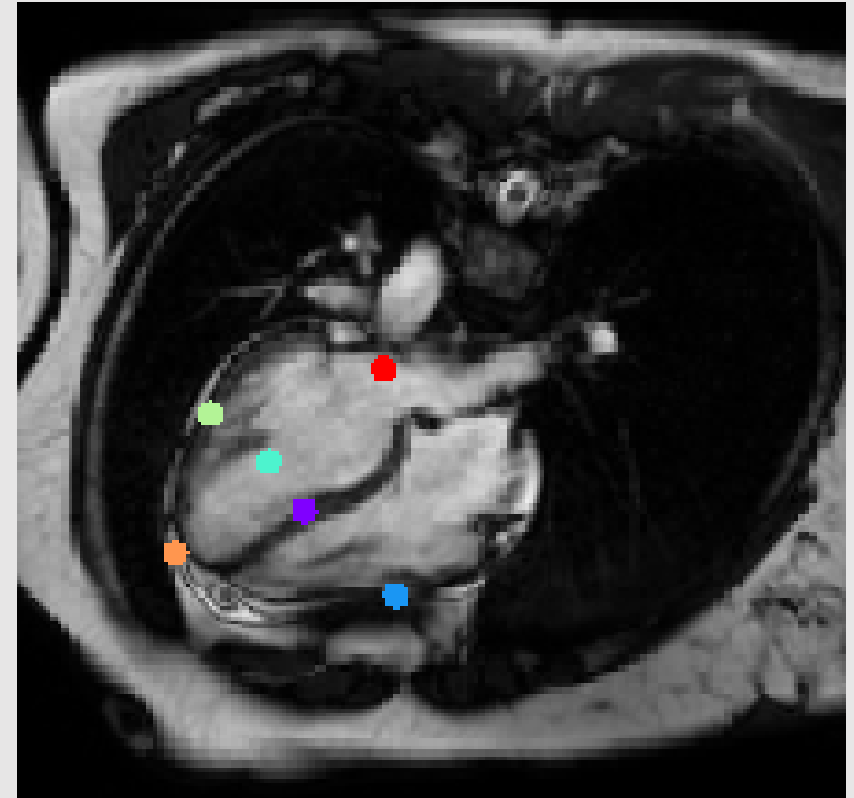
Dataset

- 455 short-axis cardiac MR of resolution 1.25x1.25x2mm obtained from the UK Digital Heart Project ^[1]
- 364 training and 91 testing

