Supplementary Material For Optimization Methods for Interpretable Differentiable Decision Trees Applied to Reinforcement Learning

Author 1
Institution 1

Author 2
Institution 2

Author 3
Institution 3

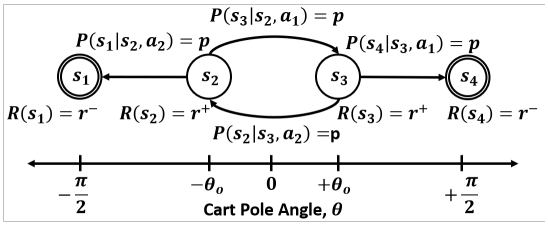


Figure 1

Appendix A: Derivation of the Optimal Policy

In this section, we provide a derivation of the optimal policy for the MDP in Fig. 1. For this derivation, we use the definition of the Q-function described in Eq. 2, where s' is the state resulting from applying action a in state s. In keeping with the investigation in this paper, we assume deterministic transitions between states (i.e., $p = 1$ from Eq. 1). As such, we can ignore $P(s'|s,a)$ and simply apply Eq. 3.

$$P(s'|s,a) = \begin{cases} 0, & \text{if } s' \in 1,4 \\ p, & \text{if } (s' = s+1, a = a_1) \vee (s' = s-1, a = a_2) \\ \frac{1-p}{|S|-1}, & \text{otherwise} \end{cases} \tag{1}$$

$$Q(s,a) := R(s,a) + \gamma \max_{a'} \sum_{s'} P(s'|s,a) Q(s',a') \tag{2}$$

$$Q(s,a) := R(s,a) + \gamma \max_{a'} Q(s',a') \tag{3}$$

Theorem 1 *The optimal policy for the MDP in Fig. 1 is to apply action a_1 in state s_2 and action a_2 in state s_3 assuming deterministic transitions between states (i.e., $p = 1$ from Eq. 1).*

We begin by asserting in Eq. 4 that the Q-values for $Q(s,a)$ are r^- given $s \in \{1,4\}$ and for any action $a \in \{a_1, a_2\}$. This result is due to the definition that states s_1 and s_4 are terminal states and the reward for those states is r^- regardless of the action applied. We note that, in our example, $r^- = 0$, but we leave it here for the sake of generality.

$$Q(s_1,a_1) = Q(s_1,a_2) = Q(s_4,a_1) = Q(s_4,a_2) = r^- \tag{4}$$

Next, we must compute the Q-values for the remaining state-action pairs, as shown in Eq. 5-8.

$$Q(s_2,a_1) = R(s_2,a_1) + \gamma \max\{Q(s_3,a_1), Q(s_3,a_2)\} \tag{5}$$

$$Q(s_2,a_2) = R(s_2,a_1) + \gamma \max\{Q(s_1,a_1), Q(s_1,a_2)\} \tag{6}$$

$$Q(s_3,a_1) = R(s_3,a_1) + \gamma \max\{Q(s_4,a_1), Q(s_4,a_2)\} \tag{7}$$

$$Q(s_3,a_2) = R(s_3,a_2) + \gamma \max\{Q(s_2,a_1), Q(s_2,a_2)\} \tag{8}$$

By the definition of the MDP in Fig. 1, we substitute in for $R(s_2,a_1) = R(s_2,a_2) = R(s_3,a_1) = R(s_3,a_2) = r^+$ as shown in Eq. 9-12.

$$Q(s_2,a_1) = r^+ + \gamma \max\{Q(s_3,a_1), Q(s_3,a_2)\} \tag{9}$$

$$Q(s_2,a_2) = r^+ + \gamma r^- \tag{10}$$

$$Q(s_3,a_1) = r^+ + \gamma r^- \tag{11}$$

$$Q(s_3,a_2) = r^+ + \gamma \max\{Q(s_2,a_1), Q(s_2,a_2)\} \tag{12}$$

We can substitute in for $Q(s_3,a_1)$ and $Q(s_2,a_2)$ given Eq. 9 and 12.

