Skill Transfer for Temporal Task Specification: Supplementary Materials

Jason Xinyu Liu*, Ankit Shah*, Eric Rosen, Mingxi Jia, George Konidaris and Stefanie Tellex

I. LPOPL

A DQN policy is a feedforward network with two hidden layers, each of which has 64 ReLU units. We used the same hyperparameters suggested in [1] for training, i.e., the learning rate is 0.0001; the size of the replay buffer is 25,000; 32 transitions are randomly sampled from the replay buffer for every update; discount factor is 0.9; exploration decreases linearly from 1 to 0.02.

II. Example LTL Formulas

We provide LTL task specifications and their interpretations from the *Hard*, *Soft*, *Strictly Soft*, *No Orders*, and *Mixed* formula types. Note that each training set contains 50 formulas (new formulas were added incrementally as the training set size increased), and each test set contains 100 formulas.

Hard: Example formulas and their interpretations from the *Hard* type are as follows:

1) $\mathbf{F} wood \land \mathbf{F} axe \land \neg wood\ \mathbf{U}\ grass \land \neg grass\ \mathbf{U}\ workbench \land \neg workbench\ \mathbf{U}\ bridge$: Visit $bridge$, $workbench$, $grass$, $wood$, and axe. Ensure that $bridge$, $workbench$, $grass$, $wood$ in that particular order. Objects later in the sequence cannot be visited before the prior objects.
2) $\mathbf{F} workbench \land \mathbf{F} factory \land \mathbf{F} iron \land \mathbf{F} shelter \land \neg factory\ \mathbf{U}\ axe$: Visit $workbench$, $factory$, $iron$, $shelter$, and axe. Ensure that $factory$ is not visited before axe.
3) $\mathbf{F} toolshed \land \mathbf{F} bridge \land \mathbf{F} factory \land \mathbf{F} axe \land \neg bridge\ \mathbf{U}\ wood$: Visit $toolshed$, $bridge$, $factory$, axe, and $wood$. Ensure that $bridge$ is not visited before $wood$.

Soft: Example formulas and their interpretations from the *Soft* type are as follows:

1) $\mathbf{F}(bridge \land \mathbf{F}(factory \land \mathbf{F}(iron \land \mathbf{F} shelter)))$: Visit $bridge$, $factory$, $iron$, and $shelter$ in that sequence. The objects later in the sequence may be visited before the prior objects, provided that they are visited at least once after the prior object has been visited.
2) $\mathbf{F} workbench \land \mathbf{F}(factory \land \mathbf{F} grass)$: Visit the $workbench$, $factory$, and $grass$: Visit $grass$ at least once after visiting the $factory$.
3) $\mathbf{F}(axe \land \mathbf{F} factory) \land \mathbf{F} workbench$: Visit axe, $factory$, and $workbench$. Ensure that $factory$ is visited at least once after axe is.

Strictly Soft: Example formulas and their interpretations from the *Strictly Soft* type are identical to the *Soft* specifications, except they do not allow for simultaneous satisfaction of multiple sub-tasks. The subtasks in sequence must occur strictly temporally after the prior subtask. This is enforced using $\mathbf{XF}a$ instead of $\mathbf{F}a$.

No Orders: These specifications only contain a list of subtasks to be completed. No temporal orders are enforced between the various subtasks.

1) $\mathbf{F} wood \land \mathbf{F} grass \land \mathbf{F} stone$: Visit $bridge$: Collect $wood$, $grass$, $stone$ in no particular order.

Mixed: Example formulas and their interpretations from the *Mixed* type are as follows:

1) $\mathbf{F} toolshed \land \mathbf{F} factory \land \neg toolshed\ \mathbf{U}\ shelter \land \mathbf{F}(grass \land \mathbf{F} bridge)$: Visit the $toolshed$, $factory$, $shelter$, $grass$, and $bridge$. Ensure that $toolshed$ is not visited before the shelter, and $bridge$ is visited at least once after $grass$.
2) $\mathbf{F} grass \land \neg grass\ \mathbf{U}\ toolshed \land \mathbf{F}(factory \land \mathbf{XF} workbench)$: Visit $grass$, $toolshed$, $factory$, and $workbench$. Ensure that $grass$ is not visited before $toolshed$, and $workbench$ is at least visited once strictly after $factory$.
3) $\mathbf{F} iron \land \neg iron\ \mathbf{U}\ toolshed \land \mathbf{F}(shelter \land \mathbf{XF} wood)$: Visit $iron$, $toolshed$, $shelter$, and $wood$. Ensure that $iron$ is not visited before $toolshed$, and $wood$ is at least visited once strictly after $shelter$.