Landmarks

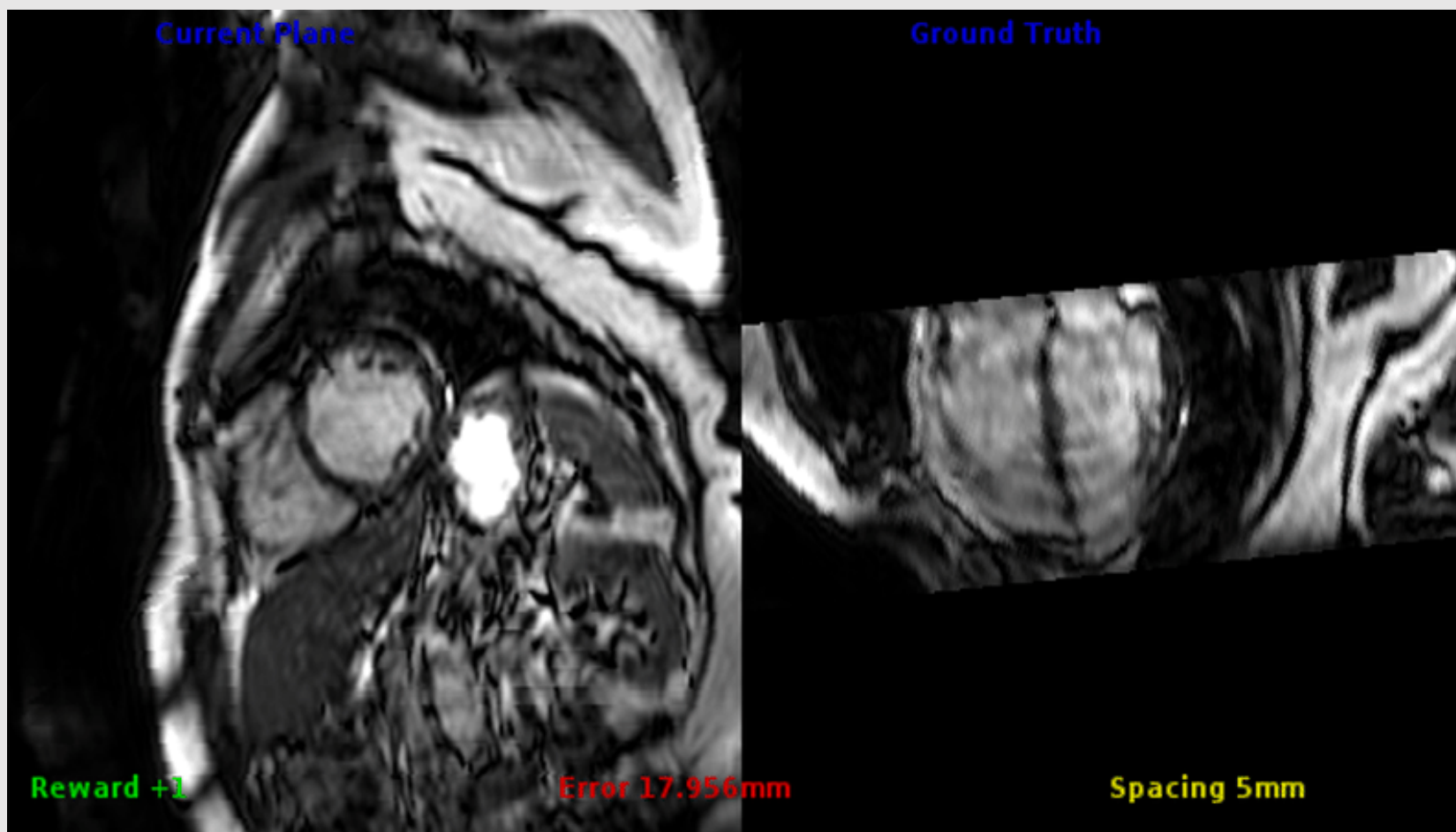
(6 landmarks projected on the 4-chamber plane)

- Two right ventricle insertion points
- Right and left ventricles lateral wall turning points
- Apex
- Center of the mitral valve



[1] Antonio de Marvao, et al. Population-based studies of myocardial hypertrophy: high resolution cardiovascular magnetic resonance atlases improve statistical power. *Journal of Cardiovascular Magnetic Resonance*, 16(1):16, 2014.[42]

Visualizations - 4CH MR-Cardiac



Results



Methods	Landmark-based ^[1]	DQN	DDQN	DuelDQN	DuelDDQN
Distance Error (mm)	5.7 ± 8.5	5.61 ± 4.09	5.79 ± 4.58	4.84 ± 3.03	5.07 ± 3.33
Angle Error (°)	17.6 ± 19.2	10.16 ± 10.62	11.20 ± 14.86	8.86 ± 12.42	8.72 ± 7.44

- Duel DQN-based architectures achieve the best results for detecting the 4-chamber plane
- Agent has to navigate in a bigger field of view
- In contrast to ^[1], our method does not require manual annotation of landmarks

[1] Lu, X., Jolly, M.P., Georgescu, B., Hayes, C., Speier, P., Schmidt, M., Bi, X., Kroeker, R., Comaniciu, D., Kellman, P., et al.: Automatic view planning for cardiac MRI acquisition. In: MICCAI. pp. 479–486. Springer (2011)

Experiment II – Brain MRI



ACPC and mid-sagittal views

Commonly used by the neuro-imaging community for:

