Success-Aware Visual Navigation Agent

Mahdi Kazemi Moghaddam, Ehsan Abbasnejad, Qi Wu, Javen Qinfeng shi and Anton Van Den Hengel
The Australian Institute for Machine Learning
The University of Adelaide
mahdi.kazemimoghaddam, ehsan.abbasnejad, qi.wu01, javen.shi and anton.vandenhengel@adelaide.edu.au

Abstract

This work presents a method to improve the efficiency and robustness of the previous model-free Reinforcement Learning (RL) algorithms for the task of object-target visual navigation. Despite achieving the state-of-the-art results, one of the major drawbacks of those approaches is the lack of a forward model that informs the agent about the potential consequences if its actions, e.g. being model-free. In this work we take a step towards augmenting the model-free methods with a forward model that is trained along with the policy, using a replay buffer, and can predict a successful future state of an episode in a challenging 3D navigation environment. We develop a module that can predict a representation of a future state, from the beginning of a navigation episode, if the episode were to be successful; we call this ForeSIM module. ForeSIM is trained to imagine a future latent state that leads to success. Therefore, during navigation, the policy is able to take better actions leading to two main advantages: first, in the absence of an object detector, ForeSIM leads mainly to a more robust policy, e.g. about 5% absolute improvement on success rate; second, when combined with an off-the-shelf object detector to help better distinguish the target object, ForeSIM leads to about 3% absolute improvement on success rate and about 2% absolute improvement on Success weighted by inverse Path Length (SPL), e.g. higher efficiency.

1. Introduction

Target-object visual navigation is a challenging problem since the agent needs to learn how to avoid obstacles and take actions by distinguishing similar but visually different targets in a complex environment [1,8,10,13]. Furthermore, there are typically multiple action sequences (*i.e.* trajectories) that could lead to a successful episode at each starting state, let alone the trajectories that might fail. Learning to select the right action at each time step to create a trajectory that leads to the specified target is the primary challenge.

Explicitly incorporating the transition in the environment for better prediction of the potential outcome of the actions, hailed model-based RL, is also developed [4–7]. However, model-based approaches are generally harder to train, especially in a 3D rich environment, since every state transition in the environment has to be accurately modelled.

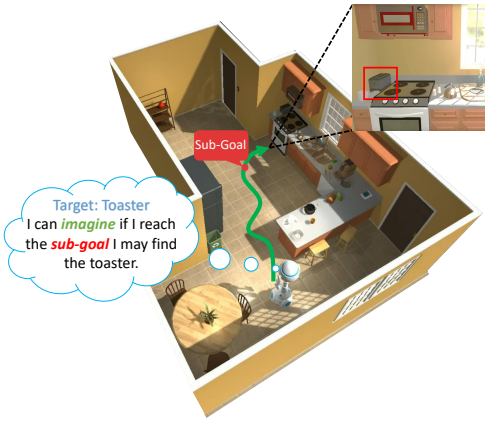

Figure 1. Enabling an agent to imagine states on the path to success improves its ability to carry out complex tasks, particularly in unseen environments. As opposed to the conventional approaches, our success-aware agent takes actions not only based on the current state, but also a prediction of a successful future, to achieve its goal.

To mitigate the above mentioned issues we propose our *Foresight Success IMaginator* (ForeSIM). Intuitively, as shown in Figure 1, ForeSIM is able to simulate a representation of a potential successful trajectory given the target and the initial state (*i.e.* the egocentric RGB view of the environment). ForeSIM provides the agent with foresight of a future sub-goal state through which the agent is more likely to successfully achieve its goal. By explicitly incorporating the sub-goal information into the policy, the agent can take better actions even when the target is not in the field of view of it.

ForeSIM helps the agent in two main ways: first, it provides the agent with an imagined representation of the sub-goal state that will help with successful task completion, e.g. stopping at the right location; our empirical results in the absence of an object detector support this claim; second, it helps the agent to constantly remember the target state to navigate to even if that state is out of the field of view of the agent shortly, e.g. it improves the navigation efficiency.

