Learning from Graphical Replay

Ge Yang*, Amy Zhang*†, Ari Morcos†, Joelle Pineau†‡ Pieter Abbeel§ and Roberto Calandra†
*Equal Contribution †Facebook AI Research ‡McGill University §UC Berkeley

I. Introduction

The ability to quickly adapt to changing situations is a critical feature in general intelligence. This is a type of *fast* learning that happens instantaneously as events occur [Botvinick et al., 2019], which relies on effective gathering of relevant new data, and a strong inductive bias over learning. In contrast, model-free reinforcement learning methods often take a delayed approach to both problems. Sampling is often done by injecting noise and past experience sits in a disorganized fashion in a linear buffer, making it difficult to accomplish fine-grained updates that overwrite specific prior notion under environmental non-stationarity.

In this work, we approach the problem of reinforcement learning (RL) as finding ways to answer two questions at the same time: how to efficiently sample data, and how to continuously learn once the data comes in. As humans, our ability to construct a mental map during problem solving underlies our ability to contextualize both [Tolman, 1948]. Much of the focus in exploration so far concern either trade-offs due to uncertainty or different ways to expand the data support over the state space. In particular, we focus on *episodic exploration* that is directed towards a specific but dynamically specified goal, under the guidance of a structured cognitive map of the domain.

A promising model-based approach inspired by cognitive mapping [Savinov et al., 2018] combines a low-level parametric policy with long-horizon planning on a semi-parametric topological map (SPTM), to accomplish goal-reaching tasks in complex visual maze domains. This approach is significant in two ways: first, the graph is episodic in nature, which means an agent could act upon new experience without going through a slow, gradient-based learning process. Second, this graph provides structure for organizing the storage of knowledge (learning), which enables contextual application of focused local updates without affecting prior knowledge saved elsewhere. A missing piece however, is a way to directly involve this world model when learning the reactive controls, for both faster adaptation and improved performance over long-horizon tasks.

To tackle this issue, we present *learning from graphical replay (LfGR)*, a framework that tightly integrates learning a semi-parametric graphical world model with learning parametric reactive control in an iterated learning loop. We improve existing model-free and model-based methods in three major ways: First, we make efficient use of off-policy dynamics data by introducing model-based graphical replay, as opposed to linear replay from regular buffers. Second, to learn truly long-horizon value estimates and a parameterized policy that can accomplish a large number of tasks, we replace traditional value-bootstrapping and policy gradient objectives with two simple, supervised scheme that distills directly from shortest-paths on the graph. Central to our approach is continuous model improvement driven by path planning on the graph that exposes discrepancies between the model and the true environment dynamics. We demonstrate our agent's ability to learn goal-conditioned reactive skills under extreme budgets on sample complexity and memory, and robustness in the face of non-stationarity.

II. Technical Background

Learning generalized universal value estimates directly with 1-step Q-learning is known to be difficult [Florensa et al., 2019, Jurgenson et al., 2019, Hartikainen et al., 2019, Yang et al., 2020]. Hindsight experience replay (HER Andrychowicz et al. 2017) takes advantage of the structure in achieving sub-goals, to populate the replay buffer with relabeled transition tuples $\langle o_t, a_t, R(o_{t+1}, \hat{o}), o_{t+1}\rangle$ containing positive rewards. This is a way to reuse sampled dynamics to supervise the value function with ground-truth reward over shorter distances. Despite its success, HER is limited to linearly relabeling goals within each trajectory. Both learning to reach a large set of goals, and learning directly from pixel input remain challenging. *In this work, we take a model-based approach to extend linear relabeling to a graphical relabeling scheme on a learned graphical model.*

III. Learning from Graphical Replay

Learning from graphical replay (LfGR) decomposes learning long-horizon visuomotor control into two problems: First, learning and improving a graphical world model. Second, supervised learning of a parameterized reactive controller π_θ from the graph.

