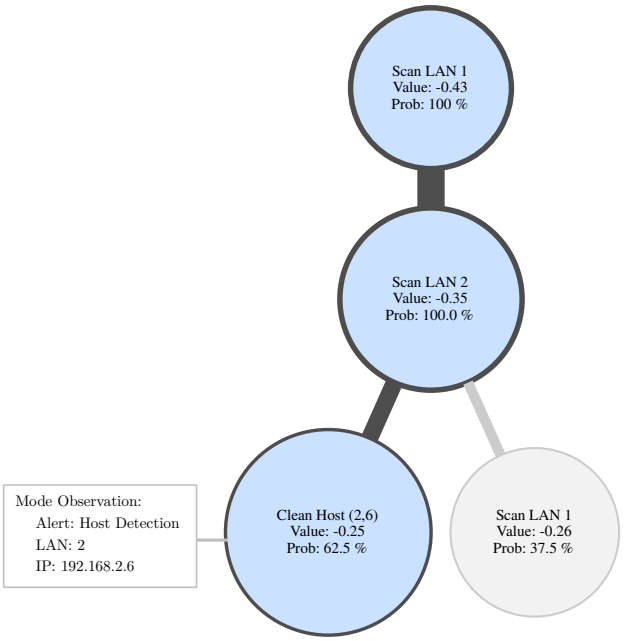

Figure 4: Cyber Security Policy Tree. The figure shows the first three layers of a surrogate tree for the cyber security task. The most frequent observation from the "Clean Host" action node set is shown.

run until all cities have zero susceptible or infected population. The reward at each time step is $r_t = a\sum_i dI_t^{(i)} + b\sum_i dD_t^{(i)}$, where a and b are parameters. We trained a neural network policy using double DQN (**?**) with n-step returns. The trained policy achieved an average score with one standard error bounds of -149.8 ± 0.6 over 100 trial episodes. We built trees for 100 random initial states with 2000 particles and a minimum particle threshold of 250 and tested their performance as forward policies. The trees achieved an average score of -152.2 ± 0.8 over the 100 trials, for an average performance drop of 1.6%. To compare to a baseline surrogate modeling approach, we also trained and tested a LIME model (**?**) with 2000 samples. The LIME average score was -184.2 ± 4.5, for an average performance loss of 23.0%.

The surrogate tree in fig. 2 provides an intuitive understanding of the learned policy. In fig. 2b we can see that the policy does not deploy vaccines to the most heavily infected cities first. It instead prioritizes cities with larger susceptible populations to give the vaccine time to take effect on a larger amount of the population.

Cyber Security

The cyber security task requires an agent to prevent unauthorized access to secure data server on a computer network. The computer network is comprised of four local area networks (LANs), each of which has a local application server and ten workstations, and a single secure data server. The compromise state of the network is not known may be observed through noisy alerts generated from malware scans. Workstations are networked to all others on their LAN and to the LAN application server. Servers are randomly connected in a complete graph. An attacker begins with a single workstation compromised and takes actions to compromise additional workstations and servers to reach the data server. The defender can scan all nodes on a LAN to locate compromised nodes with probability p_{detect}. Compromised nodes will also generate alerts without being scanned with low probability. The defender can also scan and clean individual nodes to detect and remove compromise. The reward is zero unless the data server is compromised, in which case a large penalty is incurred. The defender was trained using Rainbow DQN (**?**).

Automated systems such as this are often implemented with a human in the loop. Policies that can be more easily interpreted are more likely to be trusted by a human operator. A surrogate tree for the neural network is shown in fig. 4. Unlike the baseline neural network, the policy encoded by this tree can be easily interpreted. The agent will continually scan LAN 1 in most cases, and will only clean a workstation after malware has been detected.

Conclusions

In this work, we presented methods to construct local surrogate policy trees from arbitrary control policies. The trees are more interpretable than high-dimensional policies such as neural networks and provide quantitative estimates of future behavior. Our experiments show that, despite truncating the set of actions that may be taken at each future time step, the trees retain fidelity with their baseline policies. Experiments demonstrate the effect of various environment and algorithm parameters on tree size and fidelity in a simple grid world. Demonstrations show how surrogate trees may be used in more complex, real-world scenarios. The action node clustering presented in this work used a heuristic search method that provided good results, but without any optimality guarantees. Future work will look at improved approaches to clustering, for example by using a mixed integer program optimization. We will also explore using the scenarios simulated to construct the tree to backup more accurate value estimates, and refine the resulting policy. Including empirical backups such as these may also allow calculation of confidence intervals or bounds on policy performance (**?**).

Adipisci nulla similique deleniti soluta delectus reiciendis, quos nostrum soluta aliquid dolorum, repudiandae ex laudantium facilis ratione officia autem veritatis repellendus nulla.Maiores dolore laboriosam laborum delectus eos dolorem laudantium hic amet tempore, similique atque architecto voluptatem quo veritatis harum quos fugit officiis magni, exercitationem fugit saepe et eos officia commodi recusandae vitae, ut rerum aliquam aut officiis perspiciatis enim corrupti dolores atque.Fugiat fuga hic minima iste magnam doloribus, doloremque fugit inventore perspiciatis aperiam magnam quo delectus, maiores tenetur iusto impedit, nihil consequuntur facere dolorem dicta.Voluptatibus a totam eos voluptatum, soluta corporis rem facere sapiente animi veritatis, eum pariatur dolore ut tempora voluptas odit, recusandae aut dolor sunt odio illo consequuntur ea, pos-

simus assumenda facilis repellat ut aliquid nobis adipisci debitis quo et?