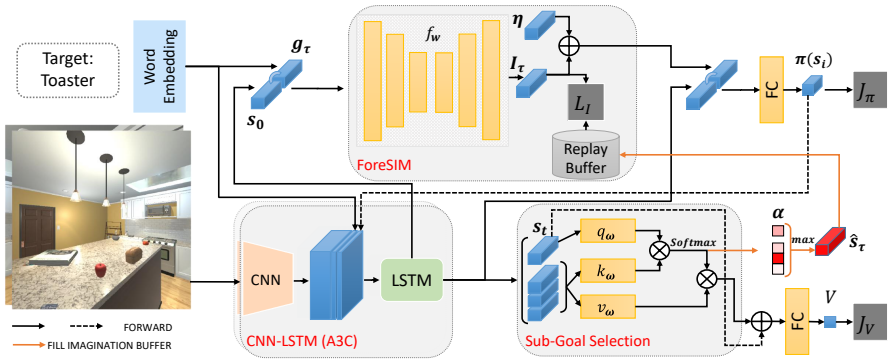

Figure 2. We augment actor-critic RL with our ForeSIM module. We alternate between training ForeSIM and the policy, sharing the same state representation. We develop an attention mechanism to identify the sub-goal state that minimises the critic error (in hindsight) and train our ForeSIM module to generate that state (in foresight).

2. Method

We present an overview of our navigation framework in Figure 2. Our framework involves two main steps: first, we consider learning to identify a sub-goal state through which a navigation episode is successful. To that end, we develop an attention mechanism to find the sub-goal in hindsight after an episode is executed [2]. We modify the value estimation objective in actor-critic RL (A3C) [9] to learn to identify the sub-goal state, as the state the maximally correlates with the successful goal state, as follows:

$$V_\theta(\mathbf{s}_t) \approx V_\theta(\mathbf{s}^\star_t), \qquad \mathbf{s}^\star_t = \sum_{j=0}^{t} \alpha_j v_{\boldsymbol{\omega}}(\mathbf{s}_j) + \mathbf{s}_t \,. \tag{1}$$

Here, $v_{\boldsymbol{\omega}}(\mathbf{s}_j)$ is a linear function of the input and α_j is the jth dimension of $\boldsymbol{\alpha}$, which is defined as follows:

$$\boldsymbol{\alpha} = \text{softmax}\left(\frac{q_{\boldsymbol{\omega}}(\mathbf{s}_t) k_{\boldsymbol{\omega}}([\mathbf{s}_0 : \mathbf{s}_t])^\top}{\sqrt{t+1}}\right). \tag{2}$$

Here, $q_{\boldsymbol{\omega}}$ and $k_{\boldsymbol{\omega}}$ are linear functions analogous to the query and key in an attention mechanism [11] with $\mathbf{s}_{0:t}$ the concatenation of the states up to time t. We denote all of our sub-goal selection parameters by the set $\boldsymbol{\omega}$. Moreover, α_j is the correlation between state j and the current state t, and its magnitude specifies the likelihood that state j is an important sub-goal to reach.

Next, we consider how to learn to generate the selected sub-goal in foresight at the beginning of each episode. To that end, we develop an algorithm that uses a replay buffer of the selected sub-goal states in the past successful episodes. The replay buffer, denoted by M is filled with tuples of $(\mathbf{s}_0, \mathbf{g}_\tau, \hat{\mathbf{s}}_\tau)$, the initial state, the embedding of the target object $\mathbf{g}_\tau$ for episode τ and the sub-goal state representation. We then devise the following objective to train our ForeSIM module, $f_{\mathbf{w}}$, parameterised with $\mathbf{w}$:

$$\min_{\mathbf{w}} \quad \mathbb{E}_{(\mathbf{s}_0, \mathbf{g}_\tau, \hat{\mathbf{s}}_\tau) \sim M} \left| \hat{\mathbf{s}}_t \; - \; f_{\mathbf{w}}([\mathbf{s}_0 : \mathbf{g}_\tau]) \right| \tag{3}$$

