Extended Abstract: Lifelong Learning for Resource-Constrained Robot Navigation in the Wild

Bo Liu[1], **Xuesu Xiao**[1], **and Peter Stone**[1, 2]
[1]The University of Texas at Austin, [2]Sony AI

Introduction

Classical mobile robots are designed to be adaptive to different navigation environments by in-situ adjustment of the underlying navigation system, such as by sensor calibration (Xiao et al. 2017) or by parameter tuning (Xiao et al. 2020). However, without adjustment from expert knowledge, the untuned system may repeat the same mistakes (e.g. stuck in the same bottleneck) even though it has navigated in the same environment multiple times.

Recent success in using machine learning for mobile robot navigation indicates the potential of improving navigation performance from a robot's past experience in the same environment (Kahn et al. 2018). When facing different navigation environments, however, learning methods cannot generalize well to unseen scenarios: They must re-learn to navigate in the new environments. More importantly, the learned system is prone to *catastrophic forgetting*, which causes the robot to forget what was learned in previous environments (French 1999).

This paper introduces a Lifelong Learning for Navigation (LLfN) framework that addresses the aforementioned challenges: Instead of learning from scratch, the navigation policy is initialized through a classical navigation algorithm, whose navigation performance does not improve with increasing experience. The robot is able to identify its suboptimal actions and learn from them. The navigation performance then improves in a self-supervised manner. When facing different navigation environments, the navigation policy is able to learn to adapt to new environments, while not forgetting how to navigate in previous ones. LLfN is implemented entirely onboard a physical robot with limited memory and computation, and demonstrated to allow the robot to navigate in three different environments (Figure 1). Links to the video and the full version of the paper are provided.[1][2]

Lifelong Navigation Problem

Under the lifelong navigation problem, a mobile robot will sequentially navigates m environments $\{\mathcal{E}_i\}_{i=1}^{m}$. Whenever the robot advances to $\mathcal{E}_k$, it no longer has access to $\{\mathcal{E}\}_{i=1}^{k-1}$.

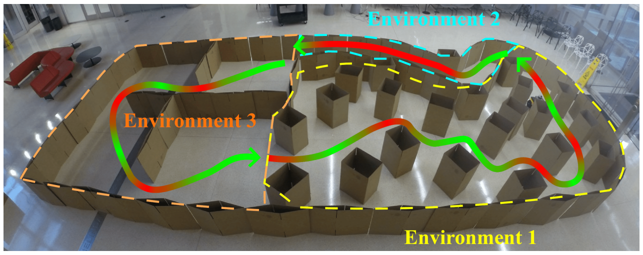

Figure 1: Three navigation environments: An initial navigation policy navigates well most of the time (green), but occasionally behaves suboptimally (e.g. moving extremely slowly or getting stuck). Lifelong Learning for Navigation learns a complementary policy deployed in conjunction with the initial policy, which gradually eliminates the suboptimal behaviors in the current environment while not diminishing performance in previous environments. During deployment, the learned policy is mostly used in the red segments.

Within the environment $\mathcal{E}_k$, the agent, at each time step t, computes a motion command $a_t \in \mathcal{A} \sim \pi_\theta(s_t)$, where $s_t \in \mathcal{S}$ is the agent's state and π_θ is a policy parameterized by θ. The goal is learn the best policy π_θ that can navigate all m environments after visiting them in the sequence. The key challenge is caused by the catastrophic forgetting problem of typical learning methods, which means learning in new environment will downgrade the performance in previous environments. Specifically, we constrain that the robot has a small fixed-size memory buffer to store any data.

Gradient Episodic Memory

Gradient Episodic Memory (GEM) (Lopez-Paz and Ranzato 2017) prevents forgetting by ensuring each learning update in new environment will not increase the loss on previous tasks. Specifically, assume the agent has already seen environments up to $\mathcal{E}_{k-1}$ and the learned policy is $\pi_{\theta_{k-1}}$. GEM assumes the agent keeps a small memory buffer $\mathcal{B} = \{\mathcal{M}_i\}_{i<k}$ that, for each previous environment $\mathcal{E}_i$, stores a few exemplary data points $\mathcal{M}_i$. GEM then optimizes the following objective:

$$\min_{\theta} \ell(\pi_\theta, \mathcal{E}_k),\ \text{s.t.}\ \ell(\pi_\theta, \mathcal{M}) \le \ell(\pi_{\theta_{k-1}}, \mathcal{M}),\ \forall \mathcal{M} \in \mathcal{B}, \tag{1}$$

[1] Video at https://tinyurl.com/lifelonglearningfornavigation.

[2] Paper at https://arxiv.org/pdf/2007.14486.pdf.

where $\ell(\pi, X)$ is the loss function that evaluates performance of π on data/environment X. For instance, if we store past state-action pairs (s, a) as demonstrations and use behavior cloning to maintain the performance on those states, then $\ell(\pi, X) = \mathbb{E}_{(s,a)\sim X}||\pi_\theta(s) - a||_2$. To efficiently solve the above optimization, GEM observes that the constraints are satisfied as long as 1) the new θ is initialized from θ_{k-1}, and 2) at each optimization step, the loss on previous tasks does not increase. Assume the optimization steps are small, we can determine whether a new update increases the loss on a previous task by computing the inner product between the gradients on the current and previous tasks. The optimization problem then becomes

$$\min_\theta \ell(\pi_\theta, \mathcal{E}_k), \text{ s.t. } \langle \frac{\partial \ell(\pi_\theta, \mathcal{E}_k)}{\partial \theta}, \frac{\partial \ell(\pi_\theta, \mathcal{M})}{\partial \theta} \rangle \geq 0, \forall \mathcal{M} \in \mathcal{B}. \tag{2}$$

In practice, GEM solves the above optimization efficiently by solving its dual problem using a quadratic program solver. GEM can maintain the learned knowledge well by only storing a few data points from the past tasks, which is particularly suitable for lifelong navigation of mobile robots.

