Exploiting Contextual Structure to Generate Useful Auxiliary Tasks

Benedict Quartey
Department of Computer Science
Brown University

Ankit Shah
Department of Computer Science
Brown University

George Konidaris
Department of Computer Science
Brown University

Abstract. Reinforcement learning requires interaction with environments, which can be prohibitively expensive, especially in robotics. This constraint necessitates approaches that work with limited environmental interaction by maximizing the reuse of previous experiences. We propose an approach that maximizes experience reuse while learning to solve a given task by generating and simultaneously learning useful auxiliary tasks. To generate these tasks, we construct an abstract temporal logic representation of the given task and leverage large language models to generate context-aware object embeddings that facilitate object replacements. Counterfactual reasoning and off-policy methods allow us to simultaneously learn these auxiliary tasks while solving the given target task. We combine these insights into a novel framework for multitask reinforcement learning and experimentally show that our generated auxiliary tasks share similar underlying exploration requirements as the given task, thereby maximizing the utility of directed exploration. Our approach allows agents to automatically learn additional useful policies without extra environment interaction.

Motivation

How do we get reinforcement learning agents to maximize the utility of their environment interaction experiences??

Proposal. Generate and learn auxiliary tasks that maximally benefit from the constrained exploration experience of single task curricula.

- Exploit the structure of the given task to obtain a task template.
- Exploit real-world contextual relationships between objects to generate new tasks with that template.

Key takeaway. Policies for auxiliary tasks can be learnt simultaneously without extra environment interaction via **counterfactual experience reasoning** and **off-policy RL**.

Framework for multi-task RL where given a task, an agent can generate auxiliary tasks such that it can perform efficient

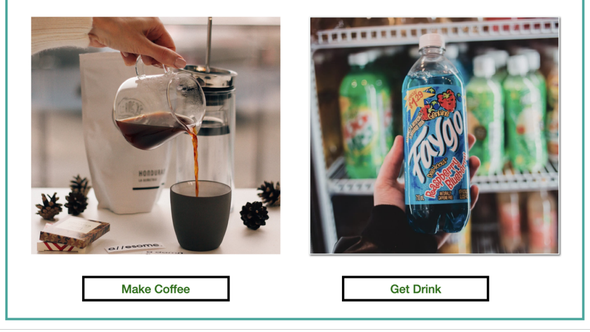
Yields constrained → Directed Exploration

Experience → Relevant Auxiliary Tasks → But reuse

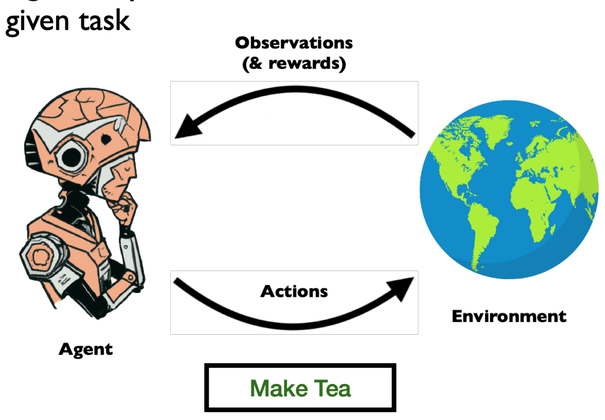
Given a task

Make Tea

Agent generates auxiliary tasks



Agent only interacts in environment to learn given task



Reuses experience learning given task to simultaneously learn auxiliary tasks

Assumptions

Tasks are represented in linear temporal logic (LTL), an expressive grammar for specifying temporal behavior composed of atomic propositions, logical connectives and the temporal operators:

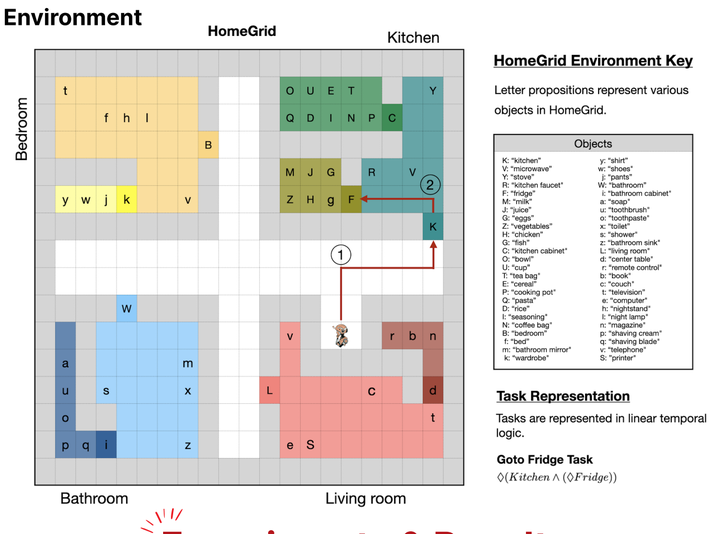
Next (○) Until (U) Always (□) Eventually (◊)