$$Q(s_2,a_1) = r^+ + \gamma \max\{\left(r^+ + \gamma r^-\right), Q(s_3,a_2)\} \tag{13}$$

$$Q(s_3,a_2) = r^+ + \gamma \max\{Q(s_2,a_1), \left(r^+ + \gamma r^-\right)\} \tag{14}$$

For the Q-value of state-action pair, $Q(s_2,a_1)$, we must determine whether $(r^+ + \gamma r^-)$ is less than or equal to $Q(s_3,a_2)$. If the agent were to apply action a_2 in state s_3, we can see from Eq. 14 that the agent would receive at a minimum $Q(s_3,a_2) \geq r^+ + \gamma\left(r^+ + \gamma r^-\right)$, because $r^+ + \gamma\left(r^+ + \gamma r^-\right) > r^+ + \gamma r^-$, $Q(s_3,a_2)$ must be the maximum from Eq. 13. We can make a symmetric argument

for $Q(s_3, a_2)$ in Eq. 14. Given this relation, we arrive at Eq. 15 and 16.

$$Q(s_2, a_1) = r^+ + \gamma Q(s_3, a_2) \tag{15}$$

$$Q(s_3, a_2) = r^+ + \gamma Q(s_2, a_1) \tag{16}$$

Eq. 15 and 16 represent a recursive, infinite geometric series, as depicted in Eq. 18.

$$\begin{aligned} Q(s_2, a_1) = Q(s_3, a_2) &= r^+ + \gamma r^+ + \gamma^2 r^+ + \dots \\ &= r^+ \left(\gamma^0 + \gamma + \gamma^2 + \dots\right) \quad (17) \\ &= r^+ \sum_{t=0}^{T} \gamma^t \quad (18) \end{aligned}$$

In the case that $T = \infty$, Eq. 18 represents an infinite geometric series, the solution to which is $\frac{r^+}{1+\gamma}$. In our case however, $T = 3$ (i.e., four-time steps). As such, $Q(s_2, a_1) = Q(s_3, a_2) = r^+(1 + \gamma + \gamma^2 + \gamma^3)$, as shown in Eq. 19.

$$Q(s_2, a_1) = Q(s_3, a_2) = r^+(1 + \gamma + \gamma^2 + \gamma^3) \tag{19}$$

Recall that $r^- < 0$ given our definition of the MDP in Fig. 1. Therefore, $Q(s_2, a_1) = Q(s_3, a_2) = \frac{r^+}{1-\gamma} \geq Q(s_2, a_2) = Q(s_3, a_1) = r^+ + \gamma r^-$. If the RL agent is non-myopic, i.e., $\gamma \in (0, 1]$, then we have the strict inequality $Q(s_2, a_1) = Q(s_3, a_2) > Q(s_2, a_2) = Q(s_3, a_1)$. For these non-trivial settings of γ, we can see that the optimal policy for the RL agent is to apply action a_1 in state s_2 and action a_2 in state s_3. Lastly, because s_1 and s_4 are terminal states, the choice of action is irrelevant, as seen in Eq. 4.

The optimal policy is then given by Eq. 20.

$$\pi^*(s, a) = \begin{cases} 1, & \text{if } s = 2, a_1 \text{ or } s = 3, a_2 \\ 0, & \text{if } s = 2, a_2 \text{ or } s = 3, a_1 \\ 1/2, & \text{otherwise} \end{cases} \tag{20}$$

Appendix B: Policy Traces and Value

This section reports the execution traces and corresponding value calculations of a Boolean decision treee with varying ϕ for the simple MDP model from Figure 1.

Appendix C: Q-learning Leaf Values

For the decision tree in Fig. 2, there are four leaf values: $\hat{y}_{a_2}^{TRUE}$, $\hat{y}_{a_1}^{TRUE}$, $\hat{y}_{a_2}^{FALSE}$, and $\hat{y}_{a_1}^{FALSE}$. Table 3 contains the settings of those parameters. In Table 3, the first column depicts the leaf parameters; the second column depicts the Q-function state-action pair; the third column contains the equation reference to Appendix A, where the Q-value is calculated; and the fourth column contains the corresponding Q-value. These Q-values assume that the agent begins in a non-terminal state (i.e., s_2 or s_3) and follows the optimal policy represented by Eq. 20.