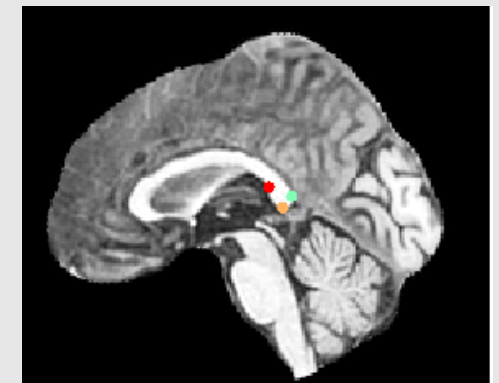
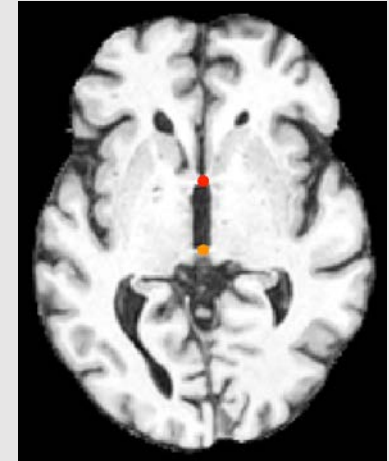
- Initial step in image registration
- Evaluation of pathological brains by estimating the departures from bilateral symmetry in the cerebrum

Dataset

- 832 isotropic 1mm MR scans from the ADNI database [1]
- 728 and 104 images for training and testing

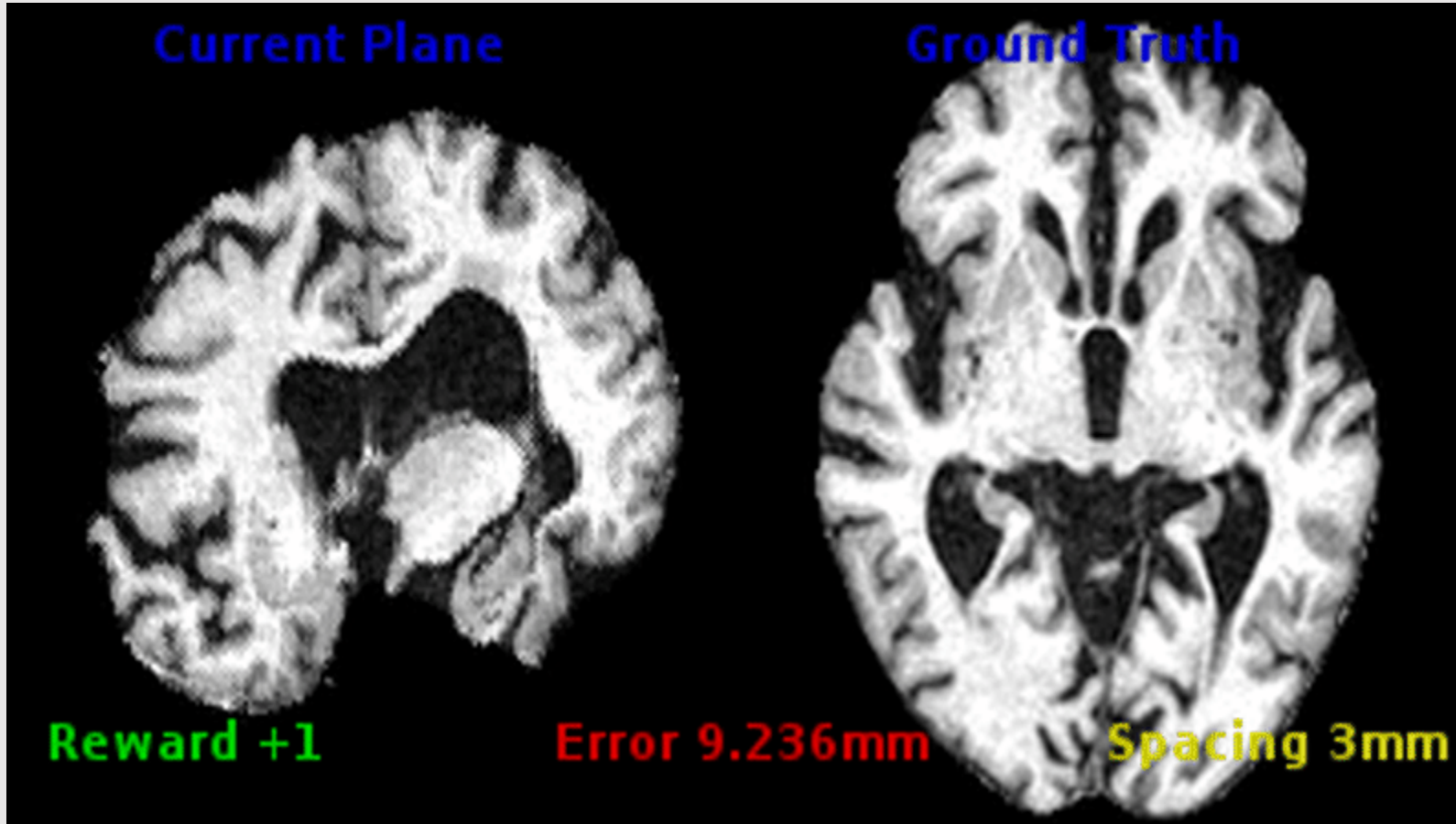
Landmarks: (5 landmarks)