LTL formulae can be progressed given a sequence of proposition truth assignments to determine which parts of the formula have been satisfied by prior states and which parts remain--a useful property for tracking non-Markovian objectives in the reinforcement learning setting.

Example. A sequential task of visiting a Kitchen then a Fridge, where Kitchen and Fridge are Boolean propositions that can be observed is represented as:

$$\Diamond(Kitchen \wedge (\Diamond Fridge))$$

Environment



Approach

Given Task Specification → (1) → **Abstract Task Template** (Object LLM Embeddings) → (2) → **Generate template conditioned auxiliary tasks** → (3) → **Given Task + Auxiliary Tasks** → (4) → **Off-Policy RL**

Q-learning update using experience (s, a, s')

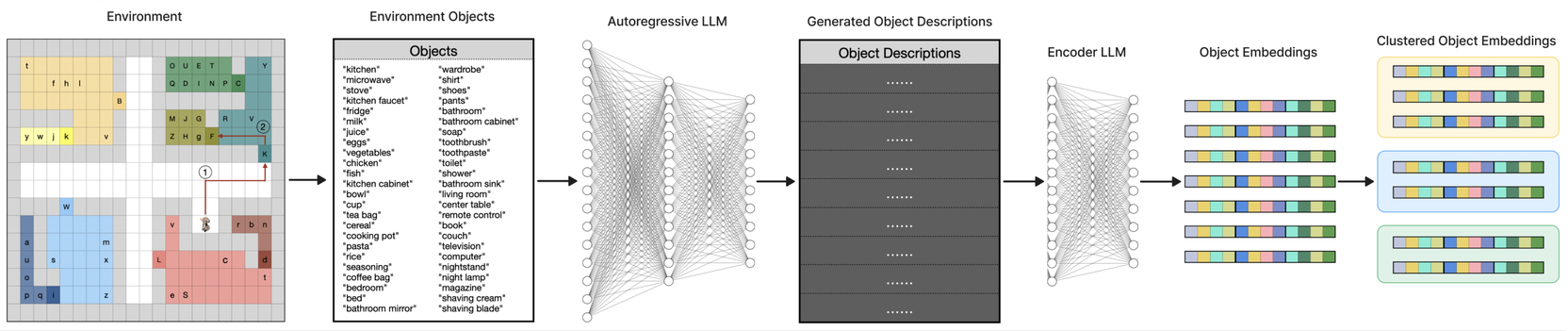
$$Q_\varphi(s,a) \xleftarrow{\alpha} r_\varphi + \gamma \max_{a'} Q_{\varphi'}(s', a')$$

where $\varphi' = \text{prog}(s', \varphi)$ and $r_\varphi = 1$ iff $\varphi \neq \varphi' = \text{true}$.