A. Bayesian Approach to Building A Graphical World Model

When modeling realistic, contact rich environment dynamics, an informative prior goes a long way in reducing the amount of data needed. While manually specifying such priors can be done, we take a more general approach by fitting a crude graphical world model using noisy background observation data, then iteratively updating parts of the model relevant to our specific task with more focused exploration (Fig.1,2).

In particular, we pose the modeling problem as learning a graph $\mathcal{G} = \{\mathcal{E}, \mathcal{V}\}$, where each vertex $v_i \in \mathcal{V}$ corresponds to an observation o_i, and the edge weight e_{ij} corresponds to the likelihood a control policy $\pi(a|o_i, o_j)$ successfully

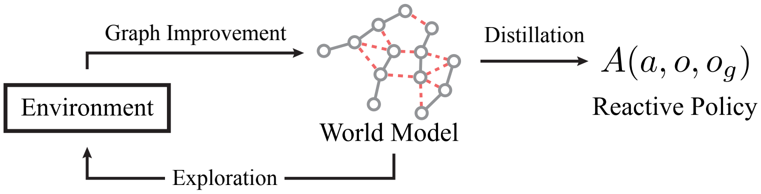

Fig. 1: Learning from Graphical Replay (**LfGR**)

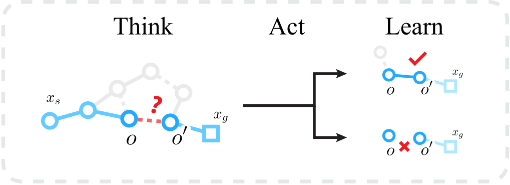

Fig. 2: Model-driven Exploration

traverses between the corresponding states. We assume the existence of a *perceptual distance* d from which we can seed the prior probability of each edge using a threshold hyperparameter d_o via $p(\varepsilon \in \mathcal{E}) = \exp(-\beta d(o', o_j))$, where $o' \sim \mathbb{P}(o' | \, o_i, a)\pi(a|o_i, o_j)$. Under this view, we can seed all $p(\varepsilon \in \mathcal{E}) = 1$ if $d(o_i, o_j) < d_0$, and then continuously update each edge individually with new evidence. Finding the *optimal* plan on this graph hence corresponds to finding the *most likely* path for a policy π to traverse. We learn the perceptive distance metric $d(o, o') \to \mathbb{R}^+$ by constructing a noise-contrastive objective using positive pairs sampled from the joint distribution $p(o, o')$ and negatives from the noisy product of the marginals $p(o)p(o')$,

$$\mathcal{L}_d = |d_\phi(o_t, o_{t+1}) - |a|\,| + |-\min(-d_\phi(o_t, o'), 0) - 2\langle|a|\rangle|\,. \tag{1}$$

We use a novel negative hinge loss in the second term, such that this objective has zero gradient for the negative pairs as long as they are more than 2 steps apart in the latent space. We use 2× the sample-mean action norm as the threshold. We further decompose d into an embedding function $\phi(o) \to z$ and an ℓ^p metric in the latent space to allow fast pair-wise distance computation $d(o, o') = \|z - z'\|_p$. We warm start learning by first training on random exploration data $\langle o_t, a_t, o_{t+1}\rangle$ that are saved in a regular linear replay buffer $\mathcal{B}$. Then we continuously add new experience to $\mathcal{B}$ as the agent acts in the environment. To grow the graph, we follow Laskin et al. and only insert new observation o_t into $\mathcal{G}$ if the closest vertex in the graph is farther than $1/2$ of the mean action norm $\langle|a|\rangle$. This is an important density-regularization that makes storage efficient, by maintaining close to uniform sample distribution over the state space.

B. Goal-directed Exploration And Causal Graph Improvement

The graph we obtained so far offers a crude approximation of the true environment dynamics including friction and physical contact. When we use this crude graph to make plans, we can often map the agent's failure to faulty transitions on the graph that *underestimate* the cost for traversal. In the proximal dynamic programming literature, overestimation of value-to-go [van Hasselt et al., 2015] is a common problem with Q-learning. Yet, in classical AI planning literature, an optimistic value heuristic is *required* for A^* to find the *shortest-path* [Hart et al., 1968]. This duality between the *admissibility* criteria in search and optimism in a failing behavior policy is often overlooked by contemporary research. Under this light, we propose to use planning-based methods' susceptibility to modeling mistakes to drive effective exploration. We refer to this as model-based, goal-directed exploration (Fig.2).

