Environment Agnostic Goal-Conditioning, A Study of Reward-Free Autonomous Learning

Hampus Åström, Elin Anna Topp, Jacek Malec

Keywords: Reinforcement Learning, Goal-Conditioning, Unknown Environment, Reward-Free

Summary

In this paper we study how transforming regular reinforcement learning environments into goal-conditioned environments can let agents learn to solve tasks autonomously and reward-free. We show that an agent can learn to solve tasks by selecting its own goals in an environment-agnostic way, at training times comparable to externally guided reinforcement learning. Our method is independent of the underlying off-policy learning algorithm. Since our method is environment-agnostic, the agent does not value any goals higher than others, leading to instability in performance for individual goals. However, in our experiments, we show that the average goal success rate improves and stabilizes. An agent trained with this method can be instructed to seek any observations made in the environment, enabling generic training of agents prior to specific use cases.

Contribution(s)

1. We apply goal-conditioning on non-goal environments, and show that ignoring external rewards and training guided by goal selection can yield comparable results to an externally guided baseline.
   Context: Prior work introduced goal-conditioning and agents that self-select goals (Andrychowicz et al., 2017; Colas et al., 2022).
2. We present a study on the success rates of multiple goals, during training with autonomous goal conditioning, in three different environments. We show that, while the success rate for individual goals varies significantly during training, the average goal success rate gradually improves and stabilizes.
   Context: In previous work, Florensa et al. (2018) study goal coverage, as an overall measure of average goal success rate, without explicitly measuring the performance of individual goals over time.
3. We provide an extension to Stable-Baselines3 (Raffin et al., 2021), which enables applying goal conditioning to non-goal environments via a wrapper, with modular goal evaluation and selection. [a]
   Context: Our code contribution relies on and merely extends the preexisting repository Stable-Baselines3 (Raffin et al., 2021).

[a] https://github.com/HampusAstrom/goal-exploration

Environment Agnostic Goal-Conditioning, A Study of Reward-Free Autonomous Learning

Hampus Åström[1], **Elin Anna Topp**[1], **Jacek Malec**[1]
{hampus.astrom,elin_anna.topp,jacek.malec}@cs.lth.se

[1]**Dept. of Computer Science, Lund University, Lund, Sweden**

Abstract

In this paper we study how transforming regular reinforcement learning environments into goal-conditioned environments can let agents learn to solve tasks autonomously and reward-free. We show that an agent can learn to solve tasks by selecting its own goals in an environment-agnostic way, at training times comparable to externally guided reinforcement learning. Our method is independent of the underlying off-policy learning algorithm. Since our method is environment-agnostic, the agent does not value any goals higher than others, leading to instability in performance for individual goals. However, in our experiments, we show that the average goal success rate improves and stabilizes. An agent trained with this method can be instructed to seek any observations made in the environment, enabling generic training of agents prior to specific use cases.

Introduction

Reinforcement learning (RL) is a good step towards the idea of machines that learn autonomously. RL only needs to provide the agent with an environment with which to interact, not an expensive trove of data. However, defining a reward function is problematic, as it often needs to be hand-made by an engineer to guide the agent to optimal behavior. Ideally, an autonomous agent would learn meaningful behavior without the external guidance of a problem-specific reward function. We propose that by wrapping environments in a goal-conditioning framework, and training without any external rewards to reach points in the environment, we can create agents that learn broad capabilities that can be accessed for specific tasks.

Contributions

- We apply goal-conditioning on non-goal environments, and show that ignoring external rewards and training guided by goal selection can yield results comparable to an externally guided baseline. This expands on prior work that introduced goal-conditioning (Andrychowicz et al., 2017) and agents that self-select goals (Colas et al., 2022).
- We present a study on the success rates of multiple goals, during training with autonomous goal conditioning, in three different environments. We show that, while the success rate for individual goals varies significantly during training, the average goal success rate gradually improves and stabilizes. In previous work, Florensa et al. (2018) study goal coverage, as a measure of average goal success rate, but without explicitly measuring the performance of individual goals over time.
- We provide an extension to Stable-Baselines3 (Raffin et al., 2021), which enables applying goal conditioning to non-goal environments via a wrapper, with modular goal evaluation and selection. [1]

[1]A link to a Github repository will be here when the submission is no longer anonymous.