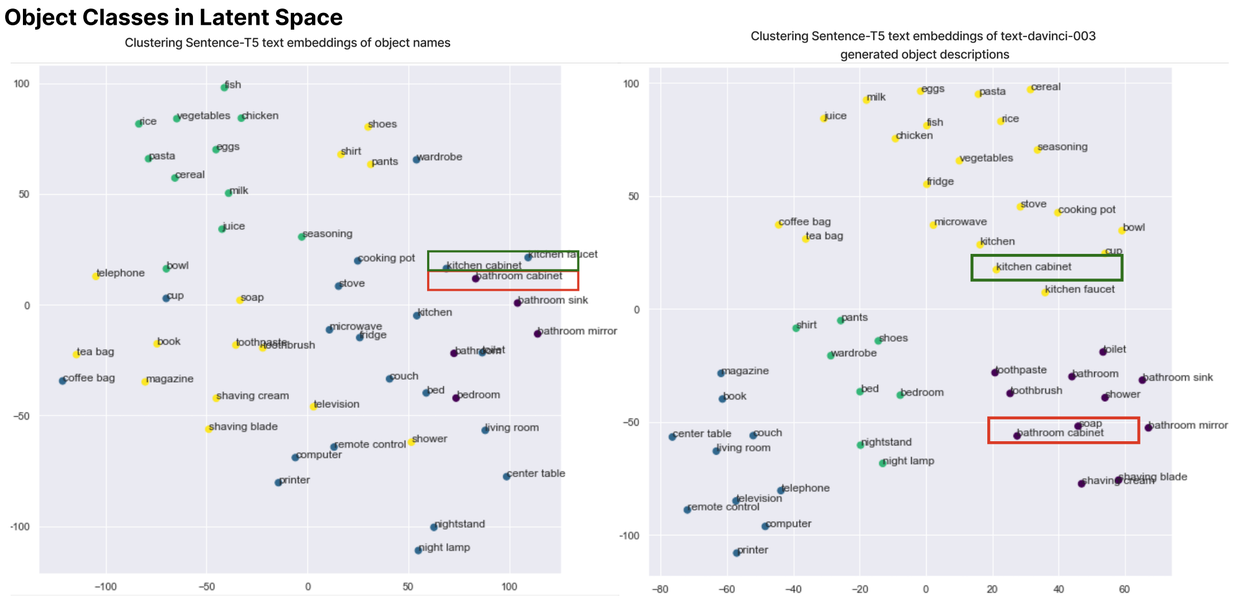
Behavior policy ε-greedy on given task

Generating Auxiliary Tasks

Exploiting Contextual Relationships between Objects



Visualizing Emergent Object Classes in Latent Space



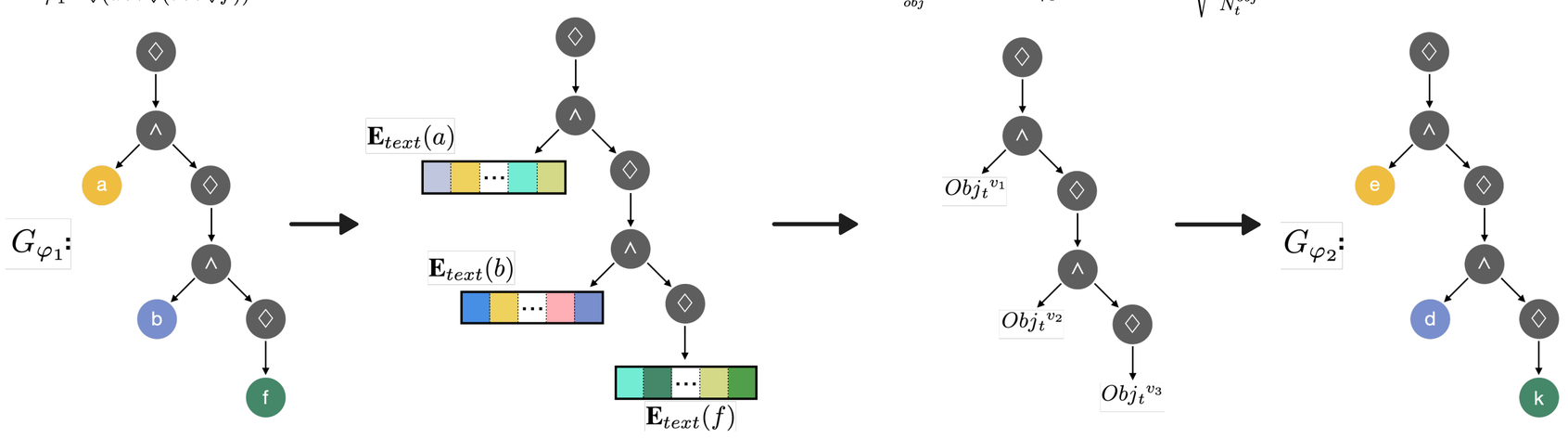
Exploiting Task Structure

$\varphi_1 : \Diamond(a \wedge \Diamond(b \wedge \Diamond f))$

$\Diamond(\mathbf{E}_{text}(a) \wedge \Diamond(\mathbf{E}_{text}(b) \wedge \Diamond \mathbf{E}_{text}(f)))$

$Obj_t{}^{v_i} = \max_{obj}[sim(\mathbf{E}_{text}(G^{v_i}_{\varphi_1}), \mathbf{E}_{text}(obj)) + c\sqrt{\frac{\log N_t}{N_t^{obj}}}]$