- ACPC
 - Anterior-posterior commissure points (red and yellow)
- Mid-sagittal
 - outer aspect (green)
 - Inferior tip (yellow)
 - Inner aspect (red) points of splenium of corpus callosum



[1] Susanne G Mueller, Michael W Weiner, Leon J Thal, Ronald C Petersen, Clifford Jack, William Jagust, John Q Trojanowski, Arthur W Toga, and Laurel Beckett. The Alzheimer's disease neuroimaging initiative. *Neuroimaging Clinics*, 15(4):869–877, 2005.

Visualizations – ACPC Axial Plane



Results



- Mid-sagittal plane

Error	DQN	DDQN	DuelDQN	DuelDDQN
Distance (mm)	1.65 ± 1.99	2.08 ± 2.58	1.69 ± 1.98	1.53 ± 2.20
Angle (°)	2.42 ± 5.27	3.44 ± 7.46	3.82 ± 7.15	2.44 ± 5.04

- ACPC axial plane

Error	DQN	DDQN	DuelDQN	DuelDDQN
Distance (mm)	2.61 ± 5.44	1.98 ± 2.23	2.13 ± 1.99	5.30 ± 11.19
Angle (°)	3.23 ± 6.03	4.48 ± 14.00	5.24 ± 13.75	5.25 ± 12.64

Runtime

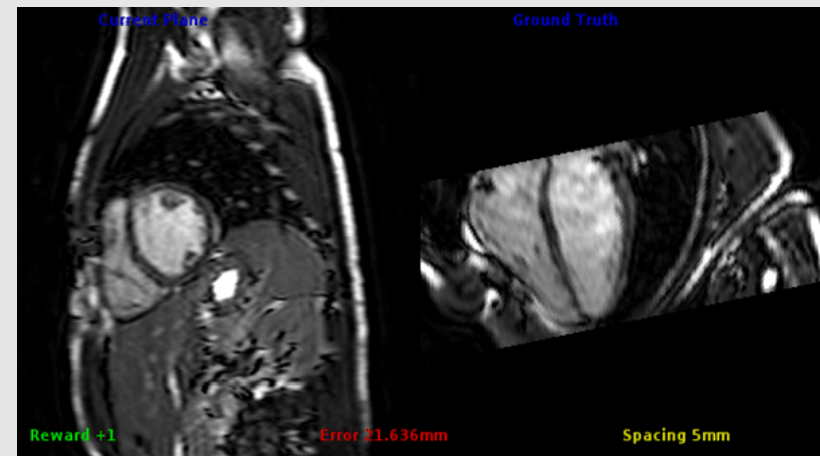
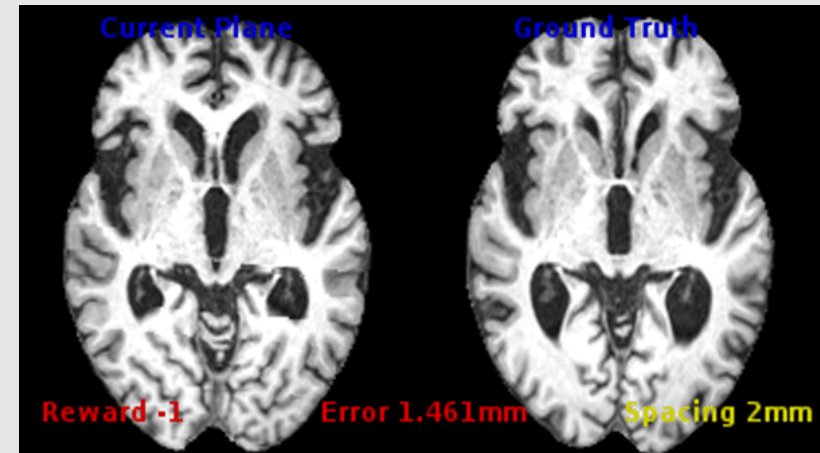