III. Additional Experimental Results

Learning Curves for Various Training Sets: We present the results for learning curve of the success rate when transferring policies learned on different specification types.

The learning curves for training on formulas from the *Hard* training set with both the edge matching criteria are depicted in Figure 1.

The learning curves for training on formulas from the *Soft* training set with both the edge matching criteria are depicted in Figure 2.

The learning curves for training on formulas from the *Strictly Soft* training set with both the edge matching criteria are depicted in Figure 3.

The learning curves for training on formulas from the *No Orders* training set are being generated at the time of submission, and are expected to share nearly identical trends as the learning curves from the other training sets given the completed data points. We will include the plots in the final version of the paper.

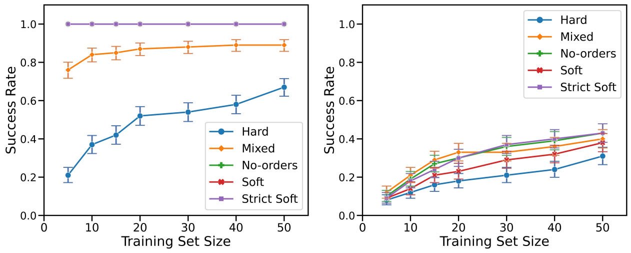
Note that for training on each of the specification types, the learning curve trends are nearly identical to the learning curves on training with *Mixed* specification types as depicted

*These authors contributed equally.

in Figure 3 in the main paper. *Hard* specification types remain the most challenging specification ordering types to transfer to.

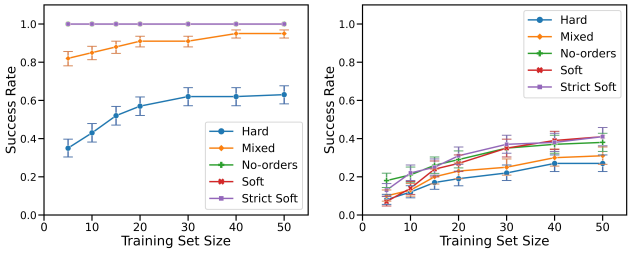
Failure Analysis: As described in the main paper, we logged the reason for failure of each unsuccessful transfer attempt as one of three possible causes: *specification failure*, where the agent violates a constraint and the reward machine is progressed to an unrecoverable state; *no feasible path*, where there are no matched transition-centric options for paths connecting the start state to an accepting state; *options exhausted*, where there are no further transition-centric options available to the agent to further progress the state of the task.

Figure 4 depicts the relative frequency of the failure modes when LTL-Transfer is trained and tested on *mixed* task specifications. Note that with the *Constrained* edge-matching criterion, absence of feasible paths connecting the start and the accepting state is the primary reason for failure (Figure 4b, whereas with the *Relaxed* edge-matching criterion, the agent utilizing all available safe options without progressing the task is the primary reason for failure (Figure 4a).



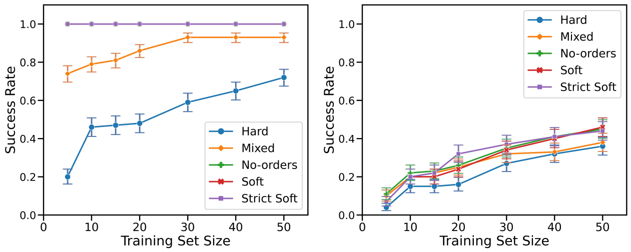
(a) *Relaxed* Edge Match (b) *Constrained* Edge Match

Fig. 1: Figure 1a depicts the success rate of LTL-Transfer after being trained on *Hard* training sets of various sizes using LTL-Transfer with the *Relaxed* edge-matching criterion when transferring to test sets of various specifications types. Figure 1b depicts the success rates with the *Constrained* edge-matching criterion. Note that the error bars depict the 95% credible interval if the successful transfer was modeled as a Bernoulli distribution.



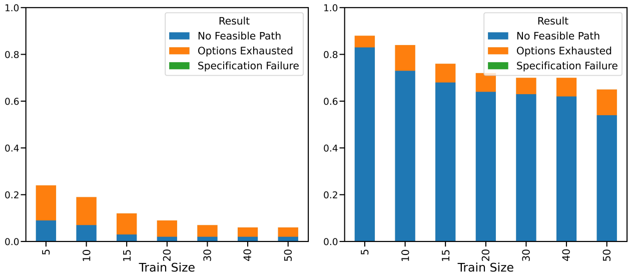
(a) *Relaxed* Edge Match (b) *Constrained* Edge Match

Fig. 2: Figure 2a depicts the success rate of LTL-Transfer after being trained on *Soft* training sets of various sizes using LTL-Transfer with the *Relaxed* edge-matching criterion when transferring to test sets of various specifications types. Figure 2b depicts the success rates with the *Constrained* edge-matching criterion. Note that the error bars depict the 95% credible interval if the successful transfer was modeled as a Bernoulli distribution.



