Learning Symbolic Temporal Advice for Guiding Reinforcement Learning (Extended Abstract)

Paulius Skaisgiris[1,*], Celeste Veronese[2] and Daniele Meli[2]

[1]Institute of Logic and Computation, TU Wien, Vienna, 1040, Austria
[2]Department of Computer Science, University of Verona, Verona, 37134, Italy

Abstract

Reinforcement learning is sample-inefficient and opaque—problems that reward shaping and feature engineering often fail to address. We introduce a method to learn temporal advice from expert demonstrations via inductive logic programming, transforming the advice into answer set automata for transparent RL guidance and accelerating learning. Our advice is expressed in $\text{LTL}_\text{f}^\text{ASP}$ (LTL_f with Answer Set queries), enabling generalizable specifications that transfer from small to large map instances. In RL benchmarks, this temporal guidance outperforms unguided Q-learning and non-temporal baselines in both convergence rate and final performance.

Keywords

Inductive Logic Programming, LTLf with Answer Set Queries, Neurosymbolic Reinforcement Learning

1. Introduction

Reinforcement learning (RL) is often sample-inefficient and opaque, complicating its use in safety-critical domains. While symbolic guidance can partly mitigate these issues, manual specification is a bottleneck requiring domain expertise and risking reward hacking [1, 2]. Recent work learns guiding structures (e.g. reward machines [3], subgoal automata [4]) from data, but these are high-level, task-specific and not reusable in other environments.

We argue that advice should be *temporal, symbolic, and action-level* and we express it in $\text{LTL}_\text{f}^\text{ASP}$ [5]. The logic $\text{LTL}_\text{f}^\text{ASP}$ augments LTL_f [6] with answer set queries, providing a mechanism to evaluate the truth of queries against background knowledge and symbolic environmental data. This allows us to capture action preconditions, make relational statements about objects, and encode numeric constraints concisely, enabling reusable guidance across domains and tasks.

We use inductive learning from answer set programs (ILASP) [7] to automatically learn such temporal advice from expert demonstrations. The learned advice are then converted to Answer Set Automata [8] (automata with ASP normal rules on the edges guarding the transitions) which are subsequently used to guide tabular Q-learning.

Our contributions: (1) an ILASP framework inducing action-level $\text{LTL}_\text{f}^\text{ASP}$ advice from partially observable traces, unifying time-independent ASP advice [9] with temporal learning [10]; (2) extension of Veronese et al. [11]'s guidance from static rules to temporal advice via answer set automata; (3) empirical evaluation showing learned advice is interpretable, generalizes, and accelerates Q-learning versus unguided, and timeless and macro-action guiding baselines.

2. Methodology and empirical evaluation

2.1. Learning temporal advice

We use ILASP [7] to learn $\text{LTL}_\text{f}^\text{ASP}$ from examples. A *partial interpretation* $e = \langle e^{\text{inc}}, e^{\text{exc}} \rangle$ specifies atoms that must (e^{inc}) or must not (e^{exc}) appear in interpretations. A *context-dependent partial interpretation*

24th International Workshop on Nonmonotonic Reasoning, July 17–19, 2026, Lisbon, Portugal
*Corresponding author.
paulius.skaisgiris@tuwien.ac.at (P. Skaisgiris); celeste.veronese@univr.it (C. Veronese); daniele.meli@univr.it (D. Meli)
0000-0002-0877-7063 (P. Skaisgiris); 0009-0007-7461-4039 (C. Veronese); 0000-0002-3162-388X (D. Meli)

(CDPI) $\langle e, C\rangle$ adds contextual information via ASP program C; program P accepts a CDPI if some answer set of $P \cup C$ extends e. A learning task $\mathcal{T} = \langle \mathcal{B}, \mathcal{S}_M, E^+, E^-\rangle$ consists of background knowledge $\mathcal{B}$, hypothesis space $\mathcal{S}_M$, and positive/negative examples. A hypothesis $H \subseteq \mathcal{S}_M$ is an *inductive solution* iff: (1) for each $\langle e^+, C\rangle \in E^+$, some answer set of $\mathcal{B} \cup C \cup H$ extends e^+ (*brave*); (2) for each $\langle e^-, C\rangle \in E^-$, no answer set extends e^- (*cautious*). An *optimal solution* minimizes $|H|$. ILASP implements this, returning the shortest hypothesis and reporting uncovered examples from noisy data.