Related Works

In regular reinforcement learning, a predefined reward function describes what behavior is desirable. This most often means that there is a single optimal policy. Even when well-defined external rewards exist, they can be sparse and deceptive. During learning, an agent often cannot rely solely on these signals to guide its actions. Pathak et al. (2017) suggest solving this with *Intrinsic Curiosity*, a method that treats errors in a transition model as intrinsic rewards to guide exploration towards interesting states. Although efficient, this relies on distorting the agent's perception and can suffer from problems with detachment and derailment (Ecoffet et al., 2021). Exploration can also be done with the help of goals (Florensa et al., 2018; Ecoffet et al., 2021; Colas et al., 2022).

Goal-conditioned reinforcement learning (Andrychowicz et al., 2017) proposes that we describe to the agent what we want it to achieve in each episode. This can be done by adding a goal description to the observations and providing a goal selection method and a goal evaluation scheme. This is particularly valuable if one wants to learn different or parameterized tasks, as well as in autonomous learning (Colas et al., 2022; 2019; Florensa et al., 2018; Liu et al., 2022). If goal-conditioning can be applied in an environment-agnostic way, agents can learn autonomously and reward-free, in unknown environments.

Method

This paper explores whether and how goal conditioning with self-selection of goals can be formulated in a generic way. This approach is evaluated by training agents with a goal wrapper in an environment-agnostic, reward-free formulation and comparing them to regular RL methods that are aware of external goals. Agents trained in such a way can be supplied with tasks via the goal formulation, post-training, enabling flexible agents that can do various tasks in the environment they trained on.

We have created a wrapper in the Stable-Baselines3 framework (Raffin et al., 2021), to enable goal-conditioning for non-goal RL environments. It takes observations from the underlying environment, together with goals, and uses them to evaluate when goals have been achieved.

Goal success evaluation is modular; in our experiments we have used normalized distance to determine goal success in continuous environments and exact matching in discrete environments. We terminate when the agent reaches a goal.

Goal selection is also modular, and we present three different example methods that are used to select goals for the agent during training: uniform sampling, novelty selection, and intermediate difficulty selection. Goals should prioritize exploration, while making sure that the agent can improve its capacity to reach found goals and avoid forgetting previous capabilities. In this way, goal selection selectively collects training data. In future work, our aim is to explore supplementing goal selection with smarter filling and sampling from the replay buffer to get more efficient and stable goal learning.

We constrain our study to single observation goals, rather than goal trajectories, as goal conditioning already expands the task considerably. With external rewards, there is often a single optimal strategy, but with goal-conditioning, each goal might need a unique strategy. Hindsight Experience Replay (Andrychowicz et al., 2017) reduces issues with the expanding scope, by enabling learning from every episode, by the principle that each episode shows a way to reach the observations in that episode. We use off-policy learning, and since we only present studies on environments with discrete action spaces here, we use Raffin et al. (2021)'s implementation of Deep Q-Networks (DQN) (Mnih et al., 2013) (though we have confirmed that our method works with Soft Actor-Critic (Haarnoja et al., 2018) as well).

Goal selection strategies

In the general case, a goal selection strategy might want to take into account the current state or observation, especially when starting conditions change drastically or when goals are re-selected during an episode. There are only minor variations in start conditions of the environments of our trials here, and we terminate whenever a goal is achieved, so in this work that factor is ignored when selecting goals.