(a) *Relaxed* Edge Match (b) *Constrained* Edge Match

Fig. 3: Figure 3a depicts the success rate of LTL-Transfer after being trained on *Strictly Soft* training sets of various sizes using LTL-Transfer with the *Relaxed* edge-matching criterion when transferring to test sets of various specifications types. Figure 3b depicts the success rates with the *Constrained* edge-matching criterion. Note that the error bars depict the 95% credible interval if the successful transfer was modeled as a Bernoulli distribution.



(a) *Relaxed* Edge Match (b) *Constrained* Edge Match

Fig. 4: Reasons for failed task executions after being trained and evaluated on *Mixed* task specification datasets. Note that all values are depicted in fractions.

IV. Selected Solution Trajectories in Simulation

Consider the case with *mixed* training set with 5 formulas on *map 0*. The training formulas are:

- $\mathbf{F}\mathit{grass} \wedge \mathbf{F}\mathit{shelter} \wedge \mathbf{F}(\mathit{wood} \wedge \mathbf{XF}\mathit{workbench})$
- $\mathbf{F}\mathit{toolshed} \wedge \mathbf{F}\mathit{workbench} \wedge \mathbf{F}\mathit{shelter} \wedge (\neg \mathit{toolshed}\ \mathbf{U}\ \mathit{shelter}) \wedge \mathbf{F}(\mathit{grass} \wedge \mathbf{F}\mathit{bridge})$
- $\mathbf{F}\mathit{toolshed} \wedge \mathbf{F}(\mathit{shelter} \wedge \mathbf{F}(\mathit{axe} \wedge \mathbf{F}\mathit{wood}))$
- $\mathbf{F}\mathit{iron} \wedge \mathbf{F}(\mathit{shelter} \wedge \mathbf{XF}(\mathit{bridge} \wedge \mathbf{XF}\mathit{factory}))$
- $\mathbf{F}\mathit{factory}$

One of the *Mixed* test formulas was $\varphi_{test} = \mathbf{F}\mathit{workbench} \wedge \mathbf{F}\mathit{grass} \wedge \mathbf{F}\mathit{axe}$. The reward machine for this task specification is depicted in Figure **??**. Given the training set of formulas, and the use of the *Constrained* edge matching criterion, the start state is disconnected from all downstream states as no transition-centric options match with the edge transitions. Therefore, the agent does not attempt to solve the task and returns failure with the reason being *no feasible path*, i.e., a disconnected reward machine graph after removing infeasible edges.

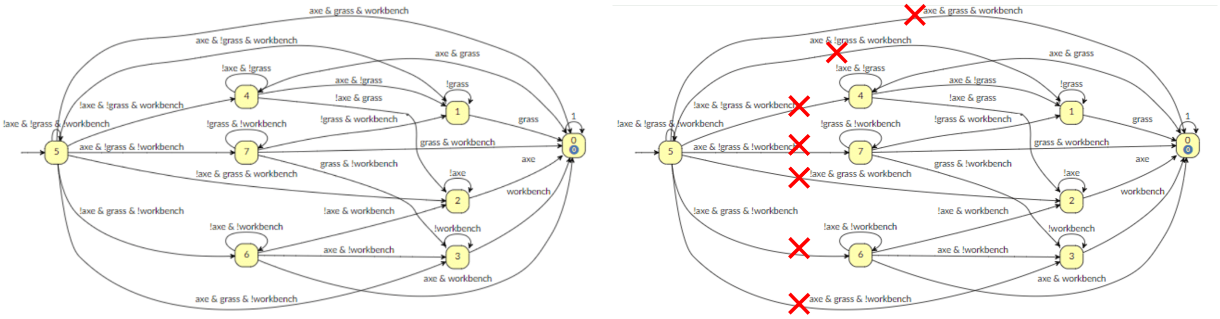
If the *Relaxed* edge matching criterion is used, there are matching transition-centric options for each of the RM edges. The trajectory adopted by the agent when transferring the policies is depicted in Figure 6. The agent collects all the three requisite resources before it terminates the task execution. Further note that the agent passes through a $wood$ resource grid as the specification does not explicitly prohibit it.