Table 1: The set of execution traces for a Boolean decision tree with varying ϕ, assuming $s_o = 3$. Columns indicate increasing time, rows indicate settings for ϕ, and entries indicate $(s_t, R(s_t, a_t), a_t)$.

ϕ	$t=0$	$t=1$	$t=2$	$t=3$
0	$(3, r^+, 2)$	$(2, r^+, 2)$	$(1, r^-, 2)$	
1	$(3, r^+, 2)$	$(2, r^+, 2)$	$(1, r^-, 1)$	
2	$(3, r^+, 2)$	$(2, r^+, 1)$	$(3, r^+, 2)$	$(2, r^+, 1)$
3	$(3, r^+, 1)$	$(4, r^-, 2)$		
4	$(3, r^+, 1)$	$(4, r^-, 1)$		

Table 2: Derived from Table 1, the values V^{π_ϕ} of Boolean decision tree policies π_ϕ with varying ϕ and assuming $s_o = 3$.

ϕ	$\gamma^t r^t$				$V^{\pi_\phi}(s_3)$
	$t=0$	$t=1$	$t=2$	$t=3$	
0	r^+	$r^+\gamma$	$r^-\gamma^2$		$r^+(1+\gamma)+r^-$
1	r^+	$r^+\gamma$	$r^-\gamma^2$		$r^+(1+\gamma)+r^-$
2	r^+	$r^+\gamma$	$r^+\gamma^2$	$r^+\gamma^3$	$r^+(1+\gamma+\gamma^2+\gamma^3)$
3	r^+	$r^-\gamma$			$r^+ + r^-\gamma$
4	r^+	$r^-\gamma$			$r^+ + r^-\gamma$

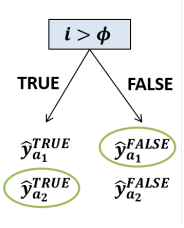


Figure 2: This figure depicts the tree for our case study.

Table 3: Derived from Table 1, the values V^{π_ϕ} of Boolean decision tree policies π_ϕ with varying ϕ and assuming $s_o = 3$.

Leaf	Q-function	Eq.	Q-value
$\hat{y}_{a_2}^{FALSE}$	$Q(s_2, a_2)$	Eq. 10	$r^+ + \gamma r^-$
$\hat{y}_{a_1}^{TRUE}$	$Q(s_3, a_1)$	Eq. 11	$r^+ + \gamma r^-$
$\hat{y}_{a_1}^{FALSE}$	$Q(s_2, a_1)$	Eq. 19	$r^+(1+\gamma+\gamma^2+\gamma^3)$
$\hat{y}_{a_2}^{TRUE}$	$Q(s_3, a_2)$	Eq. 19	$r^+(1+\gamma+\gamma^2+\gamma^3)$

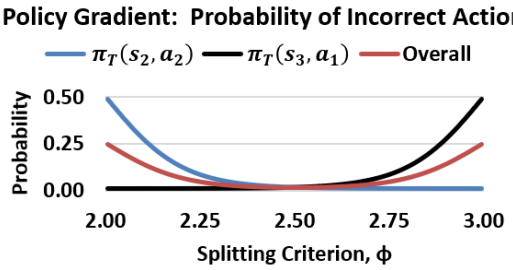


Figure 3: This figure demonstrates the probability of taking a "wrong" action for PG with $\gamma = 0.95$, $a = 10$, $r^+ = 1$, and $r^- = -1$.

Appendix D: Probability of Incorrect Action

$$\pi_T(s, a) = \mu(s)\hat{y}_a^{\text{TRUE}} + (1 - \mu(s))\,\hat{y}_a^{\text{FALSE}} \tag{21}$$

The output of the differentiable tree is a weighted, nonlinear combination of the leaves (Eq. 21). Using PG, one samples actions probabilistically from $\pi_T(s, a)$. The probability of applying the "wrong" action (i.e., one resulting in a negative reward) is $\pi_T(s_3, a_1)$ in state s_3 and $\pi_T(s_2, a_2)$ in state s_2. Assuming it equally likely to be in states s_3 and s_2, the overall probability is $\frac{1}{2}\left(\pi_T(s_2, a_2) + \pi_T(s_3, a_1)\right)$. These probabilities are depicted in Fig. 3, which shows how the optimal setting, ϕ^*, for ϕ should be $\phi^* = 2.5$ using PG.