We use three main methods to select goals: Uniformly random, novelty seeking, and intermediate success rate selection (the latter inspired by intermediate difficulty goal selection (Florensa et al., 2018). With all methods, we add some uniform random selection to avoid getting stuck in a local goal subset.

Uniform random

The simplest way to select goals is to randomly sample over the observation space uniformly. In environments where only a fraction of the observation space are viable, reachable observations, this method risks selecting a large fraction of nonviable goals.

Novelty selection

Our novelty selection method selects goals by valuing less visited areas higher. In environments with discrete observations, this is simply based on counting visits to each observation, while for continuous environments, we use a grid to collect visitation statistics. The relative novelty weight w_N for selecting a cell is computed by

$$w_N(\text{cell}) = \frac{1}{(p_v(\text{cell}) + \epsilon)^n} \tag{1}$$

where $p_v(\text{cell})$ is the number of steps in the cell in relation to total steps taken, ϵ is a small positive number (0.01), and n is a decay factor guiding how heavily it should prioritize cells with low visitations counts. Grid methods scale poorly with observation size, but the environments in this study are simple enough. Once a grid cell is selected, a goal point is selected by sampling uniformly within it.

Intermediate success rate selection

With the intermediate success rate goal selection method, a grid of cells is mapped onto the observation space in the same way as for novelty selection. Instead of visitations, targeted and successful goals are tracked in each cell. The rationale here is that we should learn some goals decently well before moving on to others. The relative weight w_S for selecting a cell is defined similarly to novelty selection, with

$$w_S(\text{cell}) = \frac{1}{(|p_s(\text{cell}) - p_{st}| + \epsilon)^n} \tag{2}$$

where $p_s(\text{cell})$ is the success rate in each cell, p_{st} is a target success rate hyperparameter, ϵ is a small positive number (0.01), and n is a hyperparameter that governs the degree to which proximity to the target success rate should be emphasized. This is complemented with randomly selecting goals among visited but not targeted cells, in proportion to how many cells have been visited but not targeted, as well as some uniform random selection over all observations.

Environments

We have applied our method to three environments: Cliff Walking, Frozen Lake and Pathological Mountain Car, with trials on more environments being ongoing work. The former two are part of the OpenAI Gym framework (Brockman et al., 2016), and the latter is an adaptation of Mountain Car from the same framework. All environments currently examined are fully observable with discrete actions.

The Cliff Walker environment is the simplest, it is discrete and deterministic, but with a few unreachable states, providing a simple environment and testing how much of an issue unreachable states are. Frozen Lake has stochastic transitions, with different maximum achievable success rates for each state; this can be deceptive when success rates are used to guide goal selection. The Pathological Mountain Car has continuous states and observations, exposing questions on how precise goal evaluation should be, if there are drawbacks with goals constrained to points, and how well our method can handle environments where actions are note easily reversible.

Cliff Walking

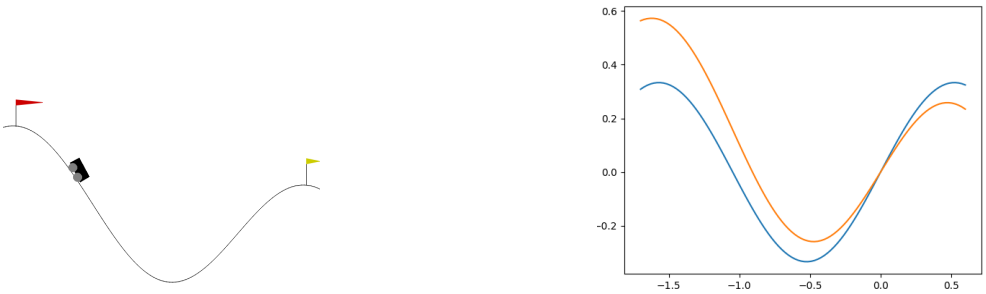
The Cliff Walking environment (Brockman et al., 2016) is small, with a discrete grid of observations (represented by a single number each), and most observations (falling of the cliff moves the agent to the starting state). Transitions are deterministic and can in most cases easily be reversed by performing the inverse action, with termination only on the environment goal. The external rewards are -1 for each step and -100 for falling off the cliff. We truncate episodes after 300 steps, to allow for hindsight experience replay.