- The agent finds the target location using sequential steps
- Total runtime depends on the starting point – the further it is, the longer it will take to find the target landmark
- Training takes 2-4 days
- Testing can take less than 2 second to find the target plane

Current Challenges



- Noise resulting from sampling errors in different orientations
- Scanned field of view and orientation
- No terminal state by following a long circular path around the target. This can be alleviated by using bigger memory to trace agent's recent path and detect oscillations frequencies



RL-limitations



- Reinforcement learning is a difficult problem that needs a careful formulation of its elements
- Our results show that the optimal algorithm for achieving the best performing agent depends on the target plane (environment-dependent) – similarly on different Atari games

Conclusion



- A novel and feasible reinforcement learning approach for the view planning task that could open up new directions for future improvements
- Our approach is capable of finding standardized planes in short time, which in turn enables accelerated image acquisition

Challenges

- Noise resulting from sampling errors in different orientations
- RL is a difficult problem that needs a careful formulation of its elements
- The optimal algorithm for achieving the best performing agent depends on the target plane (environment-dependent), similar to RL on different Atari games

Future Work



- Competitive/collaborative multi-agents to detect single or multiple views
- Learn from experienced operators by interaction and accumulate their experience, inspired by AlphaGo [D. Silver et al. 2016]

RL References



- [1] Richard S Sutton and Andrew G Barto. “Reinforcement learning: An introduction,” MIT press Cambridge, 1998.
- [2] Christopher JCH Watkins and Peter Dayan. “Q-learning.” Machine learning, 1992.
- [3] Richard Bellman. “Dynamic programming.” Courier Corporation, 2013.
- [4] V. Mnih, et al. “Human-level control through deep reinforcement learning.” Nature, 2015.
- [5] L. Lin. “Reinforcement learning for robots using neural networks.” Technical report, Carnegie-Mellon Univ Pittsburgh PA School of Computer Science, 1993.
- [6] Hado V Hasselt. “Double Q-learning.” Advances in Neural Information Processing Systems, 2010.
- [7] Hado Van Hasselt, Arthur Guez, and David Silver. “Deep Reinforcement Learning with Double Q-Learning.” AAAI, 2016.
- [8] Ziyu Wang, et al. “Dueling network architectures for deep reinforcement learning.” arXiv preprint arXiv:1511.06581, 2015.
- [9] David Silver, et al. “Mastering the game of go with deep neural networks and tree search.” nature, 2016.

