Hybrid Offline Passive Grammatical Inference and Online Planning for Non-Markovian Tasks

Mahyar Alinejad[1], Alvaro Velasquez[2], Yue Wang[1], George Atia[1]

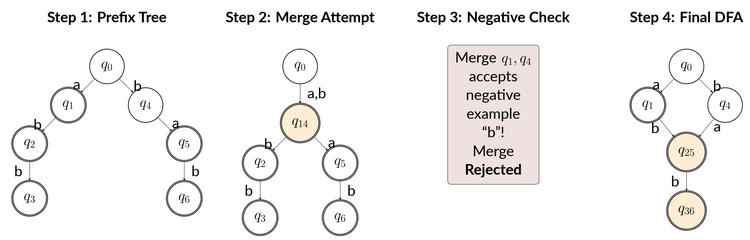
[1]Department of Electrical and Computer Engineering (and Computer Science), University of Central Florida, Orlando, FL
[2]Department of Computer Science, University of Colorado Boulder, CO

Abstract

In non-Markovian planning, rewards depend on the full state history, limiting standard RL methods. We introduce a hybrid approach combining **passive grammatical inference** (RPNI) with online planning to minimize costly environment interactions. Offline, a deterministic finite automaton (DFA) is learned from labeled positive and negative trajectories to capture task structure. Online, Q-learning is performed over a product MDP that integrates the environment with the learned DFA. Experiments show faster convergence and lower sample complexity than active grammar-based RL methods, validating the efficiency of our offline-online framework.

Illustrative Example of RPNI and Product MDP

Alphabet: $\{a, b\}$, **Positives:** $\{a, ab, abb, ba, bab\}$, **Negatives:** $\{b, aa, bb\}$



- **Step 1:** Build Prefix Tree from positive examples.
- **Step 2:** Attempt merging states q_1 and q_4.
- **Step 3:** Merge rejected due to accepting negative example ("b").
- **Step 4:** Final minimal DFA with allowed merges.

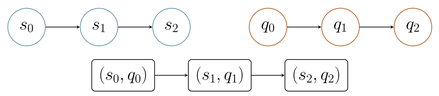


Figure 1. Product MDP: Combines environment state and DFA state to track task progress.

Problem Formulation

We consider a non-Markovian reward decision process (NMRDP)

$$\mathcal{M} = (S, s_0, A, P, \mathsf{AP}, L, R),$$

where S is a finite set of states, A is an action set, and P specifies transitions. A labeling function $L : S \to 2^{\mathsf{AP}}$ maps states to atomic propositions AP, and the reward R depends on the entire state-history.

A DFA $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$ tracks valid event sequences: Q finite set of automaton states, $\Sigma = 2^{\mathsf{AP}}$ task related labels, $\delta : Q \times \Sigma \to Q$, and $F \subseteq Q$ accepting states. We combine $\mathcal{M}$ and $\mathcal{A}$ into a *product MDP* $\mathcal{M}^{\otimes}$, where each state is (s, q). This transforms non-Markovian rewards into Markovian rewards in the augmented space.

Algorithmic Summary

HiPO (Hybrid Passive Inference and Online Planning)

1. **Offline RPNI:**
   - Input: pre-collected labeled trajectories $\{P_{traj}, N_{traj}\}$.
   - Output: learned DFA $\mathcal{A}$.
2. **Form Product MDP:** $\mathcal{M}^{\otimes} = \mathcal{M} \otimes \mathcal{A}$.
3. **Online Q-Learning** in $\mathcal{M}^{\otimes}$:

$$Q((s,q),a) \leftarrow Q((s,q),a) + \alpha\big[r + \gamma \max_{a'} Q((s',q'),a') - Q((s,q),a)\big].$$

Hybrid Approach: Offline & Online Components

Offline Passive Learning (RPNI):

1. Collect positive/negative trajectories offline via a *behavior policy*.
2. Label each trajectory using $L(s)$ to map states to alphabet $\Sigma = 2^{\mathsf{AP}}$.
3. Apply **RPNI** to build a *prefix tree acceptor* and iteratively merge states, ensuring negative sequences are rejected.

Result is a **minimal DFA** that captures valid event sequences.

Online Planning (Q-Learning in Product MDP):

1. Combine environment $\mathcal{M}$ with learned DFA $\mathcal{A}$ to form the product MDP $\mathcal{M}^{\otimes}$.
2. Run Q-learning to optimize rewards in the augmented state space (s, q).
3. By planning in $\mathcal{M}^{\otimes}$, the agent inherently follows correct label sequences to complete the non-Markovian task.