$\varphi_2 : \Diamond(e \wedge \Diamond(d \wedge \Diamond k))$



Experiments & Results

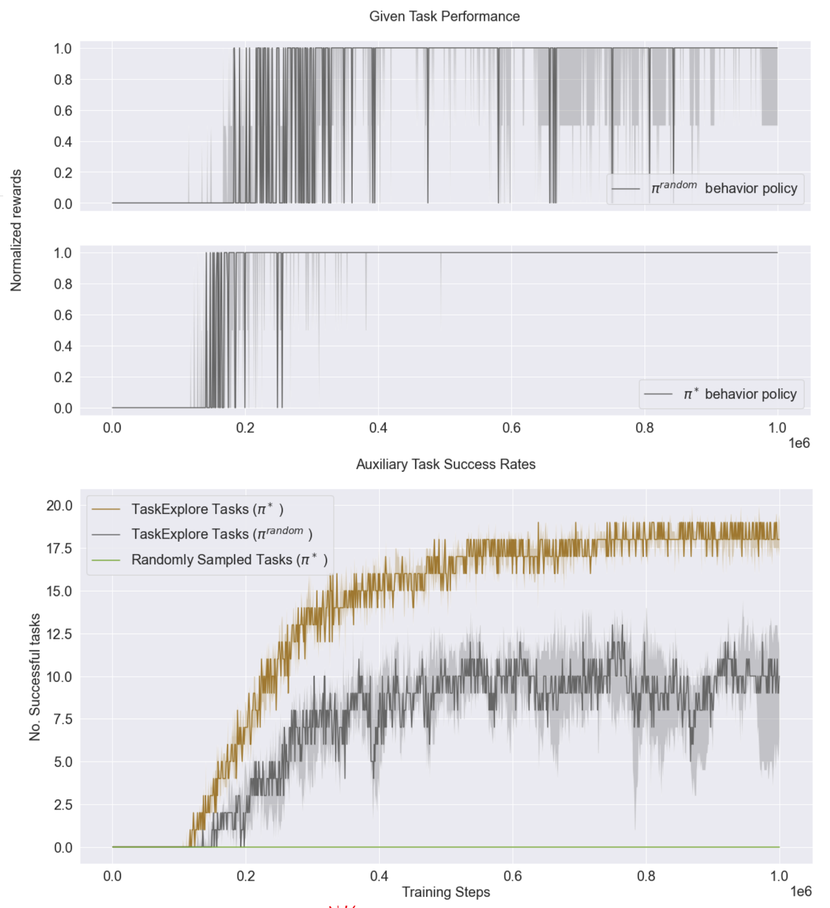
The task specification used in our experiments was a food preparation task where the agent had to go to the **kitchen cabinet**, obtain a **cooking pot**, obtain **seasoning**, then go to the **fridge**, obtain **chicken**, and finally go to the **stove**. In HomeGrid, this task corresponds to visiting the right cells in the correct order. The LTL formula below represents this task using the atomic propositions that represent each of the relevant objects:

$$\Diamond(C \wedge \Diamond(P \wedge \Diamond(I \wedge \Diamond(F \wedge \Diamond(H \wedge \Diamond Y)))))$$

Ours: Given our food preparation task φ1 we generate 20 auxiliary tasks following our approach. We learn these tasks simultaneously with φ1 with a behavior policy epsilon greedy on φ1, directing exploration towards more relevant experiences for φ1.

Baseline 1: To demonstrate that tasks generated by TaskExplore uniquely leverage the directed experience of a single-task curriculum, we repeat the approach described above, replacing the behavior policy with a random one that explores more widely.

Baseline 2: To show that tasks generated by TaskExplore more relevantly benefit from directed exploration experience than the general set of possible tasks, we generate 20 auxiliary tasks by randomly sampling equal length sequential tasks from the set of propositions in our environment, as typically done in prior works. We learn these tasks simultaneously with φ1, using a behavior policy epsilon greedy on φ1.



Conclusion

This work introduced an approach that allows agents to generate relevant auxiliary tasks that maximally benefits from the directed exploration experience of a single task curriculum.

This approach to auxiliary task generation is particularly valuable in the lifelong learning setting, as agents can generate and solve new tasks from constrained experience datasets.

Modern vision-language models (VLMs) that detect open vocabulary objects in real world environments can be employed with this approach to relax the dependence on predefined sets of object propositions from which tasks can be expressed and labelling functions that map states to proposition truth values.