The procedure is as follows: for a given goal g specified via a goal image o_g, the agent comes up with a foresight plan τ^* using heuristic search on its internal world model $\mathcal{G}$ (Fig.2 "*Think*"). At each step during implementation (Fig.2 "*Act*"), the 1-step plan between o and o' is a hypothesis that this transition is implementable. If the agent reaches a next state o_{t+1} that is within a threshold $d(o_{t+1}, o') \leq d_0$ then we consider this edge verified. Otherwise, we choose the null hypothesis (Fig.2 "*Learn*"). Instead of removing this edge, we also apply "*soft*" edge updates resembling computing the Bayesian posterior, the derivation of which we include in the appendix. We additionally search for edges that are similar to $\langle o, o'\rangle$ within a perception distance d_1 for more efficient updates.

C. Universal Value Prediction Network (UVPN)

Following dueling networks [Wang et al., 2015], we decompose the Q-function into learning an optimal state-value $V(s, g)$ and an advantage $A(a, s, g)$:

$$Q(a, s, g) = A(a, s, g) + V(s, g) \tag{2}$$

where $V(s, g) = Q^*(a^*, s, g)$ and $A(a^*, s, g) \equiv 0$. To learn the optimal state value V, we directly generate value targets by finding the shortest paths between randomly sampled pairs of vertices on the graph [Yang et al., 2020]. We combine this planning based long-horizon target with the local distance target from Eq.1 into the following objective, involving two distributions $\mathcal{L}_\text{V} = \mathcal{L}_\text{d} + \mathcal{L}_{\tau^*}$ where

$$\mathcal{L}_{\tau*} = \left| V^*(o_s, o_g) - \sum_{\tau^*(s,g)} d(o_t, o_{t+1}) \right|\,. \tag{3}$$

We can construct a target for learning the advantage function by re-arranging Eq.2:

$$A(a_t, o_t, o_g) = V(o_t, o_g) - [V(o_t, o_{t+1}) + V(o_{t+1}, o_g))]. \tag{4}$$

The action a_t on the *l.h.s.* is sampled as part of the transition tuple $\langle o_t, a_t, o_{t+1}\rangle$ together with o_t and o_{t+1}. The goal observation o_g is sampled independently from the rest of the replay buffer.

Algorithm 1 Universal Value Prediction Network

```
Require: Graph $\mathcal{G}$, search procedure $S$
Require: Value Estimator $V$ (Eq.3 and appendix)
Sample transition $\langle o_t, a_t, o_{t+1}\rangle$ from buffer
Sample $o_g$ from $\mathcal{G}$.
find optimal path $\tau^* = S(G, o_t, o_g)$, where
    $\tau = \{o_t, o_1, o_2, \cdots o_g\}$
for each epoch do
    minimize $\mathcal{L} = |A(a, o_t, o_g) - $ (Eq.4 r.h.s)$|$
```

D. Goal Relabeled Expert Distillation (GRED)

In GRED, we use a learned local inverse model to label optimal plans made on the graph with action, and use these constructed trajectories to train the long-horizon reactive policy. The local inverse model is trained with the standard maximum likelihood objective: for transition $l_{\text{Inv}}(a|o, o')$

$$\mathcal{L}_l = -\log \frac{l_{\text{Inv},\theta}(a_t|o, o')}{\sum_{a_i \sim \mathcal{A}} l_{\text{Inv},\theta}(a_i|o, o')}. \tag{5}$$

Algorithm 2 Goal Relabeled Expert Distillation

```
Require: Graph $\mathcal{G}$, search procedure $S$
Require: Inverse model $l_{\text{Inv}}$ from Eq.1
1: Sample $o_s, o_g$ from $\mathcal{G}$.
2: find optimal path $\tau^* = S(G, o_s, o_g)$, where
   $\tau = \{o_s, o_1, o_2, \cdots o_g\}$
3: for each epoch do
4:     minimize $D_{\text{KL}}\|l(o_s, o_g),\ l^{\top}_{\text{Inv}}(o_s, o_1)\| + L_{\text{Inv}}$
```

$(\cdot)^{\top}$ blocks gradient propagation.