Frozen lake

The Frozen Lake environment (Brockman et al., 2016) is similar to Cliff Walking, a discrete grid with locations to avoid. However, with default parameters, it has stochastic transitions. It has a $1/3$ chance of going in the intended direction and $1/3$ for each of the perpendicular ones, making its optimal policy less trivial. In this case pitfalls end the episode, instead of resetting to the start like in the Cliff Walker environment. It truncates after 100 steps.

Pathological Mountain Car

We provide an adaption of Brockman et al. (2016)'s Mountain Car, called Pathological Mountain Car, inspired by an environment of the same name introduced by Chakraborty et al. (2023). It differs from Open AI's Mountain Car by applying a small linear shift in height, making one hill harder to reach than the other, and placing a terminating state there, visualized in Figure 1. The rewards are 500 for reaching the summit of the tall hill, 10 for the low hill. Unlike the regular Mountain Car environment, there is no penalty for each step. It truncates after 300 steps. The Pathological Mountain car has the same continuous states and observations (horizontal position and velocity), and discrete actions, as the original. In order to reach the high-value goal, an agent would need to get quite near the low-value goal, which can make greedy agents miss the greater payout. In all mountain car environments, unlike the grid environments above, there is a lot of momentum in the system, so actions are not easily reversible.



(a) (b)

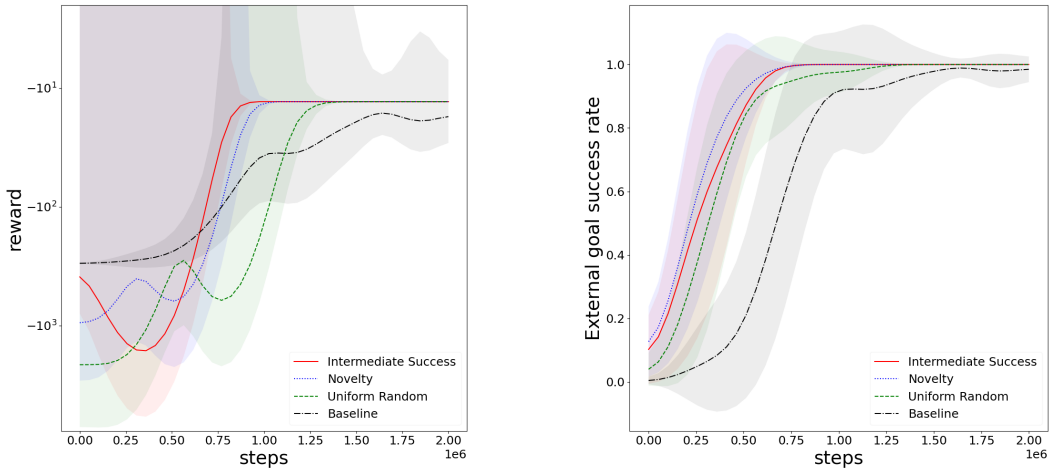
Figure 1: The Pathological Mountain Car environment, visualized in 1a. An adaption of Brockman et al. (2016)'s Mountain Car, with an additional goal, and a shift making one hill steeper (with higher external reward) and the other slightly flatter. 1b shows the difference in inclination between our implementation (orange) and the original Mountain Car (blue).

Experiments and Results

In the first section, our method is contrasted with a baseline RL algorithm using external rewards, and in the second section, we investigate the performance on different goals over time. Single standard deviations are included in all plots, and gaussian smoothing is applied for legibility.

