A Node-NMRDP function P

The one-to-many relationship function $P : \Omega \rightarrow \mathcal{M}$, given a node $\omega \in \Omega$ in the DFA, returns the set of tasks that can reach the node ω from the start node ω_0. For example, consider the DFA representation in Fig. 1. A query to the function $P(\omega_1)$ will return the set of NMRDPs that produce trajectories to reach the state ω_1. Essentially, all these trajectories will at some time collect a *tree* from the environment, or will already have a *tree* in the inital state representation. To attain the set of NMRDPs that can reach a particular node in the automaton, we calculate the lowest possible inital state OOMDP configuration whose NMRDP can produce trajectories to reach the node. Once we have that configuration (requirement of 1 *tree* to reach ω_1), we vary the initial OOMDP state parameters to give us different initial environment configurations that have the same goal. All these environment configurations correspond to different NMRDPs. Doing this for all the nodes in the DFA gives us the set of NMRDPs corresponding to each node in the DFA.

B Grid-world and Crafter-Turtlebot Domains

In the gridworld and the Crafter-Turtlebot domains, the set of actions that the agent can take are: move forward, rotate left, rotate right, break and craft. Move forward makes the agent go forward by 1 cell in the gridworld domain and by $0.1m$ in the Crafter-Turtlebot domain. The rotate actions cause the agent to rotate $\pi/2$ in the gridworld domain and by $\pi/8$ in the Crafter-Turtlebot domain. The break action causes the object in front of the agent to disappear from the environment and appear in the inventory. It does not have any effect if there is no object in front of the agent (i.e. it is empty). Similarly, the agent can craft a pogo-stick only if it has 2 trees and 1 rock in the inventory and is facing the crafting-table.

An example of the sequence-based curriculum generated by the *AGCL* procedure for the gridworld domain is shown in Fig 2. The goal of the agent in the first source task (Task-1) is to navigate to the *tree* present in the environment and break it to collect it in the inventory. Once the agent meets the *stopping criteria* required (average success rate on past 50 episodes greater than 0.7), the agent uses the learned value function to learn the next source task (Task-2) in the curriculum. The goal of the agent in the second source task is to navigate to the objects present in the environment and break it to collect them in the agent's inventory. Similarly, when the agent meets the *stopping criteria*, the value function is transferred to learn Task-3, whose goal is to break 2 *trees* and 1 *rock* and collect them in the inventory.

Once the agent has learned all the source tasks in the curriculum, the agent attempts learning the final target task, as shown in the figure. The goal of the agent in the final task is to collect 2 *trees* and 1 *rock* followed by crafting a *pogo-stick* at the *crafting table*.

To generate the source tasks in the high-fidelity environment, each of the LF source tasks is mapped to generate the HF source tasks. The HF source tasks are learned iteratively, culminating in the HF target task. The sub-goal can be either to break the required number of items or to craft a stone-axe.

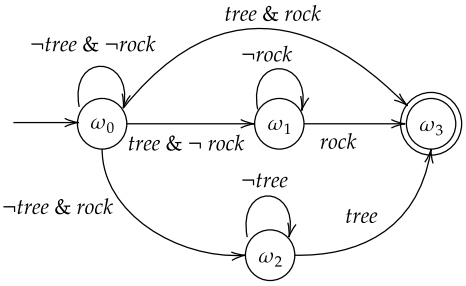

Figure 1: DFA reprsentation for the LTL: $\mathbf{F}(tree) \wedge \mathbf{F}(rock)$

C Baselines Implementation

As mentioned in the paper, we adapted four baseline approaches from the literature: Assymetric Self-Play (Sukhbaatar et al. 2018), Teacher-Student Curriculum learning (Matiisen et al. 2020), Q-learning for reward machines (Icarte et al. 2018) and Guiding Search via Reward Shaping (Camacho et al. 2018)

C.1 Assymetric Self-Play

We adapted the assymetric self-play approach proposed by (Sukhbaatar et al. 2018) to our problem of crafting a pogo-stick. Alice, the goal-proposing policy, proposes a goal for Bob, whose goal is to reach the goal proposed by Alice. We used the approach in *Repeat* mode, where Bob repeats the task instead of reversing it, and tries to reach the goal proposed by Alice.

The maximum number of steps for Alice and Bob were set to 150, equal to the steps allowed for our *AGCL* approach. We tuned the hyperparameters through experimentation for our task, using a heuristic grid search. The parameters we used were: Self-Play percentage: 10%, Self-play reward scale γ: 0.01 and the self-play mode at *repeat*.

