Success-Aware Visual Navigation Agent

Mahdi Kazemi Moghaddam, Ehsan Abbasnejad, Qi Wu, Javen Qinfeng Shi and Anton Van Den Hengel

Australian Institute for Machine Learning, The University of Adelaide

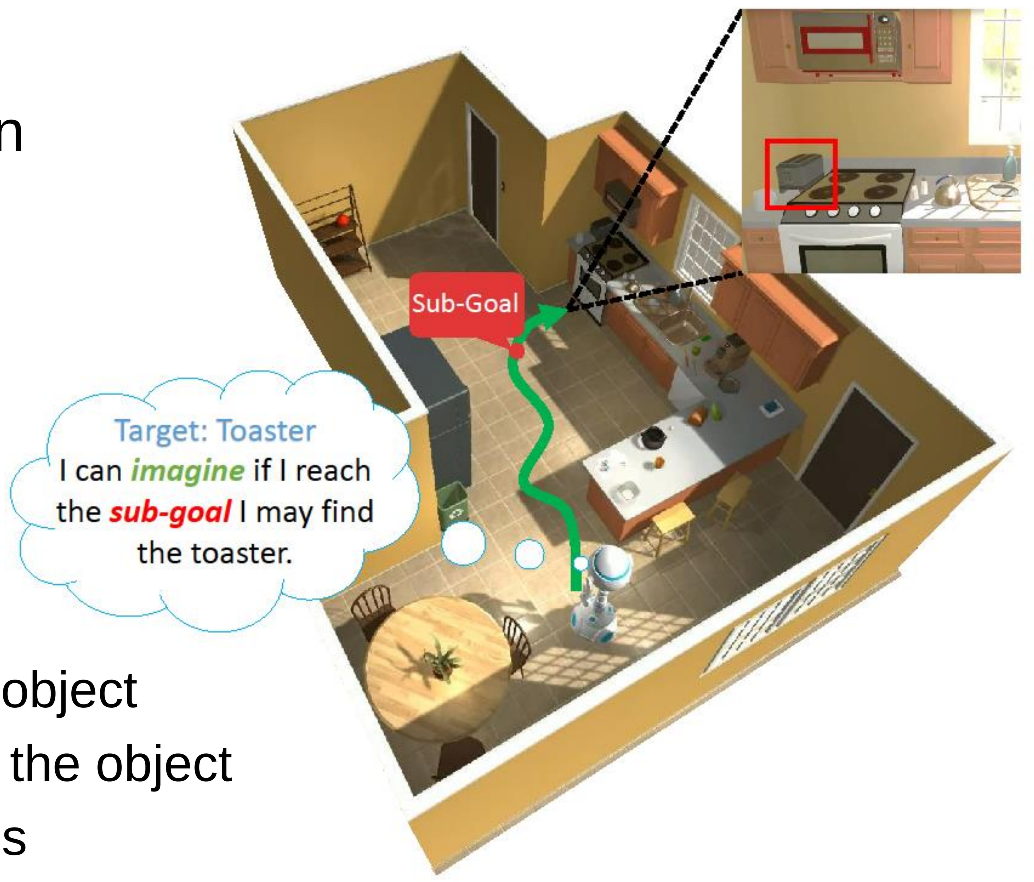
THE UNIVERSITY of ADELAIDE

AUSTRALIAN INSTITUTE FOR MACHINE LEARNING

CVPR VIRTUAL JUNE 19-25

Introduction:

- Object-target visual navigation task:
  - Given an object name perform a sequence of actions to find an instance of a given object and navigate towards it, using visual inputs only
- Challenges:
  - Efficiently explore to localise the object
  - Plan (implicitly) the path towards the object
  - Navigate while avoiding obstacles
  - Select the shortest set of actions (e.g. trajectory) among many possible
- Previous state-of-the-art model-free RL:
  - Map visual representations directly to actions
  - Learn the representation and the policy, that addresses the above challenges, in an end-to-end manner
- Model-based RL:
  - Can look ahead before acting via explicitly modelling the state transitions
  - So far limited to only toy environments such as Atari
  - Challenging to model every state transition in complex 3D environment
- Our approach: *Foresight Success IMaginator* (ForeSIM)



Method Overview:

- We introduce foresight of a successful (sub-) goal state into action selection
- Our method has three main parts:
  - Hindsight sub-goal selection
  - Training sub-goal generator module along with RL
  - Foresight sub-goal generation and action selection
- We build our framework upon A3C (actor-critic) RL algorithm

Hindsight Sub-Goal Selection:

- We modify state-value function and use multi-head attention to learn to select (sub-) goal:

$$V_\theta(\mathbf{s}_t) \approx V_\theta(\mathbf{s}_t^\star), \qquad \mathbf{s}_t^\star = \sum_{j=0}^{t} \alpha_j v_\omega(\mathbf{s}_j) + \mathbf{s}_t \,.$$

$$\boldsymbol{\alpha} = \text{softmax}\left(\frac{q_\omega(\mathbf{s}_t) k_\omega([\mathbf{s}_0 : \mathbf{s}_t])^\top}{\sqrt{t+1}}\right).$$

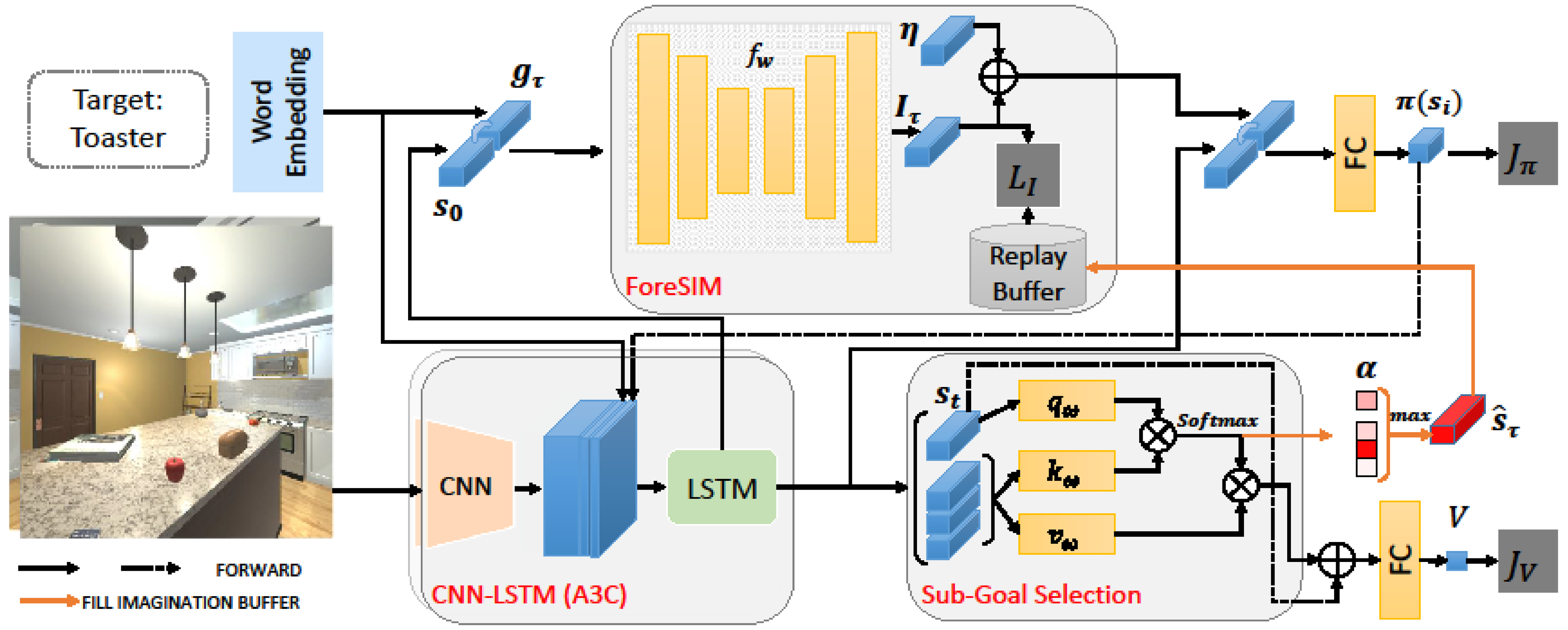
- $v_\omega(s_j)$ is a linear function of the input, α_j is the j-th dimension of $\boldsymbol{\alpha}$ the set of sub-goal weights, q_ω and k_ω are linear functions analogous to query and key in an attention mechanism

Training Sub-Goal Generator:

- We add a replay buffer to alternate between training the policy and our sub-goal generator; we use the following objective to train our sub-goal generator:

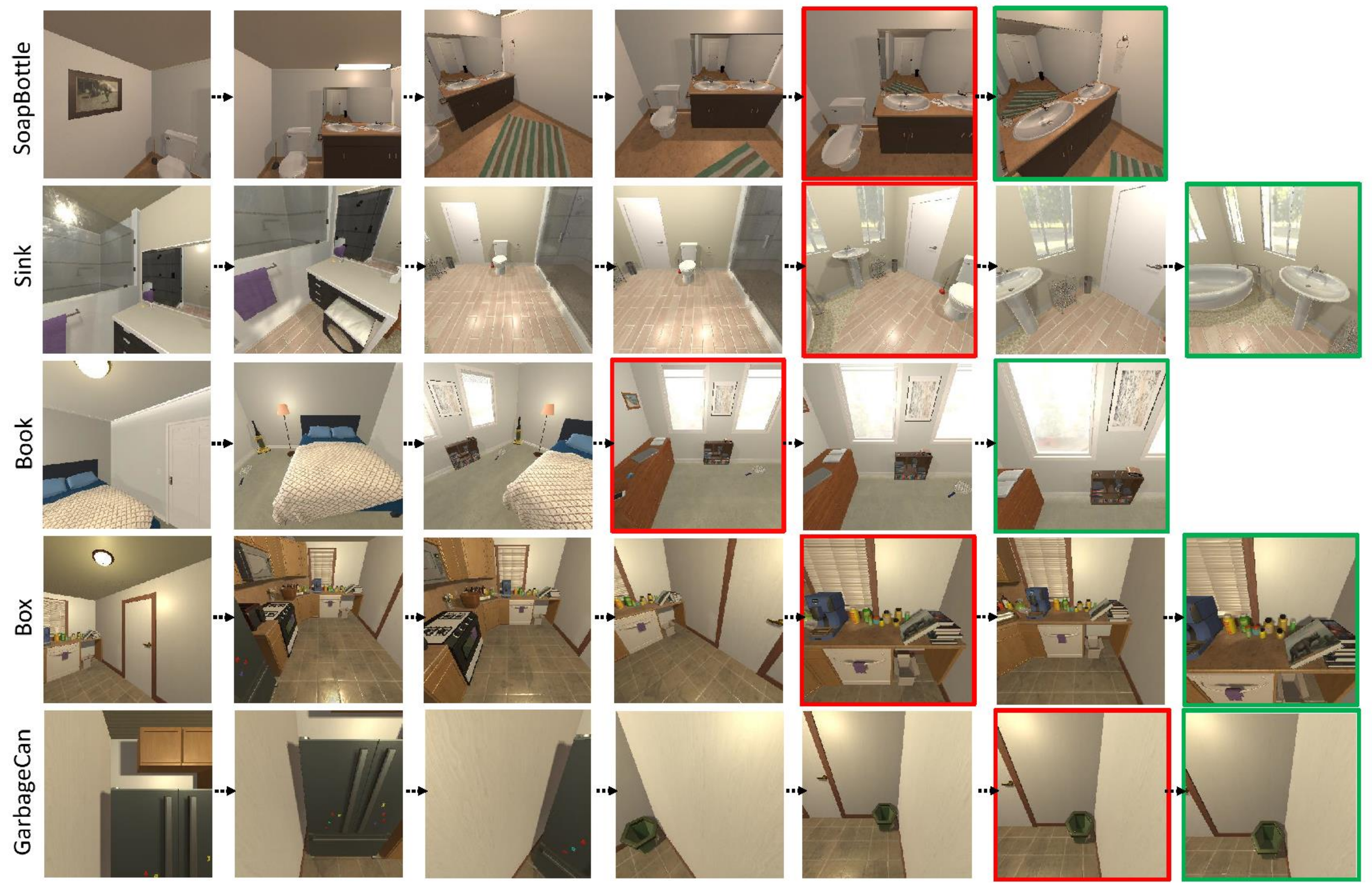
$$\min_{\mathbf{w}} \; \mathbb{E}_{(\mathbf{s}_0, \mathbf{g}_\tau, \hat{\mathbf{s}}_\tau) \sim M} \left| \hat{\mathbf{s}}_t - f_{\mathbf{w}}([\mathbf{s}_0 : \mathbf{g}_\tau]) \right|$$

- Overview of our framework below:



Sub-Goal Selection Results:

- Below are some examples of episodes and the selected sub-goal state (episodes are temporally sub-sampled for visualisation)



Navigation Results:

- We compare our method against multiple different baselines in AI2THOR simulator
- Our method improves both the robustness and efficiency of baselines, measured using Success Rate (SR) and Success weighted by Path Length (SPL)

Method	SPL	SR	SPL>5	SR>5
Without Object Detector				
A3C+MAML	16.15 ±0.5	40.86 ±1.2	13.91 ±0.5	28.70 ±1.5
A3C+MAML+ForeSIM	**16.75** ±0.5	**45.5** ±1.0	**15.8** ±0.6	**34.7** ±1.1
With Object Detector				
A3C+ORG	37.5	65.3	36.1	54.8
A3C+ORG+ForeSIM	**39.41** ±0.3	**68.0** ±0.6	**36.85** ±0.4	**56.11** ±0.8