Reliable Neural Network Controllers for Autonomous Agents in Partially Observable Environments

Nils Jansen[1], **Steven Carr**[2], **Ufuk Topcu**[2]

[1]Radboud University Nijmegen, The Netherlands
[2]The University of Texas at Austin, USA

This abstract builds on results in Carr et al. 2019; 2020; 2021.

POMDPs: Decision-making under uncertainty. Partially observable Markov decision processes (POMDPs) are the standard models for sequential decision-making under uncertainty and incomplete information. An agent that operates in an environment modeled by a POMDP cannot directly assess the state of the system but has only access to observations. By tracking sequences of these observations, an agent can infer the likelihood of the environment (and itself) being in a particular state. A central question is how to effectively represent policies for those agents. Recurrent neural networks (RNNs) process sequential data efficiently via internal memory states, such as those in long short-term memory (LSTM) architectures (Hochreiter and Schmidhuber 1997). Reinforcement learning research has shown that RNNs used in environments modeled by POMDPs perform well as either state or value estimators (Bakker 2001), or as control policies (Hausknecht and Stone 2015).

Rewards are not enough. Safety-critical environments require POMDP policies that are guaranteed to prevent unsafe behavior. The agent's behavior may have to obey more complicated task specifications than maximizing an expected reward, such as reachability, liveness, or, in general, specifications expressed in temporal logic (Pnueli 1977). Such specifications cannot be expressed using the traditional reward shaping techniques (Littman et al. 2017).

Our problem: Learning vs. verification. We state two questions that are central to our approaches.

- For a learned candidate policy, can we efficiently determine whether an agent following this policy adheres to a temporal logic specification?
- How can we learn a suitable candidate policy?

Our approach: Combine learning and verification. We combine the effectiveness of RNN-based representations from machine learning with the provable guarantees that are at the heart of formal verification. In a nutshell, we train RNN-based policy representations from sequences of data, to find candidate policies that might ensure an agent satisfies a temporal logic specification.

The central technical problem is: How to close the loop between training an RNN-based policy and efficiently verifying for a candidate policy? First, so-called finite-state controller (FSCs) (Poupart and Boutilier 2003; Junges et al. 2018) encode memory in a finite automata-style fashion. For an FSC and a POMDP, formal verification methods like model checking are able to efficiently compute the probability of satisfying a specification (Baier and Katoen 2008). We tightly integrate formal verification and machine learning towards three key steps: (1) extracting an FSC from an RNN-based policy, (2) verifying this candidate FSC for the POMDP against a temporal logic specification, and (3) if needed, either refining the FSC or generating more training data for the RNN. For an overview, see Figure 1.

Extracting and verifying the candidate policy. We employ a technique called *quantized bottleneck insertion* (Koul, Fern, and Greydanus 2019) to extract an FSC as candidate policy. An autoencoder (Goodfellow, Bengio, and Courville 2016) discretizes the activation function that is associated with the recurrent hidden node of the RNN. This discretization facilitates a mapping of the continuous memory structure in the RNN to a pre-defined number of memory nodes and transitions of an FSC. Implementing an FSC in a POMDP yields an *induced Markov chain*. For this less complex model, verification methods scale up to billions of states (Baier and Katoen 2008).

Training the policy. We demonstrate how different approaches to generating sequences of training data for the RNN-based policy impact both the computation time and the probability that the agent satisfies the specification. The first method generates sequences of data by following a baseline policy that maximizes the probability of satisfying the specification in while neglecting the partial observability. We compare this approach with one that performs a similar technique, but on a task-aware product POMDP, a larger model created by transforming the specification into an automaton and composing it with the POMDP.

We also show the trade-off between the number of memory nodes in an extracted FSC, and the probability that the agent satisfies the specification. In particular, we empirically demonstrate how increasing the number of memory nodes increases the probability that the induced Markov chain satisfies the temporal logic specification at the cost of longer

Copyright © 2022, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

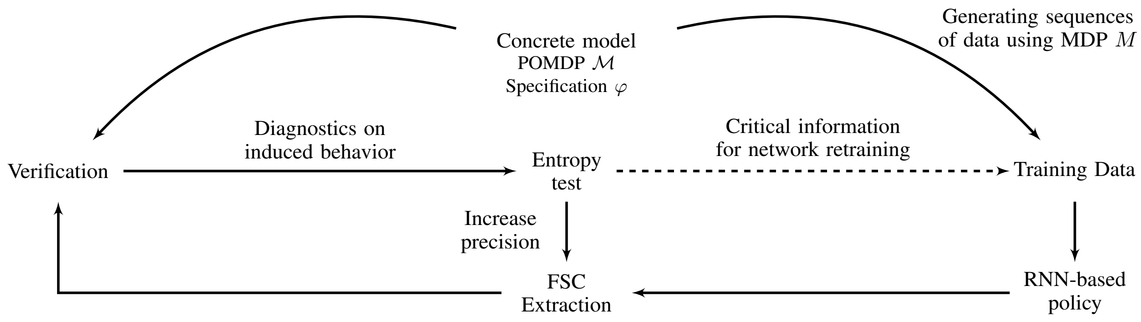

Figure 1: Summary flowchart of the RNN-based refinement loop.

verification times. However, for each FSC, there is a point of diminishing returns after which increasing the number of memory nodes only marginally increases this probability.