According to (Sukhbaatar et al. 2018), Alice's reward fuction is given by $r_A = \max(0, t_B - t_A)$ where t_B is the time taken for Bob to reach the goal state proposed by Alice, whereas t_A is the time taken by Alice to propose a goal (execute 'stop' action). In the target task, the goal is to craft a pogo-stick. We see that Alice's rewards are sparse, and for Alice to keep receiving positive rewards, it needs to propose a goal that is just difficult for Bob to repeat. In our task, the large goal space makes it difficult for Alice to propose a task that is just difficult for Bob to repeat, making it difficult to optimize the curriculum according to the goals proposed by Alice, taking it longer to obtain a curriculum.

One difficulty is that the time (in interactions) spent by Alice in proposing a task is a sunk cost, and requires tuning the self-play percentage parameter to not increase this added cost. The self-play approach does help Bob explore the space, and is responsible for the jumpstart Bob achieves initially in the target task, evident from the baseline comparison learning curves. From our experiments, we observed that the exploration helped Bob in navigation, but was not

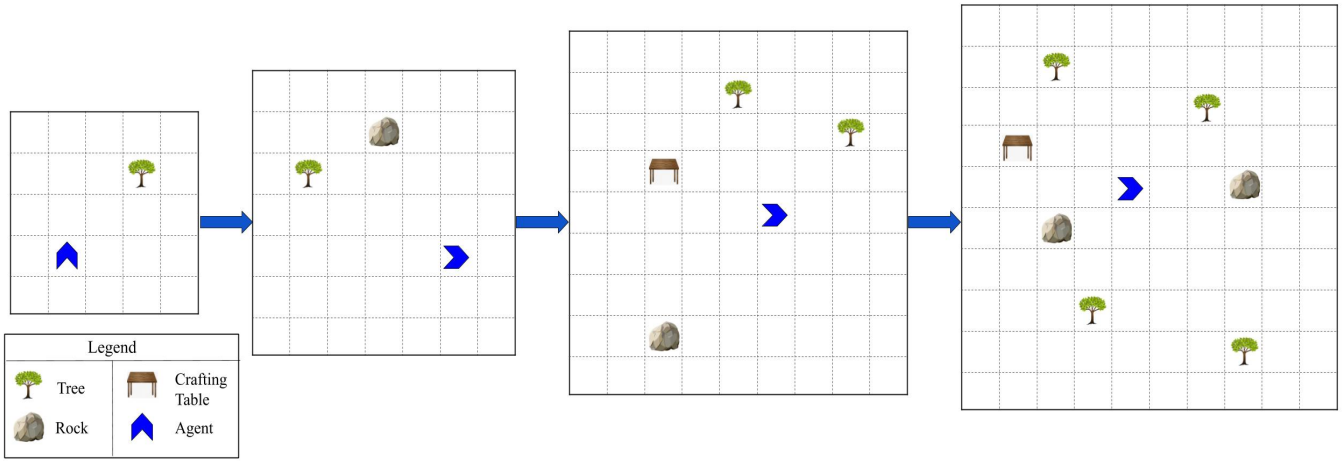

Figure 2: An example of a sequence-based curriculum. The agent learns complex tasks incrementally, before learning the final target task.

effective in performing the sequential actions required for the target task.

C.2 Teacher-Student Curriculum Learning

The proposed teacher-student curriculum learning (Matiisen et al. 2020) assumes a teacher optimizing the sequence of the tasks in the curriculum for the student. The teacher proposes those tasks for which the student shows the highest potential in learning, i.e. the tasks for which the student learns the quickest. The ability of a student to learn a task is given by its slope of the curve of plotted reward function. The steeper the slope, the quicker the student is able to learn the task. One limitation of this approach is that it assumes the source tasks of the curriculum are defined. This is difficult in settings where the parameters of the task are continuous. We used the same DQN network for teacher-student curriculum learning approach as we did for *AGCL*. We initialized 4 source tasks and one target task, with varying parameters for each source task. The performance of the agent in the final target task is plotted. We followed the same reward shaping for the tasks in the curriculum as for the source tasks of the automated curriculum transfer approach.

In teacher-student curriculum learning, the sunk cost of the algorithm is the interactions in the source task. The sunk cost is very high when the interactions are costly, as in our high fidelity setup. This contributes to the high requirement of compuational resources.

