Learning from Graphical Replay

Ge Yang, Amy Zhang[†], Ari Morcos[†], Joelle Pineau[†‡] Pieter Abbeel[§] and Roberto Calandra[†]
[†]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 [2], which relies on effective gathering of relevant new data, and a strong inductive bias over learning. We humans contextualize both sampling and learning by constructing a mental map [10]. In contrast, model-free reinforcement learning methods often take an unstructured approach to both problems. They sample by injecting noise and cache past experience 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 tackle these two key problems – how to efficiently sample data, and how to continuously learn once the data comes in by focusing on *episodic exploration* that is directed towards a specific but dynamically specified goal, under the guidance of a structured cognitive map that offers a graphical organization of past experience. Our approach is inspired by a promising model-based approach [9] that combines a low-level parametric policy with long-horizon planning on a semi-parametric topological map (SPTM). The significance of this approach is two-fold: 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 for learning reactive policies for complex and long-horizon tasks on a semi-parametric graphical world model. 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. Finally, central to our approach is our ability to apply focused model improvement driven by path planning on the graph, which exposes discrepancies between the model and the true environment dynamics.

II. Technical Background

Learning generalized universal value estimates directly with 1-step Q-learning from pixel input is known to be difficult [3, 6, 5, 14]. Hindsight experience replay (HER [1]) takes advantage of the structure in achieving sub-goals [7], to populate the replay buffer with relabeled transition tuples $\langle o_t, a_t, R(o_t, a_t, \hat{o}), o_{t+1} \rangle$ containing positive rewards. This is a covert way to reuse sampled dynamics to supervise the value function with ground-truth reward. Despite its success, HER is limited to linearly relabeling goals within each trajectory. In this work we take a model-based approach to *extend linear relabeling to a graphical relabeling scheme on a learned graphical world 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.

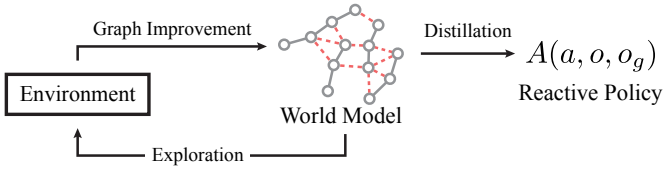

Fig. 1: Learning from Graphical Replay (**LfGR**)

A. 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)

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 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 step-size hyperparameter d_o via $p(\varepsilon \in \mathcal{E}) = \exp(-\beta(d_0 - d(o_i, o_j))^2)$, 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 $d(o, o') \rightarrow \mathbb{R}^+$ by constructing a noise-contrastive objective using positive transition tuples

sampled from the joint distribution $p(o_t, a_t, o_{t+1})$ and negatives from the noisy product of the marginals $p(o_t)p(o')$.

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

We introduce a novel negative hinge loss for 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 $2\times$ 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 $d(o, o') = \|z - z'\|_p$. Learning is warm started 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 maintaining close to uniform sample distribution over the state space by only inserting a new vertex when it is at least d_o away from existing nodes [8].

B. Goal-directed Exploration And Causal Graph Improvement

The perception distance d offers a crude approximation of the true environment dynamics that includes friction, physical contact and hysteresis. When making 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 [11] is a common problem with bootstrapped Q-learning. Yet, in classical AI planning literature, an optimistic value heuristic is *required* for A^* to find the *shortest-path* [4]. This duality between the *admissibility* criteria in search and optimism in a failing behavior policy during exploration is often overlooked by contemporary research. We propose to use planning-based methods' susceptibility to modeling mistakes to drive effective exploration under an episodic, and go-directed setting (Fig.1).

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 omit in this extended abstract. We additionally search for incoming edges into states that are "stuck", within a perception distance d_1 for more efficient edge removal.