Improving the policy. As the central building block to our method, we provide the means to re-train the policy based on data provided by verification. Recall that the RNN-based policy is just a *candidate* policy and may not ensure satisfaction of the specification. If the specification does not hold for the induced Markov chain, the proposed method iteratively improves the extracted policy, as outlined in Figure 1. A by-product of verification is diagnostic information about the states in the POMDP that are critical for the specification, in the form of so-called counterexamples (Wimmer et al. 2014). The method analyzes whether the FSC can be improved, i.e. by either training a *better* RNN-based policy or by extracting a *better* FSC. This analysis relies on examining whether the decisions made in the resulting counterexamples are considered *arbitrary* by measuring the entropy (Cover and Thomas 2012) of the action mapping for the FSC. That is, if the entropy is high across these decisions, the method deems the action mapping of the FSC at those decision-points as arbitrary. Therefore, the RNN-based policy needs to reduce the uncertainty in the action mapping at these critical states by training on more sequences of data. If the entropy is low, then increasing the number of memory nodes in the FSC may help it to approximate the decisions of the RNN-based policy more precisely (Koul, Fern, and Greydanus 2019).

Discussion and Summary. The proposed method computes RNN-based policies and then subsequently extracts candidate policies in the form of an FSC that satisfies temporal logic specifications on a set of POMDP benchmarks with up to millions of states, which is three orders of magnitude larger than comparable approaches. In particular, we benchmark the method against two well-known POMDP solvers from the formal methods (Norman, Parker, and Zou 2017) and planning (Walraven and Spaan 2017) communities. Computing policies that satisfy temporal logic specifications is undecidable for POMDPs (Madani, Hanks, and Condon 1999). Therefore, the proposed method is not *complete*, i.e. it is not guaranteed to find an FSC that ensures an agent in a POMDP satisfies temporal logic specifications. On the other hand, it is *sound*, as in each iteration verification yields provable guarantees on the induced behavior.

Future work will consider extensions to uncertain POMDPs, where probabilities are not given exactly but in form of uncertainty sets like intervals (Suilen et al. 2020; Cubuktepe et al. 2021; Suilen et al. 2022).

References

Baier, C.; and Katoen, J.-P. 2008. *Principles of Model Checking*. MIT Press.

Bakker, B. 2001. Reinforcement Learning with Long Short-Term Memory. In *NIPS*, 1475–1482. MIT Press.

Carr, S.; Jansen, N.; and Topcu, U. 2020. Verifiable RNN-Based Policies for POMDPs Under Temporal Logic Constraints. In *IJCAI*, 4121–4127. IJCAI.org.

Carr, S.; Jansen, N.; and Topcu, U. 2021. Task-Aware Verifiable RNN-Based Policies for Partially Observable Markov Decision Processes. *J. Artif. Intell. Res.*, 72: 819–847.

Carr, S.; Jansen, N.; Wimmer, R.; Serban, A.; Becker, B.; and Topcu, U. 2019. Counterexample-Guided Strategy Improvement for POMDPs Using Recurrent Neural Networks. In *IJCAI*, 5532–5539.

Cover, T. M.; and Thomas, J. A. 2012. *Elements of information theory*. John Wiley & Sons.

Cubuktepe, M.; Jansen, N.; Junges, S.; Marandi, A.; Suilen, M.; and Topcu, U. 2021. Robust Finite-State Controllers for Uncertain POMDPs. In *AAAI*, 11792–11800. AAAI Press.

Goodfellow, I. J.; Bengio, Y.; and Courville, A. C. 2016. *Deep Learning*. Adaptive computation and machine learning. MIT Press.

Hausknecht, M. J.; and Stone, P. 2015. Deep Recurrent Q-Learning for Partially Observable MDPs. In *AAAI*, 29–37. AAAI Press.

Hochreiter, S.; and Schmidhuber, J. 1997. Long short-term memory. *Neural Computation*, 9(8): 1735–1780.

Junges, S.; Jansen, N.; Wimmer, R.; Quatmann, T.; Winterer, L.; Katoen, J.-P.; and Becker, B. 2018. Finite-state controllers of POMDPs via parameter synthesis. In *UAI*, 519–529. AUAI Press.

Koul, A.; Fern, A.; and Greydanus, S. 2019. Learning Finite State Representations of Recurrent Policy Networks. In *ICLR*.

Littman, M. L.; Topcu, U.; Fu, J.; Isbell, C.; Wen, M.; and MacGlashan, J. 2017. Environment-independent task specifications via GLTL. *CoRR*, abs/1704.04341.

Madani, O.; Hanks, S.; and Condon, A. 1999. On the Undecidability of Probabilistic Planning and Infinite-Horizon Partially Observable Markov Decision Problems. In *AAAI*, 541–548. AAAI Press.

Norman, G.; Parker, D.; and Zou, X. 2017. Verification and Control of Partially Observable Probabilistic Systems. *Real-Time Systems*, 53(3): 354–402.

Pnueli, A. 1977. The Temporal Logic of Programs. In *FOCS*, 46–57. IEEE Computer Society.

Poupart, P.; and Boutilier, C. 2003. Bounded Finite State Controllers. In *NIPS*, 823–830. MIT Press.

Suilen, M.; Jansen, N.; Cubuktepe, M.; and Topcu, U. 2020. Robust Policy Synthesis for Uncertain POMDPs via Convex Optimization. In *IJCAI*, 4113–4120. ijcai.org.

Suilen, M.; Simão, T. D.; Parker, D.; and Jansen, N. 2022. Robust Anytime Learning of Markov Decision Processes. In ***NeurIPS***.

Walraven, E.; and Spaan, M. 2017. Accelerated Vector Pruning for Optimal POMDP Solvers. In *AAAI*, 3672–3678. AAAI Press.

Wimmer, R.; Jansen, N.; Ábrahám, E.; Katoen, J.; and Becker, B. 2014. Minimal counterexamples for linear-time probabilistic verification. *Theor. Comput. Sci.*, 549: 61–100.