V. Robot Demonstration

The 50 test tasks that were executed on the robot are shown in Table I, where a represents a brown desk, b represents a white desk, c represents a couch, d represents a door, s represents a bookshelf, k represents a kitchen counter.

References

[1] R. Toro Icarte, T. Q. Klassen, R. Valenzano, and S. A. McIlraith, "Teaching multiple tasks to an RL agent using LTL," in *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*, 2018, pp. 452–461.



(a) (b)

Fig. 5: Figure 5a depicts the reward machine for the the specification $\varphi_{test} = \mathbf{F}workbench \wedge \mathbf{F}grass \wedge \mathbf{F}axe$, as well as all feasible edges matched by the *Relaxed* criterion. Note that all the edges have at least one matched transition-centric option for the *Relaxed* criterion. Figure 5b depicts the edges that do not have a compatible transition-centric option for the *Constrained* edge matching criterion.

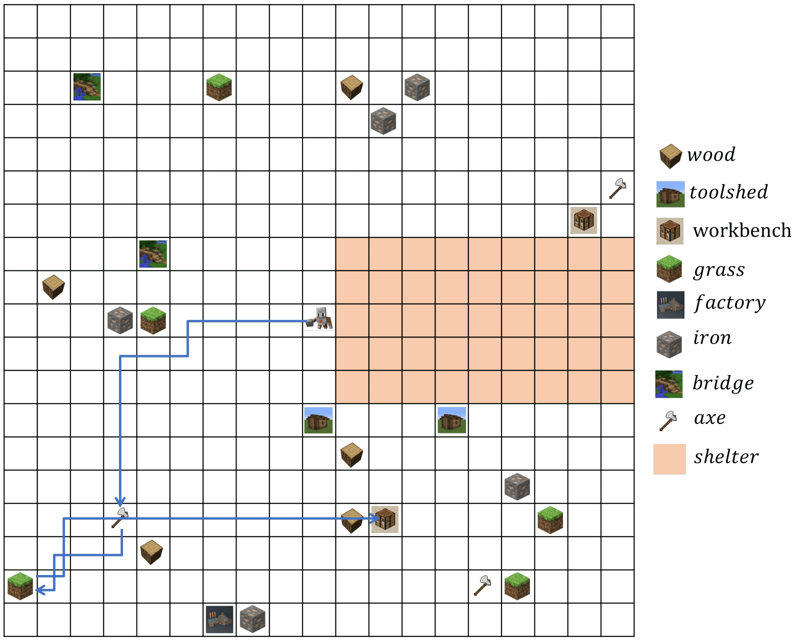


Fig. 6: Trajectory executed by the robota using LTL-Transfer on the specification $\varphi_{test} = \mathbf{F}workbench \wedge \mathbf{F}grass \wedge \mathbf{F}axe$.