C.3 Q-Learning for Reward Machines

QRM (Icarte et al. 2018) decomposes the task into sub-goals and intends to learn one q-value function for each state in the automaton. As an example, consider the automaton in Fig. 1, each of the nodes in the automaton will have a corresponding q-value function. At all times during the training, the agent keeps track of its state in the automaton. While choosing an action given the current observation, the agent chooses action depending on its current state in the reward machine. For example, if the agent is in node ω_1, it will choose an action given by the q-value function q_{ω_1} corresponding to the node ω_1. While updating the q-value function, we use the reward machine to determine the reward the agent would have received had it been in the reward state given by the action chosen, i.e. to update

$$q_\omega \xleftarrow{\alpha} r(s,a,s') + \gamma \max_{a'} q_{\omega'}(s',a')$$

Here, the maximization is over the $q_{\omega'}$ that determines the new reward state of the machine.

We implemented this approach for the grid-world domain where the LTL$_f$ objective is given by:

$$\mathbf{G}((t \rightarrow \neg r \wedge \neg p) \wedge (r \rightarrow \neg t \wedge \neg p) \wedge (p \rightarrow \neg r \wedge \neg t)) \\ \wedge (\neg p \mathbf{U}(t \wedge \mathbf{X}(\neg p \mathbf{U} t))) \wedge (\neg p \mathbf{U} r) \wedge \mathbf{F} p \quad (1)$$

where t, r, p are the atomic propositions corresponding to *tree*, *rock* and *pogo-stick* respectively.

C.4 Guiding Search via Reward Shaping

GSRS (Camacho et al. 2018) shapes the reward inversely proportional to the distance from the accepting node in the automaton. The reward shaping is given by: $R'(s,a,s') = R(s,a,s') + F(s,a,s')$ where R is the original reward function and F is the shaping function. Here, F is given by:

$$F(s,a,s') = \gamma\phi(s') - \phi(s)$$

Here, F does not depend on the previous 2 states, but the last two states visited in the reward machine. This ensures that a positive shaping reward is given to transitions that make progress toward the goal state in the automaton. We shape the reward incrementally from the start state in the automaton until the accepting state, such that everytime the agent takes an action that triggers a transition in the automaton to a different state, a positive reward is given that is higher than the previous reward. In the end, to preserve optimality, we subtract the shaped reward from the sum of rewards gained.

D Statistical Significance

To demonstrate that the average convergence rate of *AGCL* is consistently higher than the baseline approaches, we perform an unpaired t-test (Kim 2015). For the experiment, we consider a confidence interval of 95% and evaluate the p-value between the best performing *AGCL* approach (graph-based curriculum) and the best performing baseline approach by comparing the success rates of the two approaches on 100 episodes after training for 10^7 interactions. Thus, through the results, we see that our proposed approach, *AGCL* has a consistent performance in the convergence rate. The results are always statistically significant. In all the experiments, *AGCL* has a much more sample efficient performance. Thus, *AGCL* not only achieves a better success rate, but also adapts to converges quicker.

E Hyperparameters

The table below summarizes the hyperparameters for the DQN (Mnih et al. 2015) used for the experiments.

All the experiments were conducted using a 64-bit Linux Machine, having Intel(R) Core(TM) i9-9940X CPU @ 3.30GHz processor and 126GB RAM memory. The maximum duration for running the experiments was set at 24 hours.

Parameter	Value
discount factor	0.995
learning rate	1×10^{-3}
optimizer	Adam
batch size	256
action distribution	categorical with 5 bins

References

Camacho, A.; Chen, O.; Sanner, S.; and McIlraith, S. A. 2018. Non-Markovian rewards expressed in LTL: Guiding search via reward shaping (extended version). In *GoalsRL, a workshop collocated with ICML/IJCAI/AAMAS*.

Icarte, R. T.; Klassen, T.; Valenzano, R.; and McIlraith, S. 2018. Using reward machines for high-level task specification and decomposition in reinforcement learning. In *International Conference on Machine Learning*, 2107–2116. PMLR.

Kim, T. K. 2015. T test as a parametric statistic. *Korean journal of anesthesiology*, 68(6): 540–546.

Matiisen, T.; Oliver, A.; Cohen, T.; and Schulman, J. 2020. Teacher-Student Curriculum Learning. *IEEE Trans. Neural Networks Learn. Syst.*, 31(9): 3732–3740.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A. A.; Veness, J.; Bellemare, M. G.; Graves, A.; Riedmiller, M.; Fidjeland, A. K.; Ostrovski, G.; Petersen, S.; Beattie, C.; Sadik, A.; Antonoglou, I.; King, H.; Kumaran, D.; Wierstra, D.; Legg, S.; and Hassabis, D. 2015. Human-level control through deep reinforcement learning. *Nature*, 518(7540): 529–533.

Sukhbaatar, S.; Lin, Z.; Kostrikov, I.; Synnaeve, G.; Szlam, A.; and Fergus, R. 2018. Intrinsic Motivation and Automatic Curricula via Asymmetric Self-Play. In *International Conference on Learning Representations*.