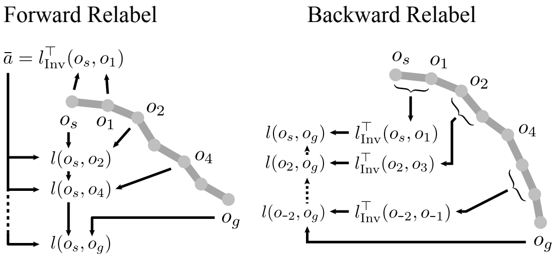

Fig. 3: Relabel Scheme for GRED.

We experimented with two variants for the goal relabeling: *forward* and *backward*. To generate action proposals for a plan made on the episodic memory graph, we need to run observations through a pre-trained inverse model $l^{\top}$. The **forward relabel** scheme only generates one action proposal $l^{\top}_{\text{Inv}}(o_s, o_1)$, but relabels the goal entry in $l(o_s, \mathbf{o'_g})$ with o'_g sampled every k steps in τ. The **backward relabel** scheme relabels the starting position in $l(\mathbf{o'_s}, o_g)$ while keeping the far-away goal o_g the same. This requires computing the action potential for a number of $\langle o'_t, o_{t+1}\rangle$ pairs using $l^{\top}$. The extra action proposal cost is offset by much more diverse labels, yielding better performance.

IV. Results

In the following experiments, we show (i) our graph enable focused adaptation behavior, with minimal sample complexity. (ii) we learn a reactive policy directly from the graphical model. Videos can be found at https://graphical-replay.github.io.

First, we re-enact the “kerplunk” experiment [Waston, 1914]. We start with a pre-trained agent that has mastered navigation of a maze, where it has learned to move around the wall to reach the goal on the other side (Fig.4a). Now, we insert a vertical separation to split the maze into 4 rooms. To prevent the wall from interfering with the convolution network that is trained on the original maze, we make the newly inserted wall transparent for the agent. We color these walls in red (Fig.4b) for illustration purposes.

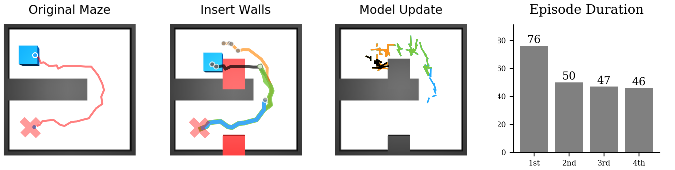

Fig. 4: Adapting to changes in the environment. **(a)**: Original plan made by the agent in Maze. **(b)**: we insert a vertical separation, shown as two red blocks. Colored lines shows the forward plan at 4 separate steps in the same episode. **(c)**: colored segments showing the edges removed at each step. **(d)** The time the agent takes to traverse the maze decreases as it improves the graph.

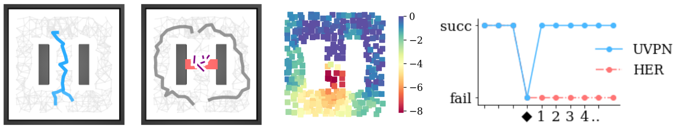

Fig. 5: UVPN and GRED directly distill from the graph, which allows local changes to quickly propagate to affect global policy change without requiring additional samples from the environment. **(a, b)** Path before and after blockage. Agent starts from the top middle to reach two goals separated by a small distance. **(b)** Purple segments indicate edges removed by the agent during the 1st trial. **(d)** The increase in distance estimate (negative value) after removing the edges along the upper wall, showing global change in value estimates due to local experience. The entire sidewalk incurs a change in value between $(-2, -4)$. Distances are measured from the starting position in the top-middle. **(e)** Success rate vs sample against linear replay (HER). Diamond indicates when the red block is added. UVPN agent recovers quickly. Agent learning from linear replay *never* manages to recover because older experiences remain in the buffer.