where $[:]$ denotes vector concatenation. For $f_{\mathbf{w}}$ we use a multi-layer Perceptron (MLP) with a bottleneck architecture with the intuition that the structure of the sub-goal distribution lies in a lower dimensional space. Finally, we integrate our sub-goal selection and ForeSIM module into A3C [9] actor and critic objectives, as follows:

$$\mathcal{J}^\star_\pi(a_t \,|\, \mathbf{s}_t, \boldsymbol{\theta}) = -\log \pi(a_t \,|\, \mathbf{s}_t, \mathbf{g}_\tau, \mathbf{I}_\tau; \boldsymbol{\theta})(r_t + \gamma V_\theta(\mathbf{s}^\star_{t+1}) - V_\theta(\mathbf{s}^\star_t)) + \beta_H H_t(\pi) \tag{4}$$

$$\mathcal{J}^\star_V(\mathbf{s}_t, \boldsymbol{\theta}) = \frac{1}{2}(V_\theta(\mathbf{s}^\star_t) - R)^2 \tag{5}$$

where $\mathbf{I}_\tau$ is the foresight imagination, V_θ is the state value function approximation, H is the entropy with its weight hyper-parameter β and R is the sum of the discounted reward.

Our policy receives both the generated sub-goal and the current state representation and maps them to actions.

Method	SPL	SR	SPL>5	SR>5
Without Object Detector				
A3C+MAML [12]	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 [3]	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

Table 1. Quantitative comparison to the previous state-of-the-art methods. *SPL>5* and *SR>5* show the metrics for episodes longer than 5 time steps. Our method improves all four commonly used evaluation metrics.

3. Results

In Table 1, we quantitatively compare our method with two prominent previous methods in AI2THOR [8] simulator. First, we add ForeSIM to **A3C+MAML** [12] and we show a significant improvement in success rate. ForeSIM helps with identifying the target object without an object detector and stopping at the right location. Second, we add ForeSIM to **A3C+ORG** [3] and we improve both the success rate and SPL. This shows more efficient navigation even when an off-the-shelf object detector is used to stop at the right location. We follow the exact same setup in both methods [3, 12] for fair comparison.

References

[1] Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sunderhauf, Ian Reid, Stephen Gould, and Anton van den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun 2018. 1

[2] Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, OpenAI Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay. *Advances in neural information processing systems*, 30:5048–5058, 2017. 2

[3] Heming Du, Xin Yu, and Liang Zheng. Learning object relation graph and tentative policy for visual navigation, 2020. 2

[4] David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2018. 1

[5] Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination, 2020. 1

[6] Danijar Hafner, Timothy Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering atari with discrete world models. *arXiv preprint arXiv:2010.02193*, 2020. 1

[7] Lukasz Kaiser, Mohammad Babaeizadeh, Piotr Milos, Blazej Osinski, Roy H Campbell, Konrad Czechowski, Dumitru Erhan, Chelsea Finn, Piotr Kozakowski, Sergey Levine, et al. Model-based reinforcement learning for atari. *arXiv preprint arXiv:1903.00374*, 2019. 1

[8] Eric Kolve, Roozbeh Mottaghi, Winson Han, Eli VanderBilt, Luca Weihs, Alvaro Herrasti, Daniel Gordon, Yuke Zhu, Abhinav Gupta, and Ali Farhadi. Ai2-thor: An interactive 3d environment for visual ai, 2017. 1, 2

[9] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1928–1937, New York, New York, USA, 20–22 Jun 2016. PMLR. 2

[10] Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, Devi Parikh, and Dhruv Batra. Habitat: A platform for embodied ai research, 2019. 1

[11] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017. 2

[12] Mitchell Wortsman, Kiana Ehsani, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi. Learning to learn how to learn: Self-adaptive visual navigation using meta-learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6750–6759, 2019. 2

[13] Fei Xia, Amir R Zamir, Zhiyang He, Alexander Sax, Jitendra Malik, and Silvio Savarese. Gibson env: Real-world perception for embodied agents. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9068–9079, 2018. 1