Hypothesis Space ($\mathcal{S}_M$): We encode formulas as parse trees over `label/2` (for propositions) and `edge/2` [10]. Crucially, we add `label/3` to bind actions with preconditions, which is semantically equivalent to the $b?h$ operator under brave semantics. Because our background knowledge lacks choice rules, brave and cautious entailment coincide. Constraints ensure each formula targets a specific action $a \in \mathcal{A}$ and each syntax node possesses exactly one rule body. **Background Knowledge ($\mathcal{B}$):** Following [10, 9], $\mathcal{B}$ includes LTL$_f$ semantics, trace constraints, and ASP range definitions. Additional constraints prevent simultaneous distinct actions (e.g., `left` and `right`) appearing on the same `label` node while permitting multiple groundings of a single schematic action (e.g., `check(X)`). The time variable ensures preconditions and actions are evaluated at the same timestep. **Examples ($E^{\pm}$):** We set $E^- = \emptyset$ as cautious induction for negative examples is overly restrictive [10]. Instead, all traces are modeled as positive examples (E^+): high-return traces must satisfy the formula ($\texttt{sat} \in e^{\text{inc}}$), while low-return traces must not ($\texttt{unsat} \in e^{\text{inc}}$). The context C for an execution λ is the lifted trace $\bigcup_{i=0}^{|\lambda|} \texttt{trace}(i) \cup \bigcup_{\lambda_i \in \lambda} \bigcup_{f \in F_{\mathcal{F}}(\lambda_i) \cup F_{\mathcal{A}}(\lambda_i)} \texttt{trace}(i, f)$ where $F_{\mathcal{F}}$ and $F_{\mathcal{A}}$ are maps from propositions to their ASP features and actions, respectively.

We use the RockSample [12] environment for our experiments. It is a $n \times n$ grid POMDP where a rover samples k rocks of unknown quality. Noisy $check(r)$ actions provide distance-dependent observations. Rewards are $+20$ for valuable rocks, -20 for worthless ones, and $+10$ for exiting the grid. The goal is to maximize cumulative reward via movement, $check$, and $sample$ actions.

ILASP successfully induced several temporal advice rules for the RockSample domain. All induced rules achieved 100% test accuracy (according to a held-out test set) and zero training counterexamples, with the exception of the `sample` action, which had 18 counterexamples. The intuitive meanings of the rules for each action are as follows:

- `check`: 'Never check a rock at a distance greater than 0.'
- `sample`: 'If a rock at a distance > 0 will eventually be checked, then sample a rock now if its goodness probability exceeds 50%.'
- down: 'If a rock at a distance > 0 will eventually be checked, then always move down.'
- `left`: 'It is never the case that you move left until a rock at a distance > 0 is checked.'
- up: 'Move up until it is no longer the case that a rock at a distance > 0 is checked.'
- `right`: 'Move right until it is no longer the case that a rock at a distance > 0 is checked.'

These patterns are consistent with our handcrafted policy, which never checks rocks from a distance, and it is noteworthy that the learned rules encode this behaviour in several, slightly different temporal forms. However, to truly evaluate their effectiveness in guiding agents, these rules must be integrated into a reinforcement learning framework, as demonstrated in the following section.

2.2. Biasing Q-learning exploration and exploitation with temporal advice

Tabular Q-learning is an off-policy, model-free RL algorithm [13] that learns the optimal action-value function $Q^*(s, a)$. Updates are computed from experience tuples (s_t, a_t, r_t, s_{t+1}), which record the current state, action, received reward, and next state. Learning balances *exploitation* (acting "greedily" by choosing actions with the highest Q-value) and *exploration* (randomizing decisions to discover better strategies).

To guide this agent, we construct *advice automata*—symbolic Answer Set Automata (ASA) [8] derived from LTL$_{\text{f}}^{\text{ASP}}$ formulas. Following De Giacomo and Vardi [6], we compile each formula into a DFA while preserving ASP rules as transition guards. Since ASP operators ($b?h, c?h$) are non-temporal, they modify transition evaluation semantics without altering the automaton's structure.

The synthesis follows three steps: **(i)** inducing LTL$_{\text{f}}^{\text{ASP}}$ formulas for each action $a \in \mathcal{A}$ via ILASP; **(ii)**

Algorithm 1 Action weights according to advice automata $\mathcal{M}$

Input: actions A, feature maps $F_{\mathcal{F}}$, $F_{\mathcal{A}}$, automata $\mathcal{M}$, state s_t, discount factor $\gamma_{\mathcal{M}}$

for action a in A **do**
  fluents $\leftarrow F_{\mathcal{F}}(s_t) \cup F_{\mathcal{A}}(a)$
  $D \leftarrow \{\text{next\_dist}(\mathcal{M}_i, \text{fluents}) : \mathcal{M}_i \in \mathcal{M}\}$
  $w_a(D, \gamma, \mathcal{M}) = \begin{cases} 1 - \rho(\mathcal{M}_i) & \text{if } D_i = \infty \\ \rho(\mathcal{M}_i) \cdot (\gamma_{\mathcal{M}})^{D_i} & \text{otherwise} \end{cases}$
  $w_A[a] \leftarrow \sum_i^{|\mathcal{M}|} w_a(D_i, \gamma, \mathcal{M}_i)$
end for
return w_A