Appendix E: Architecture Sweeps

We performed architecture sweeps, as mentioned in the main paper, across all types of models. We found that the MLP requires small models for simple domains, the DDT methods are all relatively unaffected by increased depth, representing a benefit of applying DDTs to various RL tasks. For this result, see Figure 4. As shown in Figure 5, in more complex domains, the results are less conclusive and increased depth does not show clear trends for any approach. Nonetheless, we show evidence that DDTs are at least competitive with MLPs for RL tasks of varying complexity, and that they are more robust to hyperparameter tuning with respect to depth and number of layers.

We find that the MLP with no hidden layers performs the best on the two OpenAI Gym domains, cart pole and lunar lander. The best differentiable decision tree architectures for the cart pole domain are those with two leaves and two rules, while the best architectures for lunar lander include 32 leaves and 16 rules.

In the wildfire tracking domain, the 8-layer MLP performed the best of the MLPs, while the 32-leaf differentiable decision tree was the top differentiable decision tree, and the 32-rule differentiable rule list performed the best of the differentiable rule lists.

Finally, the MLP in the FindAndDefeatZerglings domain is an 8-layer MLP, and the differentiable decision tree uses 8 leaves while the differentiable rule list uses 8 rules.

MLP hidden layer sizes preserve the input data dimension through all hidden layers until finally downsampling to the action space for the final layer. MLP networks all use the ReLU activation after it performed best in a hyperparameter sweep.

Appendix F: Domain Details

0.1 Wildfire Tracking

The wildfire tracking domain input space is:

- Fire 1 Distance North (float)
- Fire 1 Distance West (float)
- Closest To Fire 1 (Boolean)
- Fire 2 Distance North (float)
- Fire 2 Distance West (float)
- Closest To Fire 2 (Boolean)

Distance features are floats, representing how far north or west the fire is, relative to the drone. Distances can also be negative, implying that the fire is south or east of the drone.

0.2 StarCraft II Micro-battle Evaluation

The FindAndDefeatZerglings manufactured input space is:

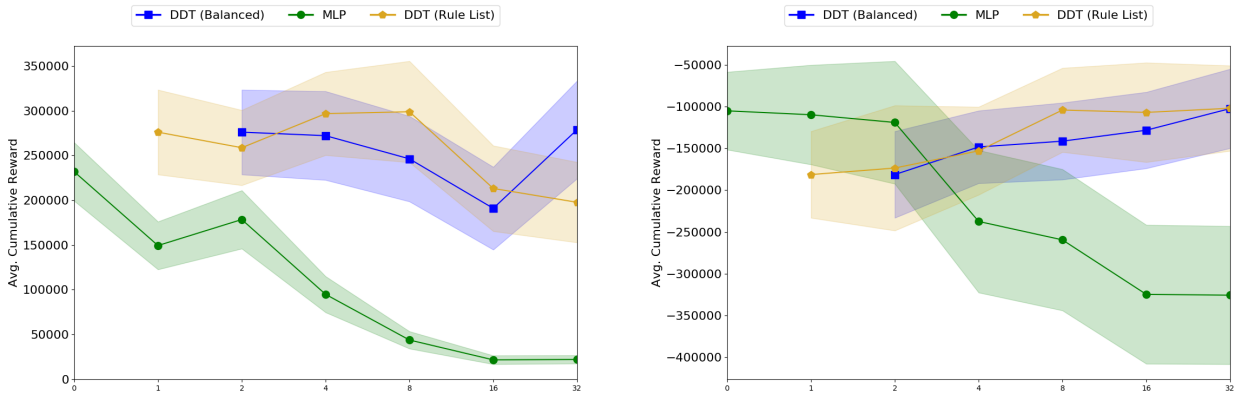
- X Distance Away (float)
- Y Distance Away (float)
- Percent Health Remaining (float)
- Percent Weapon Cooldown Remaining (float)

for each agent-controlled unit and 2 allied units, as well as:

- X Distance Away (float)
- Y Distance Away (float)
- Percent Health Remaining (float)
- Percent Weapon Cooldown Remaining (float)
- Enemy Unit Type (Boolean)

for the five nearest enemy units. Missing data is filled in with -1. The action space for this domain consists of:

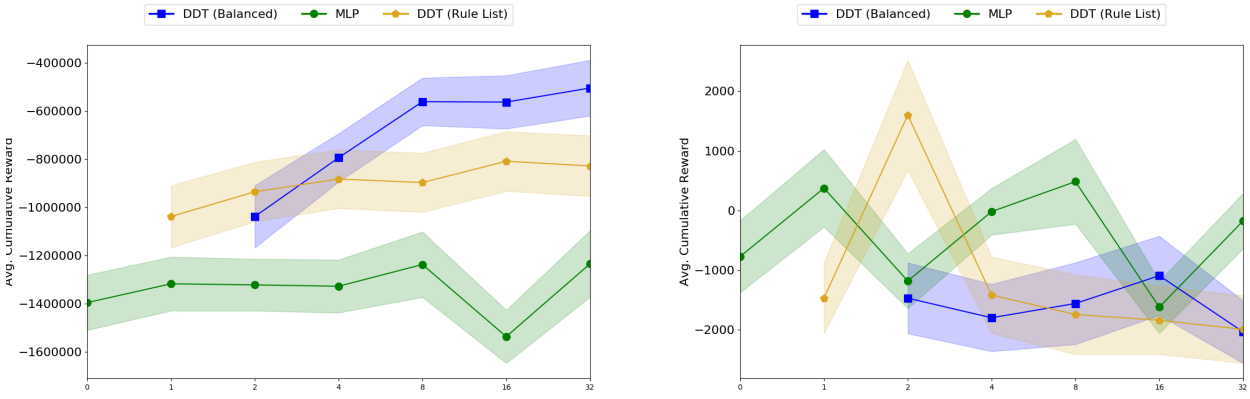
- Move North
- Move East
- Move West
- Move South
- Attack Nearest Enemy
- Attack Second Nearest Enemy
- Attack Third Nearest Enemy
- Attack Second Farthest Enemy
- Attack Farthest Enemy
- Do Nothing



(a) Cart Pole

(b) Lunar Lander

Figure 4: Average cumulative reward and standard deviation across architectures of various sizes in the Gym domains. MLP with number of hidden layers, DDT (Rule List) with number of rules, and DDT (Balanced) with number of leaves.



(a) Wildfire Tracking

(b) FindAndDefeatZerglings

Figure 5: Average cumulative reward and standard deviation across architectures of various sizes in the wildfire and SC2 domains. MLP with number of hidden layers, DDT (Rule List) with number of rules, and DDT (Balanced) with number of leaves.

Interpretable Policies

Here we include interpretable policies for each domain, without the pruning that is included in versions in the main body. See Figures 6, 7, 8, and 9. Finally, we also include examples of two MLPs represented as decision-making aids. The first is the one-hot MLP that was given to study participants for evaluation of interpretability and efficiency, shown in Figure 10. The second is the true cart pole MLP, available in Figure 11. This decision-making aid turned out to be exceptionally complicated, even with no activation functions and no hidden layer.

Sample Survey Questions

Survey questions included Likert scale questions ranging from 1 (Very Strongly Disagree) to 7 (Very Strongly Agree). For both the MLP and decision trees, some questions included:

1. I understand the behavior represented within the model.
2. The decision-making process does not make sense.
3. The model's logic is easy to follow
4. I like the level of readability of this model.
5. The model is difficult to understand.