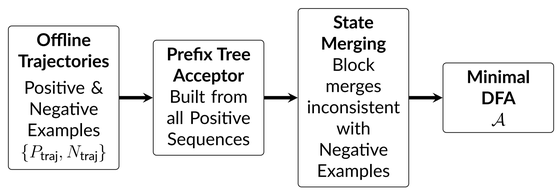


Figure 2. RPNI workflow: positive examples form a prefix tree; invalid merges are blocked by negative examples to learn a minimal DFA.

Illustrative Figures

Environments:

- *Office GridWorld* (9×12)*:* tasks: {coffee, mail, office}.
- *Minecraft GridWorld* (15×15)*:* tasks: {wood, iron, factory, bridge}.

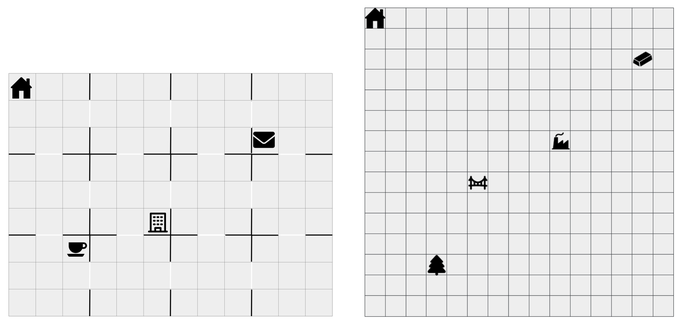


Figure 3. Two labeled gridworlds used for experiments: Office (left) and Minecraft (right). Each environment requires collecting items in a specific sequence.



Figure 4. An example DFA inferred for the Office environment.

Experimental Results

Baselines compared:

- **AGI** [**topper2022active**], **JIRP** [**xu2020joint**], **AFRAI** [**xu2021active**] (active automaton inference).
- **QAS** (manual augmented-state Q-learning).

Key Observations:

- **HiPO converges faster** in reward vs. time due to offline DFA learning.
- **Lower sample complexity**, as offline data reduces repeated online exploration of invalid sequences.
- **Robust to environment scale**: as tasks grow in complexity, offline inference offers more advantage.

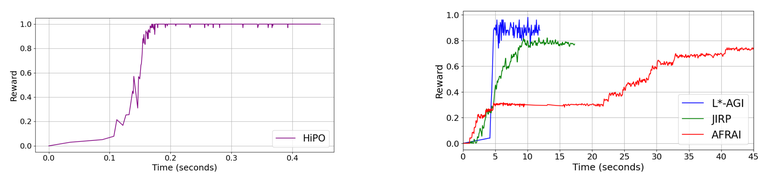


Figure 5. Fig. 3. (Left) Reward versus time for HiPO in the Office. (Right) Reward over time for AGI, JIRP, and AFRAI in the Office.

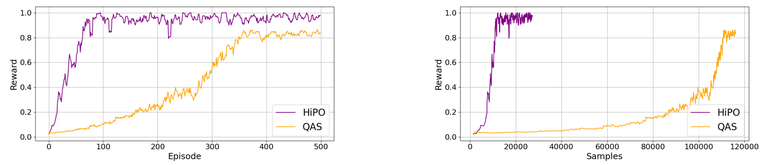


Figure 6. Fig. 5. (Left) Reward per episode for HiPO and QAS in Office. (Right) Samples vs reward for HiPO and QAS in Office.

Conclusion

HiPO combines offline, passive automaton learning (RPNI) and online planning (Q-learning in product MDP). This hybrid method:

- Reduces the need for extensive environment queries (active membership tests).
- Learns the non-Markovian task structure *once*, reusing it for efficient planning.
- Empirically yields faster convergence, lower sample complexity, and improved scalability in labeled gridworlds.

Future Directions:

- Offline RL integration: refine Q-values directly from offline data.
- Extension to partially observable tasks.

Acknowledgment: This work was supported by DARPA under Agreement No. HR0011-24-9-0427 and NSF under Award CCF-2106339.

References

[1] Z. Xu et al., "Joint Inference of Reward Machines and Policies for RL," *ICAPS*, 2020.
[2] N. Topper et al., "Active Grammatical Inference for Non-Markovian Planning," *ICAPS*, 2022.
[3] Z. Xu et al., "Active Finite Reward Automaton Inference ...," *AI Journal*, 2021.