Comparison to external reward aware baseline

Compared to non-goal-conditioned reinforcement learning with DQN, our method reaches optimum faster on the Cliff Walker problem, as shown in Figure 2b. Since our method does not see the external reward signal, it does not avoid cliffs as well in early training, seen in Figure 2a, but still stabilizes to the optimal solution slightly faster than the baseline with all goal selection methods.



(a) (b)

Figure 2: Cliff Walker environment, evaluation reward with symlog scale 2a, as a function of training steps, and optimal behavior rate, 2b, with 8 experiments for each method. **Takeaway:** Our solution reaches the optimal policy quicker.

For the Frozen Lake environment, our method reaches a stable level of similar reward 2-3 times slower than the baseline, see Figure 3, but neither reaches the optimal average reward (of about 0.7) in the allotted training time. Intermediate difficulty selection performs worse on Frozen Lake. Since the environment is stochastic, the optimal success rate for several of the goals is less than 100%, and sometimes is similar to the target success rate. This leads to a goal selection bias, focusing training on such goals even when the agent already has the optimal policy for them.

The Pathological Mountain Car environment is more difficult to learn than the two earlier environments. As Figure 4 shows, neither the baseline nor our method manage to reach the harder goal consistently. Our method reaches it intermittently, while the baseline reaches it in two out of eight experiments, and in one of those cases then learns to reach it more consistently. Experiments with a longer training time could determine which method is most consistent in learning to reach the hard goal.

This task exposes an issue with our method; our goals are observational points, specifying both a target position and a target velocity. The real goal here is to get the (horizontal) position to be less than -1.6, independent of velocity, and thus our method ignores many ways to complete the task and could even terminate prematurely if the validation goals are not set well. A dedicated solution could, of course, ignore or reformat the goal format to account for the particular environment, but we want to provide an environment-agnostic method. In ongoing work, we are expanding the goal formulation to take goal ranges. This would both allow evaluation with goals like the ones in this

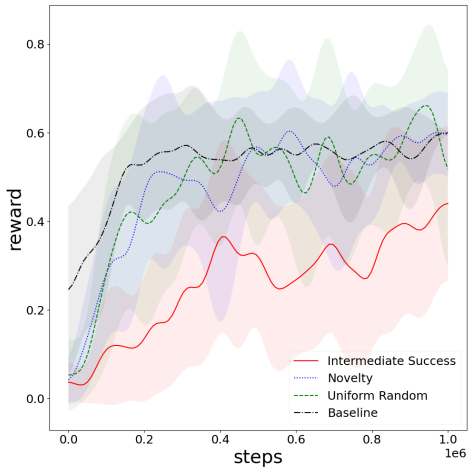
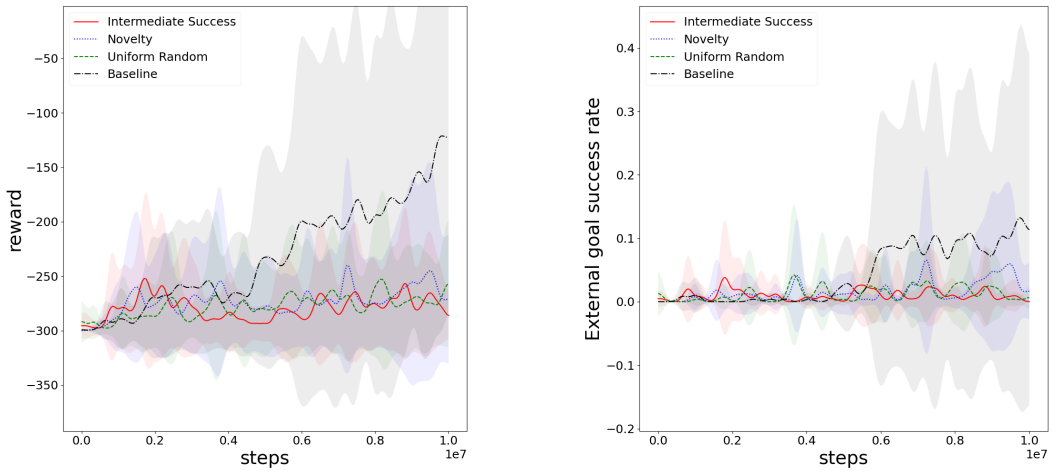