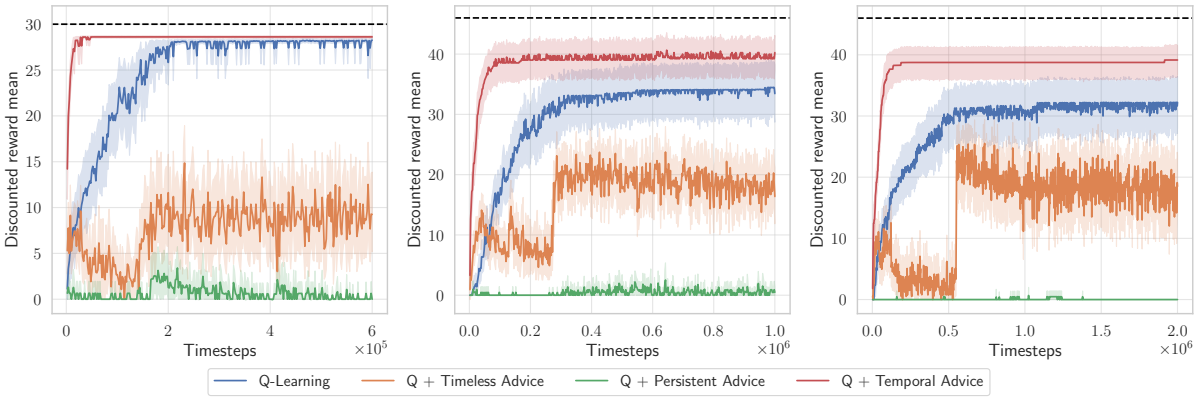

Figure 1: Algorithm evaluation across 5, 10, and 15 grid sizes and advice regimes, averaged over 5 maps and 8 seeds. Dashed line indicates theoretical maximum reward.

selecting the formula per action with the highest test accuracy, which serves as a confidence metric ρ (strength of the particular automaton on the agent); and **(iii)** converting these formulas into ASAs (e.g., using LTLf2DFA [14]). While ASAs are generally non-deterministic due to overlapping features, our setting is effectively deterministic due to our assumption of exactly one action per time step. Then, a transition triggers only if its ASP body holds *and* the agent executes the specific action corresponding to the rule's head, ensuring a unique successor state.

In standard ϵ-greedy policies such as in Q-learning, *exploration* step selects actions uniformly at random. Instead, we would like to bias this step to incorporate the temporal advice. Essentially, we aim to increase the likelihood of actions that progress advice automata toward an accepting state, thereby providing a soft bias that encourages the agent to comply with the advice. Algorithm 1 computes an action probability distribution w_A that reflects which actions most advance advice automata toward acceptance. Weights w_A bias exploration by sampling actions non-uniformly. Automata states are shared across exploration and exploitation and update based on the sampled action and environment state. They reset at episode start or upon acceptance, enabling advice reuse.

During the *exploitation* phase, we scale Q-values by factor k_a (computed from exploration factor ϵ and advice weights $w_a \in w_A$): $k_a = 1 + (\epsilon \cdot w_a)$ if $Q(s, a) \geq 0$, else $k_a = \frac{1}{1+(\epsilon \cdot w_a)}$ as Veronese et al. [11] did. With ϵ decaying linearly to ϵ_f over ϵ_r fraction of total timesteps, the agent prefers actions advancing automata toward acceptance earlier in the learning.

Figure 1 shows that standard Q-learning converges competently across all map sizes. Surprisingly, Q-learning with timeless and persistent advice performs worse than unguided Q-learning, though these advice types improved POMDP solvers in prior work [9, 15]. In contrast, our learned temporal advice outperforms all baselines across all map sizes, with advice from smaller maps generalizing effectively to larger ones. The benefit is most pronounced on larger maps, where unguided Q-learning fails to converge to optimality. This demonstrates that temporal advice learning produces generalizable advice that effectively guides RL agents.

Acknowledgments

This research was funded in part or in full by the Austrian Science Fund (FWF) 10.55776/COE12. The computational results have been achieved using the Austrian Scientific Computing (ASC) infrastructure.