Poster Session

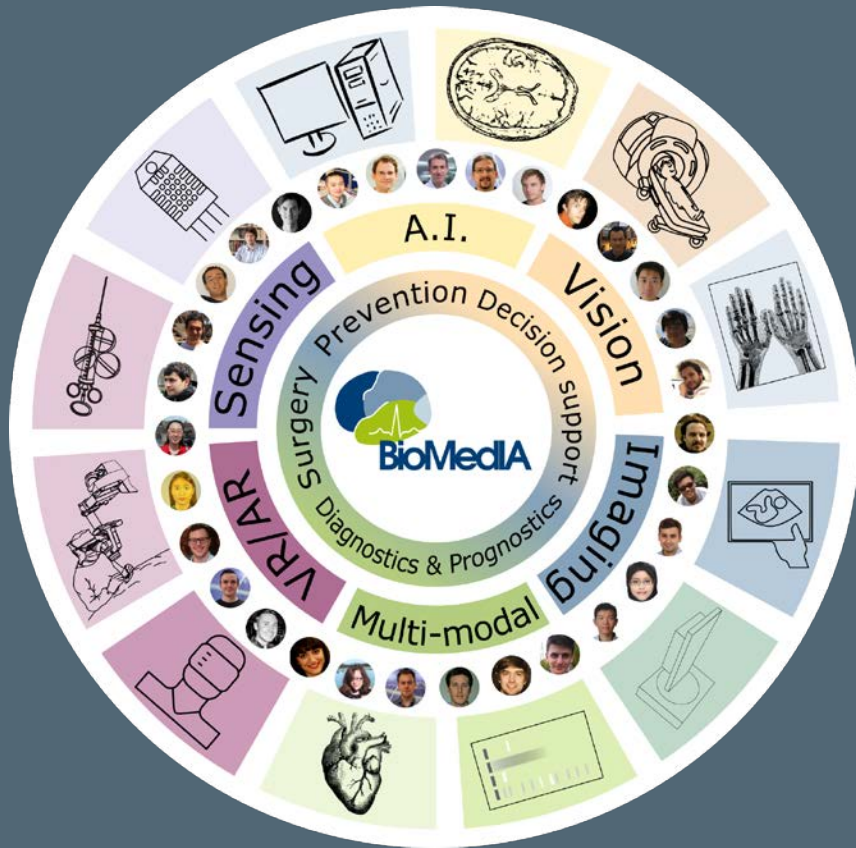
Tomorrow 11:30-12:30

M-33

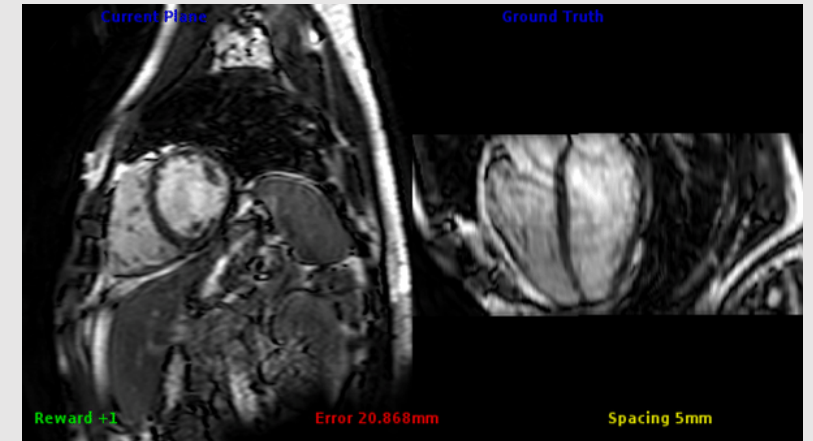
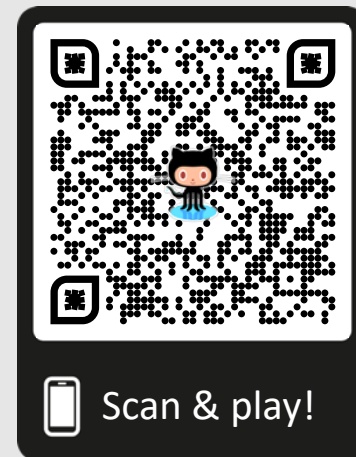
Automatic View Planning with Multi-scale Deep Reinforcement Learning Agents

Amir Alansary, Loic Le Folgoc, Ghislain Vaillant, Ozan Oktay, Yuanwei Li, Wenjia Bai, Jonathan Passerat-Palmbach, Ricardo Guerrero, Konstantinos Kamnitsas, Benjamin Hou, Steven McDonagh, Ben Glocker, Bernhard Kainz and Daniel Rueckert

Thank you!... Questions?



Code is publicly available



<https://github.com/amiralansary/tensorpack-medical>