Figure 3: Evaluation reward for Frozen Lake, as a function of training steps, with 4 experiments for each method. **Takeaway:** Our method gets comparable results, reward-free, with all but intermediate difficulty selection, but our methods and the baseline are worse than an oracle (with ≈ 0.7 average reward).

environment, and allow the goal selector to tune the goal difficulty by describing goals of varying scope, without leaving the environment-agnostic formulation.



(a) (b)

Figure 4: Evaluation reward, 4a, and rate of success for reaching the hard goal, 4b, with 8 experiments for each method on Pathological Mountain Car, as a function of training steps. When evaluating goal methods, the hard hill is given as target goal. **Takeaway:** Our method reaches the hard goal faster than the baseline, but does not retain the ability to reach it consistently.

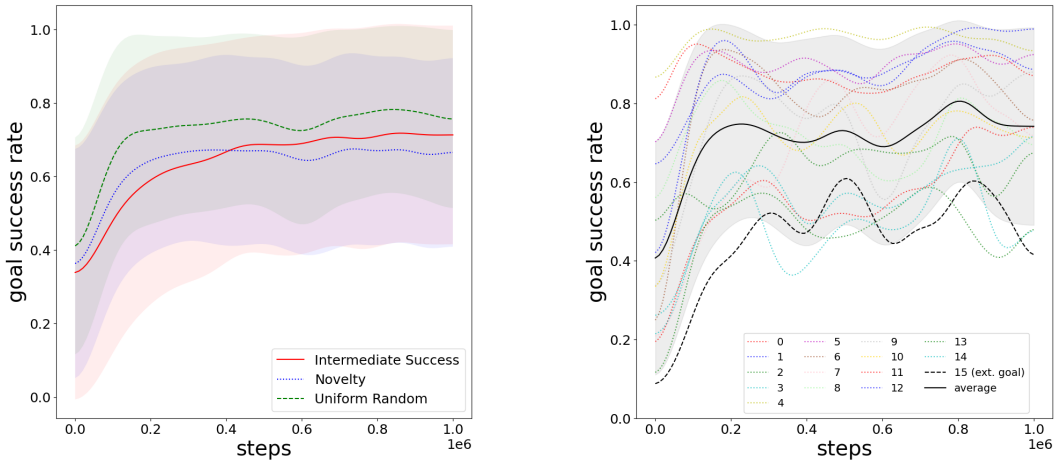
Goal success rates

Learning to reach any goal is a much broader task than external reward reinforcement. There is at least one optimal policy for each goal, instead of a global optimum for the entire environment. Agents trained with our method do not know about external rewards; instead, they are taught to reach all goals presented to them (as targeted goals, or hindsight goals during experience replay).

To evaluate their performance, one should therefore track performance on a wide set of goals in the environment.

In the Cliff Walker environment, all goal selection methods converge quickly. The optimal policy is found for all goals at nearly the same time, and the average success rate for all goals reaches the expected $38/48 \approx 80\%$. It cannot reach 100% since the ten cliff locations reset to the starting position and cannot be reached.

The Frozen Lake environment is mainly characterized by its stochastic transitions, leading to extra uncertainty in both learning and evaluation. Despite this, the average goal performance is fairly stable, as shown in Figure 5a. When inspecting the training of a single agent, Figure 5b, one observes that performance on individual goals is unstable. If this is due to forgetting, the stochastic nature of the environment, or something else, is yet to be determined.