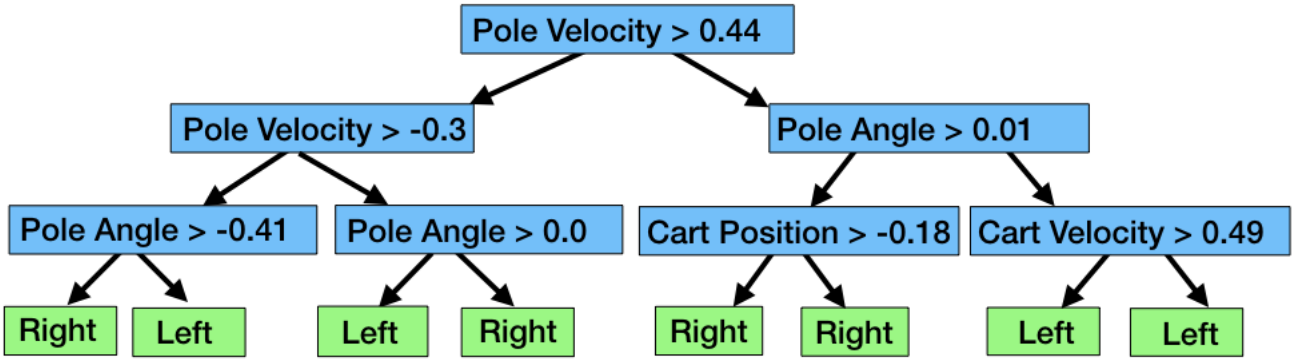


Figure 6: Full interpretable cart pole policy. Two decision nodes are redundant, leading to the same action regardless of how the node is evaluated.