Lifelong Learning for Navigation

We propose the Lifelong Learning for Navigation framework (LLfN) to tackle the lifelong navigation problem. Specifically, LLfN interleaves between a classical sampling based planner π_0 and a learnable planner π_θ. During execution, LLfN records any state (e.g. s) where sub-optimal navigation behavior occurs (e.g. moving too slowly or doing recovery behavior). Then it uses π_0 to sample motion commands until it goes through the difficulty at s. LLfN then looks for an action a from the trajectory nearby s that possibly leads to its success and records (s, a) into the memory buffer. Finally, π_θ learns from (s, a) while using GEM to maintain previously learned behaviors. Specifically LLfN has the following components:

- An initial sampling-based navigation planner π_0 and a learnable policy π_θ, parameterized by θ.
- A scoring function $D : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ that evaluates how good an action a is at the state s, i.e. larger $D(s, a)$ indicates a is a better action at s. In practice, we use $D(s, a) = a_v$ where a_v is the velocity.
- A streaming memory buffer $\mathcal{B}_{\text{stream}}$ that stores the past T-step trajectory, i.e. $\mathcal{B}_{\text{stream}} = \{s_j, a_j\}_{j=t-T+1}^{t}$.
- A per-environment memory $\mathcal{M}_k : |\mathcal{M}_k| = n/k$ (n is the memory budget) that stores the exemplar training data (self-generated data that are worth learning from) from environment $\mathcal{E}_k$. The entire memory before entering $\mathcal{E}_k$ is therefore a set of sets: $\mathcal{B} = \{M_i\}_{i<k}$.
- An algorithm A_{correct} that given a recent suboptimal behavior $(s, a) \in \mathcal{B}_{\text{stream}}$, finds exemplar training data $(s', a') \in \mathcal{B}_{\text{stream}}$ such that learning from (s', a') improves the navigation performance at s.
- A continual learning algorithm A_{cl} (e.g. GEM) that updates π_θ given $\mathcal{M}_k$ and $\mathcal{B}$. A_{cl} should retain performance on previous environments.

The LLfN algorithm is summarized in Algorithm 1.

Algorithm 1 Lifelong Learning for Navigation (LLfN)

```
Inputs: $\pi_0$, $\pi_\theta$, $D$, $A_{\text{correct}}$, $A_{\text{cl}}$, $\mathcal{E}_{k=1}^{m}$, and a threshold $\eta$.
$\mathcal{B} \leftarrow \emptyset$, $\mathcal{B}_{\text{stream}} \leftarrow \emptyset$, and initialize $\theta_0$ randomly
// Training
for environment $k = 1 : m$ do
    $\mathcal{M}_{\text{k}} \leftarrow \emptyset$
    while navigating in $\mathcal{E}_k$ do
        progress to state $s_t$ and generate $a_t \sim \pi_0(s_t)$
        execute $a_t$ and update $\mathcal{B}_{\text{stream}}$ with $(s_t, a_t)$
        let $p = \lfloor t - T/2 \rfloor$ and select $(s_p, a_p) \in \mathcal{B}_{\text{stream}}$
        if $D(s_p, a_p) < \eta$ then
            $(s', a') = A_{\text{correct}}(s_p, \mathcal{B}_{\text{stream}})$
            update $\mathcal{M}_{\text{k}}$ with $(s', a')$
        end if
    end while
    $\theta_k \leftarrow A_{\text{cl}}(\pi_{\theta_{k-1}}, \mathcal{M}_{\text{k}}, \mathcal{B})$        ▷ Lifelong learning
    Shrink $\mathcal{B}$ to size $(n - |\mathcal{M}_{\text{k}}|)$ and $\mathcal{B} = \mathcal{B} \cup \{\mathcal{M}_k\}$
end for
// Execution
while navigating in $\mathcal{E}$ do
    progress to state $s_t$
    generate $a_0 \sim \pi_0(s_t)$, $\hat{a} \sim \pi_{\theta_k}(s_t)$
    execute $a_t = \text{argmax}_{a\in\{a_0,\hat{a}\}} D(s_t, a)$
end while
```

Conclusion

We present the lifelong navigation problem and propose Lifelong Learning for Navigation framework (LLfN) as a solution. Specifically, LLfN enables a mobile robot to continually learn across environments without forgetting in a fully self-supervised fashion.

References

French, R. M. 1999. Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences* 3(4): 128–135.

Kahn, G.; Villaflor, A.; Ding, B.; Abbeel, P.; and Levine, S. 2018. Self-supervised deep reinforcement learning with generalized computation graphs for robot navigation. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 1–8. IEEE.

Lopez-Paz, D.; and Ranzato, M. 2017. Gradient episodic memory for continual learning. In *Advances in neural information processing systems*, 6467–6476.

Xiao, X.; Dufek, J.; Woodbury, T.; and Murphy, R. 2017. UAV assisted USV visual navigation for marine mass casualty incident response. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 6105–6110. IEEE.

Xiao, X.; Liu, B.; Warnell, G.; Fink, J.; and Stone, P. 2020. APPLD: Adaptive Planner Parameter Learning From Demonstration. *IEEE Robotics and Automation Letters* 5(3): 4541–4547.