(a) (b)

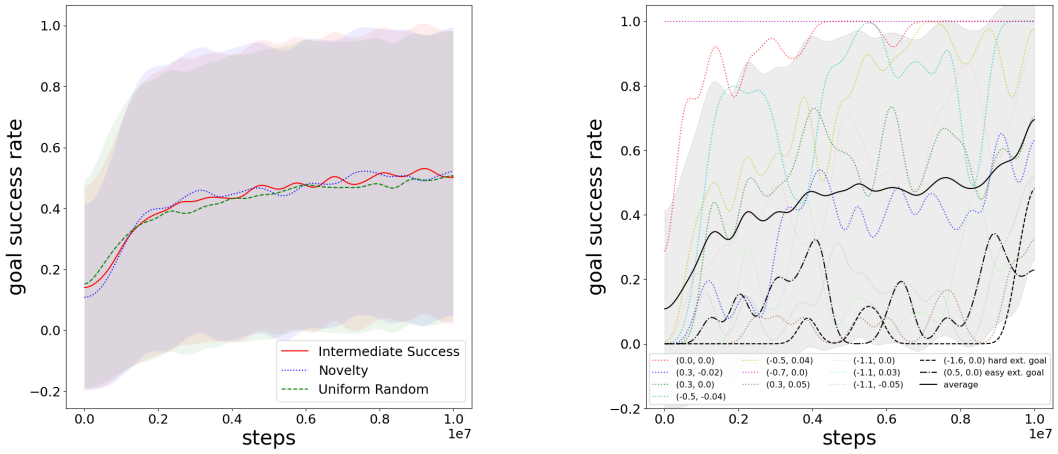
Figure 5: The average goal success rate for our methods on Frozen Lake, 5a, and an example of how success rates vary for goals when training an agent, 5b. In 5b goal 15 has the external reward (though our agent is unaware of that). **Takeaway:** The average stabilizes quickly, but individual goal performance fluctuates wildly, though this could in part be due to the stochasticity of the environment.

When studying goal performance in the Pathological Mountain Car environment, measurements cannot capture the goal performance on all goals, as it is an infinite set. A set of spread-out goals is selected as a sample of the overall goal success rate. Evaluations of this set during training, Figures 6a, 6b, show that while the average goal success rate is similar for most goal selection methods, it is very unstable for individual goals. This is similar to the results for the Frozen Lake environment, but here the environment is deterministic, suggesting that forgetting or other training instability might be the cause. It should be noted that the average goal success rate is slowly growing, indicating that longer training time could be enough to achieve a consistent policy for all goals.

Conclusion

In this paper, we have shown how goal conditioning can be used to learn reward-free in an environment-agnostic way, and we provide code to apply this method to non-goal environments. There are many open questions, but so far we can conclude that by autonomously selecting goals, our method can solve different tasks with accuracy and training times comparable to a non-goal conditioned externally reward guided baseline.

In ongoing and future work, we will explore how this method performs in more varied environments. We are interested in whether model-based methods can improve training efficiency by decoupling



(a) (b)

Figure 6: The average goal success rate for our methods when trained on the Pathological Mountain Car environment, 6a, and an example of how success rates vary for goals when training an agent, 6b. In 6b goal are described as $(x, \dot{x})$ where x is horizontal position, and $\dot{x}$ is horizontal velocity, with $(-1.6, 0.0)$ and $(0.5, 0.0)$ corresponding to reaching the hard goal and easy goals respectively (while becoming stationary). **Takeaway:** The success rates for individual goals are unstable, even though the environment is deterministic, but the average success rate for all measured goals increases steadily.

the learning of a transition model from the goal-based evaluation of states, observations, or actions. We also want to study the impact of more flexible goal formulation. A study on goal selection, its interaction with hindsight experience replay, and other ways to gather and use training data is ongoing. For complex and partially observed environments, other means of representing goals might be needed, and we are interested in exploring if goals can be selected within an embedding space, or if such changing representations are unsuitable.

Acknowledgments

Thank you Samuel Blad, for helping me find a bug that had evaded me for months.