Declaration on Generative AI

During the preparation of this work, the authors used Claude Haiku 4.5 in order to: Grammar and spelling check, Paraphrase and reword. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] K. Y. Rozier, Specification: The Biggest Bottleneck in Formal Methods and Autonomy, in: S. Blazy, M. Chechik (Eds.), Verified Software. Theories, Tools, and Experiments, volume 9971, Springer International Publishing, Cham, 2016, pp. 8–26. doi:10.1007/978-3-319-48869-1_2.

[2] J. Skalse, N. Howe, D. Krasheninnikov, D. Krueger, Defining and characterizing reward gaming, in: S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, A. Oh (Eds.), Advances in Neural Information Processing Systems, volume 35, Curran Associates, Inc., 2022, pp. 9460–9471. URL: https://proceedings.neurips.cc/paper_files/paper/2022/file/3d719fee332caa23d5038b8a90e81796-Paper-Conference.pdf.

[3] R. Toro Icarte, T. Q. Klassen, R. Valenzano, S. A. McIlraith, Reward Machines: Exploiting Reward Function Structure in Reinforcement Learning, Journal of Artificial Intelligence Research 73 (2022) 173–208. doi:10.1613/jair.1.12440.

[4] D. Furelos-Blanco, M. Law, A. Jonsson, K. Broda, A. Russo, Induction and Exploitation of Subgoal Automata for Reinforcement Learning, Journal of Artificial Intelligence Research 70 (2021) 1031–1116. doi:10.1613/jair.1.12372.

[5] G. De Giacomo, V. Fionda, N. Gigante, A. Ielo, F. Ricca, A. Russo, A declarative framework for temporal reasoning in green-aware applications, in: R. Cantini, D. M. Longo, D. Thakur (Eds.), Proceedings of the 1st AIxIA Workshop on Green-Aware Artificial Intelligence co-located with the 23rd International Conference of the Italian Association for Artificial Intelligence (AIxIA 2024), Bolzano, Italy, November 27-28, 2024, volume 3934 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2024, pp. 40–49. URL: https://ceur-ws.org/Vol-3934/paper6.pdf.

[6] G. De Giacomo, M. Y. Vardi, Linear temporal logic and linear dynamic logic on finite traces, in: Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence, Ijcai '13, AAAI Press, Beijing, China, 2013, pp. 854–860.

[7] M. Law, A. Russo, K. Broda, The ILASP System for Inductive Learning of Answer Set Programs (2020).

[8] N. Katzouris, G. Paliouras, Answer Set Automata: A Learnable Pattern Specification Framework for Complex Event Recognition, IOS Press, 2023. URL: http://dx.doi.org/10.3233/FAIA230399. doi:10.3233/faia230399.

[9] D. Meli, A. Castellini, A. Farinelli, Learning Logic Specifications for Policy Guidance in POMDPs: An Inductive Logic Programming Approach, Journal of Artificial Intelligence Research 79 (2024) 725–776. doi:10.1613/jair.1.15826.

[10] A. Ielo, M. Law, V. Fionda, F. Ricca, G. De Giacomo, A. Russo, Towards ILP-based LTLf passive learning, in: E. Bellodi, F. A. Lisi, R. Zese (Eds.), Inductive Logic Programming, Springer Nature Switzerland, Cham, 2023, pp. 30–45.

[11] C. Veronese, A. Farinelli, D. Meli, Sample-efficient neurosymbolic deep reinforcement learning, in: Proceedings of the 25th International Conference on Autonomous Agents and Multiagent Systems, AAMAS '26, International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 2026, p. 3456–3458. URL: https://doi.org/10.65109/RPUM9981. doi:10.65109/RPUM9981.

[12] T. Smith, R. Simmons, Heuristic search value iteration for POMDPs, in: Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence, UAI '04, AUAI Press, Arlington, Virginia, USA, 2004, pp. 520–527.

[13] C. J. C. H. Watkins, Learning from Delayed Rewards, Ph.D. thesis, King's College, University of Cambridge, 1989.

[14] F. Fuggitti, LTLf2DFA, 2019. URL: https://doi.org/10.5281/zenodo.3888410. doi:10.5281/zenodo.3888410.

[15] C. Veronese, D. Meli, A. Farinelli, Learning symbolic persistent macro-actions for pomdp solving over time, in: L. H. Gilpin, E. Giunchiglia, P. Hitzler, E. van Krieken (Eds.), Proceedings of The 19th International Conference on Neurosymbolic Learning and Reasoning, volume 284 of *Proceedings of Machine Learning Research*, PMLR, 2025, pp. 1026–1040. URL: https://proceedings.mlr.press/v284/veronese25a.html.