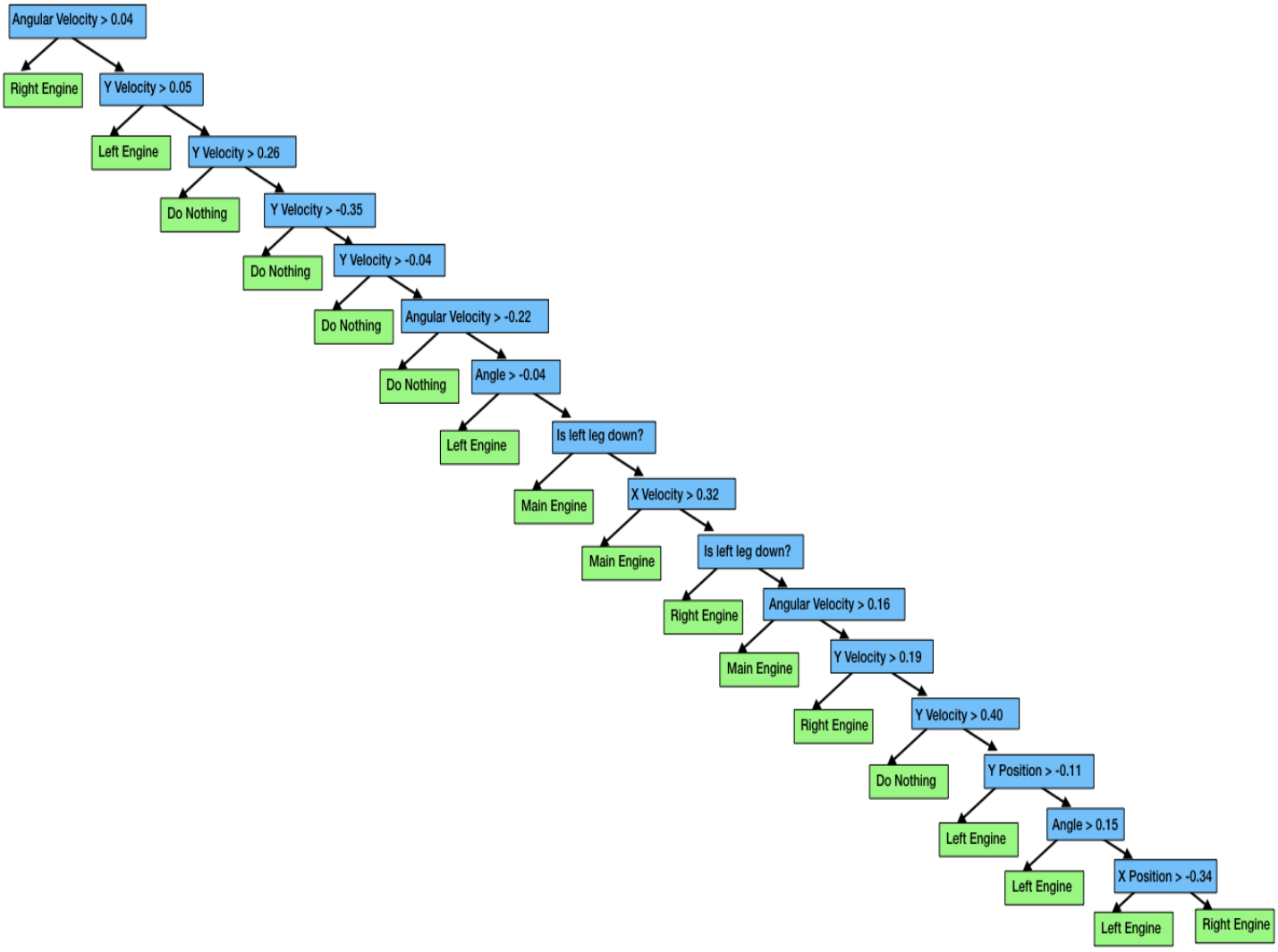


Figure 7: Full interpretable lunar lander rule list policy. Many nodes in the list are not reachable due to previous nodes.

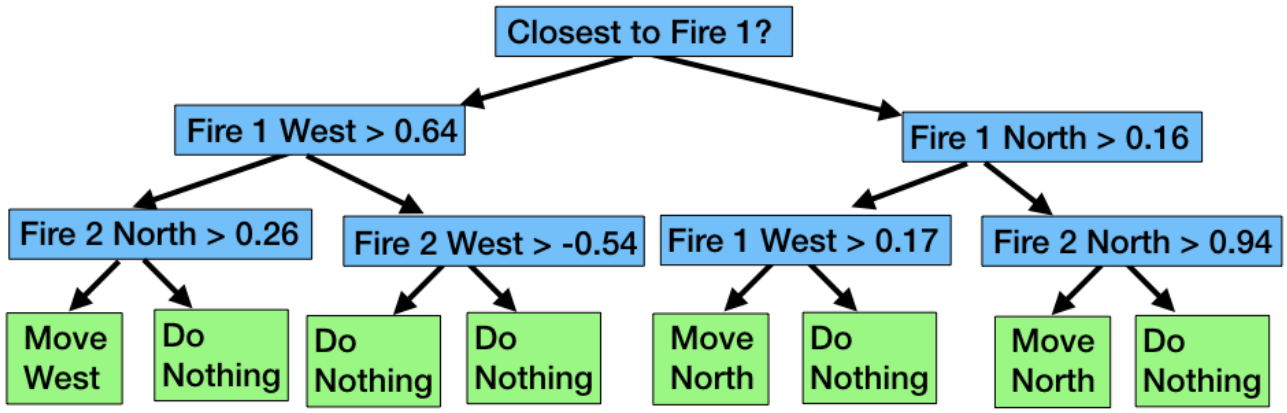


Figure 8: Full interpretable wildfire tracking policy. One node is redundant, leading to the same action regardless of how it is evaluated.