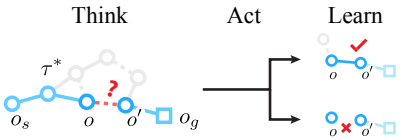

Fig. 2: Model-driven Exploration

C. Universal Value Prediction Network (UVPN)

Following dueling networks [12], we decompose the Q-function into learning an optimal state-value $V(s, g)$ and an advantage $A(a, s, g)$ via $Q(a, s, g) = A(a, s, g) + V(s, g)$ 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 [14]. We combine this planning based long-horizon target with the local distance target from Eq.1 into the following objective $\mathcal{L}_V = \mathcal{L}_d + \mathcal{L}_{\tau^*}$ where $\mathcal{L}_{\tau^*} = \left|V^*(o_s, o_g) - \sum_{\tau^*(s,g)} d(o_t, o_{t+1})\right|$. To learn the advantage function, we can construct the following target $A(a_t, o_t, o_g) = [V(o_t, o_{t+1}) + V(o_{t+1}, o_g))] - V(o_t, o_g)$. 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_ϕ
1: Sample transition $\langle o_t, a_t, o_{t+1}\rangle$ from buffer, Sample o_g from $\mathcal{G}$.
2: find optimal path $\tau^* = S(G, o_t, o_g)$, where $\tau^* = \{o_t, o_1, o_2, \cdots o_g\}$
3: **for** each epoch **do**
4: minimize $\mathcal{L}_V$

D. Goal Relabeled Expert Distillation (GRED)

With GRED, we construct a target distribution $p(\tau^*)$ of long-horizon expert trajectories by relabeling optimal plans made on the graph using a local inverse model l_{Inv}. We train l_{Inv} with the standard maximum likelihood objective: $\mathcal{L}_{\text{Inv}} = -\log l_{\text{Inv},\theta}(a_t|o, o')$.

Algorithm 2 Goal Relabeled Expert Distillation

Require: Graph $\mathcal{G}$, search procedure S
Require: Inverse model l_{Inv}
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 $D_{\text{KL}}\|l(a|o_s, o_g),\ l^\top_{\text{Inv}}(a|\cdot)\| + \mathcal{L}_{\text{Inv}}$

$(\cdot)^\top$ blocks gradient propagation, $(\cdot)$ indicate relabled observations

We experiment with a *forward* and a *backward* relabeling scheme. $l^\top_{\text{Inv}}$ in Alg.2 indicates that we do not propagate gradient through the local model. The **forward relabel** scheme only generates one action proposal $l^\top_{\text{Inv}}(o_s, o_1)$ per plan, but relabels the goal entry in $l(o_s, \mathbf{o'_g})$ with o'_g that are sub-sampled every k steps from 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 cost in computing action proposals is offset by the ability to batch and much more diverse labels, yielding better performance.

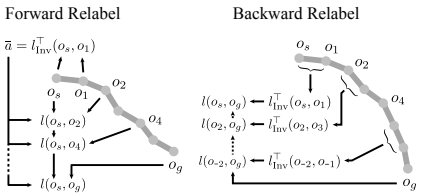

Fig. 3: Relabel Schemes for GRED.

IV. Results

We evaluate *learning from graphical replay* in this section to show (a) the episodic graph enables focused adaptation behavior

in non-stationary environments, (b) local changes to the graph can affect global changes in policy without additional samples from the environment, and (c) we can learn long-horizon reactive policy by supervised learning from the graph. Videos can be found at https://geyang.github.io/graphical-replay.

Episodic Robustness We re-enact the "kerplunk" experiment [13] by altering a C-shaped maze that the agent has already mastered through past experience (Fig.4a). We insert a vertical separation (colored in red (Fig.4b) 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.

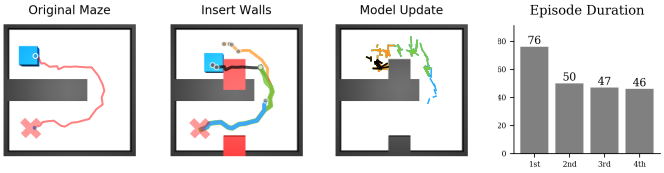

Fig. 4: Adapting to changes in the environment. **(a)**: Original plan made by the agent in Maze. **(b)**: we insert a vertical separation, shown as two red blocks. Colored lines shows the forward plan at 4 separate steps in the same episode. **(c)**: colored segments showing the edges removed at each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

The agent acts according to plan, but it gets stuck due to the new wall. This prompts the agent to update its world model by removing similar edges from the graph (Fig.4c), and then re-plans. We plot several steps during this adaptation process in color **black**, **orange**, **green**, and **blue**. This is a form of **episodic learning**, the progress of which can be see in Fig.4d, as the number of steps the agent takes to complete the task decreases over more trials.