This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation.

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. *Advances in neural information processing systems*, 30, 2017.

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

Souradip Chakraborty, Amrit Singh Bedi, Kasun Weerakoon, Prithvi Poddar, Alec Koppel, Pratap Tokekar, and Dinesh Manocha. Dealing with Sparse Rewards in Continuous Control Robotics via Heavy-Tailed Policy Optimization. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 989–995, May 2023.

Cédric Colas, Pierre Fournier, Mohamed Chetouani, Olivier Sigaud, and Pierre-Yves Oudeyer. Curious: intrinsically motivated modular multi-goal reinforcement learning. In *International conference on machine learning*, pp. 1331–1340. PMLR, 2019.

Cédric Colas, Tristan Karch, Olivier Sigaud, and Pierre-Yves Oudeyer. Autotelic Agents with Intrinsically Motivated Goal-Conditioned Reinforcement Learning: A Short Survey. *Journal of Artificial Intelligence Research*, 74:1159–1199, July 2022.

Adrien Ecoffet, Joost Huizinga, Joel Lehman, Kenneth O. Stanley, and Jeff Clune. First return, then explore. *Nature*, 590(7847):580–586, February 2021.

Carlos Florensa, David Held, Xinyang Geng, and Pieter Abbeel. Automatic goal generation for reinforcement learning agents. In *International conference on machine learning*, pp. 1515–1528. PMLR, 2018.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pp. 1861–1870. PMLR, 2018.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

Minghuan Liu, Menghui Zhu, and Weinan Zhang. Goal-Conditioned Reinforcement Learning: Problems and Solutions. In *Thirty-First International Joint Conference on Artificial Intelligence*, volume 6, pp. 5502–5511, July 2022.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

Deepak Pathak, Pulkit Agrawal, Alexei A. Efros, and Trevor Darrell. Curiosity-Driven Exploration by Self-Supervised Prediction. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 488–489, Honolulu, HI, USA, July 2017. IEEE.

Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann. Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 22(268):1–8, 2021. URL http://jmlr.org/papers/v22/20-1364.html.

Supplementary Materials

The following content was not necessarily subject to peer review.

Hyperparameters

For non-listed hyperparameters, Stable-Baselines3 defaults are used, but if we missed listing something, don't hesitate to reach out.

Goal evaluation

In the presented evaluations, goals are considered successful if a normalized distance vector between the observation and the goal is less than 0.1.

Goal selection

For both novelty and intermediate success rate selection, $\epsilon = 0.01$. For data shown in this paper $n = 2$, but $n = 1$ produces similar results. The grid used to collect statistics for goal selection has $10\,000$ cells for Pathological Mountain Car, and matches obs space exactly for discrete environments.

With intermediate success rate selection, 0.9 is the target success rate for CliffWalker, 0.75 for FrozenLake and Pathological Mountain Car.

In all these cases, 10% of the time, a random goal is selected from the observation space.

Network

Two Q-networks are used, one simple with 3 fully connected layers of 256 nodes each, and one resnet inspired network with 4 sequential residual nodes. Each residual node consists of 4 fully connected layers with 256 nodes, with the output of the final one being additive with the input of the residual node, similar to the original ResNet by He et al. (2016). The simple network is used for the Frozen Lake environment and the ResNet inspired network for Cliff Walker and Pathological Mountain Car. We used Stable Baseline's implementation of Hindsight Experience Replay, adapted to handle termination when reaching goals.

Agents are trained with 0.001 learning rate, gamma 0.95, batch size 512, and train frequency 512. HER replay buffer uses "future" goal reselection, with 4 hindsight goals for each use of unaltered experience. The replay buffer is $1\,000\,000$ steps long.

Cliffwalker was trained for $2\,000\,000$ steps, FrozenLake $1\,000\,000$ steps and Pathological mountain car $10\,000\,000$ steps.