TABLE I: Test Tasks Executed on the Robot

LTL Task Specification	Type of Task	Results
0. $\mathbf{F}a$	navigation	success
1. $\mathbf{F}a \wedge \mathbf{F}b$	navigation	success
2. $\mathbf{F}a \wedge \mathbf{F}b \wedge \mathbf{F}c$	navigation	success
3. $\mathbf{F}a \wedge \mathbf{F}b \wedge \mathbf{F}s$	navigation	success
4. $\mathbf{F}a \wedge \mathbf{F}b \wedge \mathbf{F}k$	navigation	success
5. $\mathbf{F}a \wedge \mathbf{F}b \wedge \mathbf{F}c \wedge \mathbf{F}d$	navigation	success
6. $\mathbf{F}a \wedge \mathbf{F}b \wedge \mathbf{F}c \wedge \mathbf{F}s$	navigation	success
7. $\mathbf{F}a \wedge \mathbf{F}b \wedge \mathbf{F}c \wedge \mathbf{F}k$	navigation	success
8. $\mathbf{F}a \wedge \mathbf{F}b \wedge \mathbf{F}c \wedge \mathbf{F}k \wedge \mathbf{F}s$	navigation	success
9. $\mathbf{F}(b \wedge \mathbf{F}(a \wedge \mathbf{F}(c \wedge \mathbf{F}d))))$	navigation	success
10. $\mathbf{F}(s \wedge \mathbf{F}a)$	fetch and deliver	success
11. $\mathbf{F}(s \wedge \mathbf{F}b)$	fetch and deliver	success
12. $\mathbf{F}(a \wedge \mathbf{F}b)$	navigation	success
13. $\mathbf{F}(b \wedge \mathbf{F}a)$	navigation	success
14. $\mathbf{F}(a \wedge \mathbf{F}(s \wedge \mathbf{F}c))$	fetch and deliver	success
15. $\mathbf{F}(b \wedge \mathbf{F}(s \wedge \mathbf{F}c))$	fetch and deliver	success
16. $\mathbf{F}(s \wedge \mathbf{F}(a \wedge \mathbf{F}c))$	fetch and deliver	success
17. $\mathbf{F}(a \wedge \mathbf{F}(b \wedge \mathbf{F}c))$	navigation	success
18. $\mathbf{F}(s \wedge \mathbf{F}(a \wedge \mathbf{F}(k \wedge \mathbf{F}a))))$	fetch and deliver	success
19. $\mathbf{F}(a \wedge \mathbf{F}(b \wedge \mathbf{F}(c \wedge \mathbf{F}d))))$	navigation	success
20. $\mathbf{F}(s \wedge \mathbf{XF}a)$	fetch and deliver	success
21. $\mathbf{F}(b \wedge \mathbf{XF}s)$	navigation	success
22. $\mathbf{F}(a \wedge \mathbf{XF}b)$	navigation	success
23. $\mathbf{F}(b \wedge \mathbf{XF}a)$	navigation	success
24. $\mathbf{F}(a \wedge \mathbf{XF}(b \wedge \mathbf{XF}c))$	navigation	success
25. $\mathbf{F}(a \wedge \mathbf{XF}(s \wedge \mathbf{XF}b))$	fetch and deliver	success
26. $\mathbf{F}(b \wedge \mathbf{XF}(s \wedge \mathbf{XF}a))$	fetch and deliver	success
27. $\mathbf{F}(s \wedge \mathbf{XF}(b \wedge \mathbf{XF}a))$	fetch and deliver	success
28. $\mathbf{F}(k \wedge \mathbf{XF}b)$	fetch and deliver	success
29. $\mathbf{F}(k \wedge \mathbf{XF}a)$	fetch and deliver	success
30. $\neg a \mathbf{U} s \wedge \mathbf{F}a$	fetch and deliver	success
31. $\neg b \mathbf{U} a \wedge \mathbf{F}b$	navigation	success
32. $\neg a \mathbf{U} b \wedge \mathbf{F}a$	navigation	success
33. $b \mathbf{U} a \wedge \neg c \mathbf{U} b \wedge \mathbf{F}c$	navigation	success
34. $b \mathbf{U} k \wedge \mathbf{F}b$	fetch and deliver	success
35. $\neg b \mathbf{U} c \wedge \mathbf{F}b$	navigation	success
36. $\neg a \mathbf{U} s \wedge \neg b \mathbf{U} a \wedge \mathbf{F}b$	fetch and deliver	success
37. $\neg s \mathbf{U} a \wedge \neg b \mathbf{U} s \wedge \mathbf{F}b$	fetch and deliver	success
38. $\neg b \mathbf{U} a \wedge \neg s \mathbf{U} b \wedge \mathbf{F}s$	navigation	success
39. $\neg a \mathbf{U} b \wedge \neg s \mathbf{U} a \wedge \mathbf{F}s$	navigation	success
40. $\mathbf{F}a \wedge \mathbf{F}(b \wedge \mathbf{F}c)$	navigation	success
41. $\mathbf{F}a \wedge \neg c \mathbf{U} b \wedge \mathbf{F}c$	navigation	success
42. $\mathbf{F}(a \wedge \mathbf{F}b) \wedge \neg c \mathbf{U} a \wedge \mathbf{F}c$	navigation	success
43. $\mathbf{F}a \wedge \mathbf{F}(b \wedge \mathbf{XF}c)$	navigation	success
44. $\mathbf{F}(a \wedge \mathbf{F}b) \wedge \neg c \mathbf{U} b \wedge \mathbf{F}c$	navigation	success
45. $\mathbf{F}(b \wedge \mathbf{F}a) \wedge \neg c \mathbf{U} b \wedge \mathbf{F}c$	navigation	success
46. $\mathbf{F}c \wedge (\neg s \mathbf{U} a \wedge \mathbf{F}s)$	navigation	success
47. $\mathbf{F}c \wedge (\neg a \mathbf{U} s \wedge \mathbf{F}a)$	navigation	success
48. $\mathbf{F}c \wedge (\neg s \mathbf{U} b \wedge \mathbf{F}s)$	navigation	success
49. $\mathbf{F}c \wedge (\neg b \mathbf{U} s \wedge \mathbf{F}b)$	navigation	success