Sample-efficient Adaptation A major benefit of our model-base approach, is that local changes to the model can propagate quickly to affect the policy in a global manner. In this experiment we compare graphical replay against standard, linear replay in HER by looking at the learning dynamics under non-stationarity. We use a fixed start and goal position for evaluation purposes. The agent enters from the top trying to reach a goal at the bottom. As we insert a block (colored in **red**), the agent removes those edges that it deems infeasible (**purple** segments, Fig.5b), which is immediately picked up by the UVPN value distillation loop. Visualization of the change in value estimate in Fig.5c shows that despite the change is local, it affects the value estimate of the entire domain. Color indicates the $\Delta \max_{a\sim\mathcal{A}} Q(a, o, o_g)$ between before and after inserting the block. The entire sidewalk area incurs a decrease in value between $(-2, -4)$. We further compare qualitatively against linear goal-relabeling (HER) which never manages to recover, whereas our method succeeds 100% after a single episode of exploration, despite that going around the wall is more difficult than the original route. The UVPN policy is retrained from random initial weights for each data point.

Learning Long-horizon Reactive Policy In Fig.6a we compare the success rate with our method against a local policy,

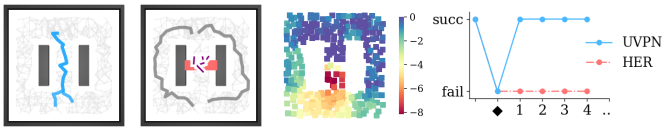

Fig. 5: UVPN and GRED directly distill from the graph, which allows local changes to quickly propagate to affect global policy change without requiring additional samples from the environment. **(a, b)** Path before and after blockage. Agent starts from the top middle to reach two goals separated by 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.

both UVPN and GRED are able to reach farther. Our ablation test indicates that graph improvement by removing infeasible edges is critical for the success of the reactive policy. Fig.6b shows edges removed from the wall in the middle of the C Maze environment.

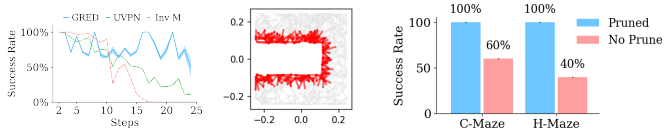

Fig. 6: **(a)** Reactive performance at different planning horizon, with a *partially pruned graph*. Inverse model baseline drops off quickly as the goal moves farther away, whereas UVPN and GRED using a partially pruned graph maintain a slower drop-off. All traces averaged over three seeds and 200 task 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.

Reducing Spurious Motion Plans made on the graph are spurious due to the discrete nature of the graph. In Fig7 we show that behavior traces made by the hierarchical MPC consistently makes sharp turns, due to constant re-localization and replan, reflected by a tight grouping and high average acceleration in (b). LfGR agents on the other hand generate smoother trajectories. The learned reactive control(π) only makes 1/3 of the turns, with an average acceleration close to zero.

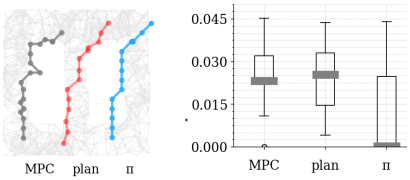

Fig. 7: Quantitative comparison between high-level plan, path executed via MPC, versus the reactive policy. (b) distribution of the spuriousness for trajectories in (a). Offset to the side added in (a) for clarity.

References

[1] 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.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

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

[7] Leslie P Kaelbling. Learning to achieve goals. *IJCAI*, 1993. ISSN 1045-0823.

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

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

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

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

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

[13] John Broadus Waston. The kerplunk experiment. 1914.

[14] 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.