V. Conclusion

We have presented a new framework – Learning from Graphical Replay (LfGR) – for learning reactive visuomotor policies directly from a structured graphical world model. We believe this is a step towards unified agents that can "think, fast and slow".

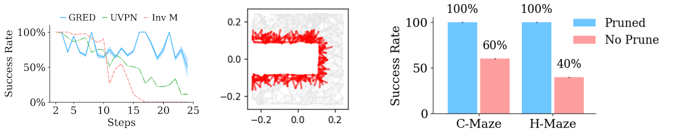

Fig. 6: **(a)** Reactive performance at different planning horizon, with a *partially pruned graph*. Inverse model baseline drops off quickly as the goal moves farther away, whereas UVPN and GRED using a partially pruned graph maintain a slower drop-off. All traces averaged over three seeds and 200 task configurations. **(b)** Example of edges removed through goal-directed exploration. **(c)** Performance of the parametric policy learned with GRED on C-Maze and H-Maze, with and without model-improvement. Model improvements greatly increase the performance of the learned reactive policy by avoiding physical contact. Evaluated over 5 task configurations, averaged over 3 seeds.

References

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. In *Advances in Neural Information Processing Systems*, pages 5048–5058, 2017.

Matthew Botvinick, Sam Ritter, Jane X Wang, Zeb Kurth-Nelson, Charles Blundell, and Demis Hassabis. Reinforcement learning, fast and slow. *Trends Cogn. Sci.*, 23(5): 408–422, May 2019. ISSN 1364-6613, 1879-307X. doi: 10.1016/j.tics.2019.02.006.

Carlos Florensa, Jonas Degrave, Nicolas Heess, Jost Tobias Springenberg, and Martin Riedmiller. Self-supervised learning of image embedding for continuous control. *arXiv preprint arXiv:1901.00943*, 2019.

P E Hart, N J Nilsson, and B Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2):100–107, July 1968. ISSN 2168-2887. doi: 10.1109/TSSC.1968.300136.

Kristian Hartikainen, Xinyang Geng, Tuomas Haarnoja, and Sergey Levine. Dynamical distance learning for unsupervised and semi-supervised skill discovery. *arXiv preprint arXiv:1907.08225*, 2019.

Tom Jurgenson, Edward Groshev, and Aviv Tamar. Sub-goal trees – a framework for goal-directed trajectory prediction and optimization. June 2019.

Michael Laskin, Scott Emmons, Ajay Jain, Thanard Kurutach, Pieter Abbeel, and Deepak Pathak. Sparse graphical memory for robust planning. arXiv:2003.06417.

Nikolay Savinov, Alexey Dosovitskiy, and Vladlen Koltun. Semi-parametric topological memory for navigation. *arXiv preprint arXiv:1803.00653*, 2018.

E C Tolman. Cognitive maps in rats and men. *Psychol. Rev.*, 55(4):189–208, July 1948. ISSN 0033-295X. doi: 10.1037/h0061626.

Hado van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. September 2015. URL http://arxiv.org/abs/1509.06461.

Ziyu Wang, Tom Schaul, Matteo Hessel, Hado van Hasselt, Marc Lanctot, and Nando de Freitas. Dueling network architectures for deep reinforcement learning. November 2015.

John Broadus Waston. The kerplunk experiment. 1914.

Ge Yang, Amy Zhang, Ari S. Morcos, Joelle Pineau, Pieter Abbeel, and Roberto Calandra. Plan2vec: Unsupervised representation learning by latent plans. In *Proceedings of The 2nd Annual Conference on Learning for Dynamics and Control*, volume 120 of *Proceedings of Machine Learning Research*, pages 1–12, 2020. arXiv:2005.03648.