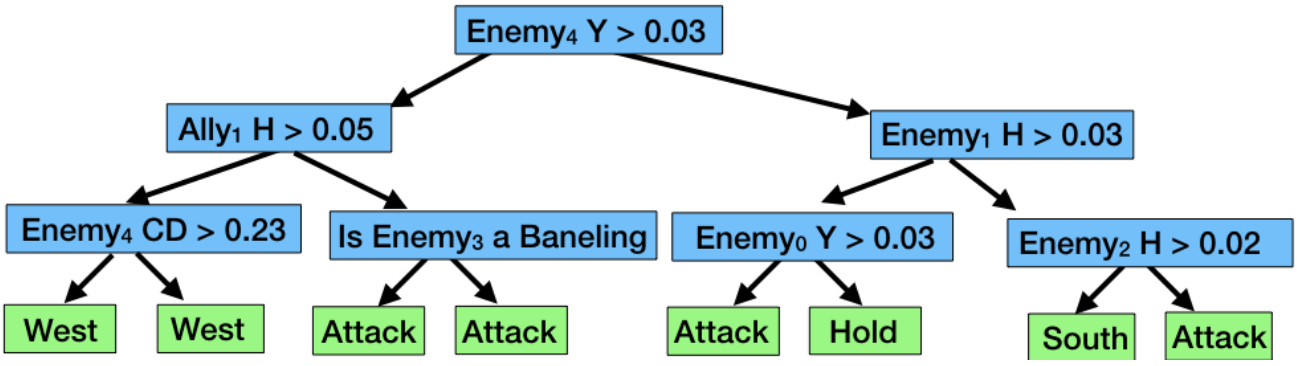


Figure 9: Full interpretable FindAndDefeatZerglings policy. One node is redundant, leading to the same action regardless of how it is evaluated.

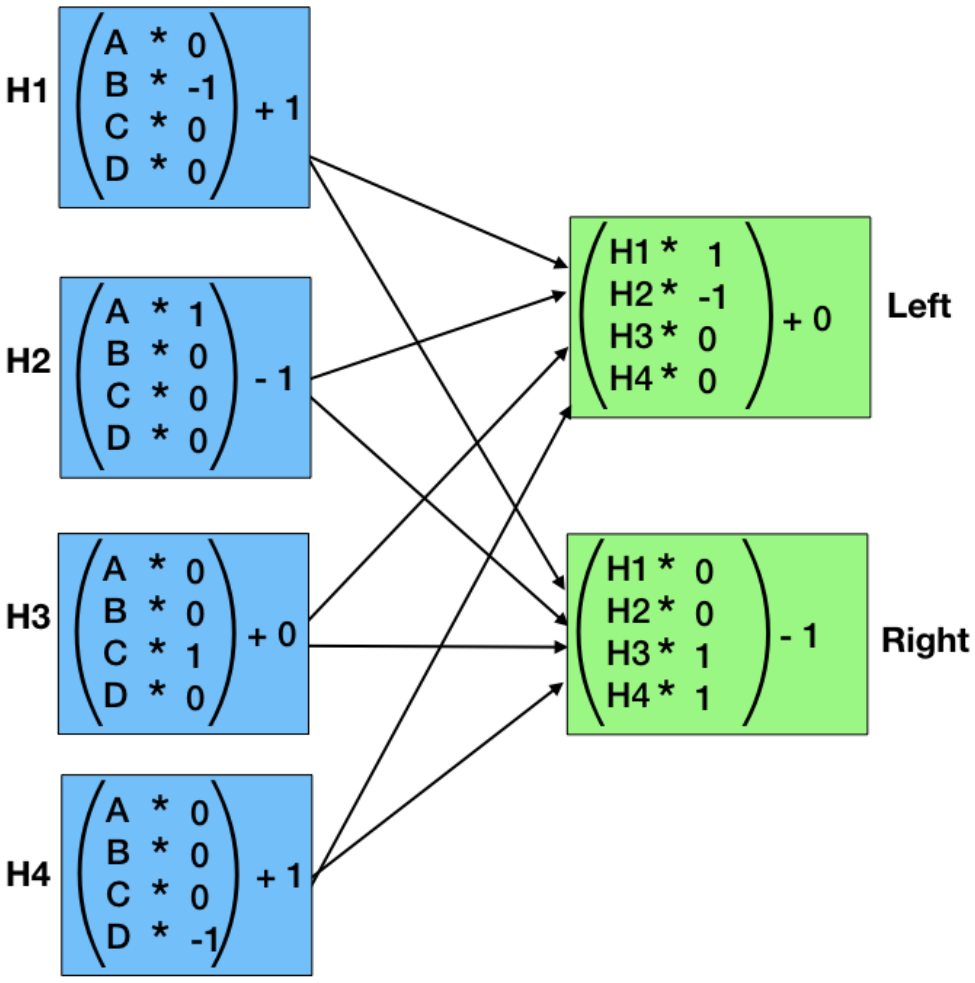


Figure 10: The MLP given to participants for our user study.

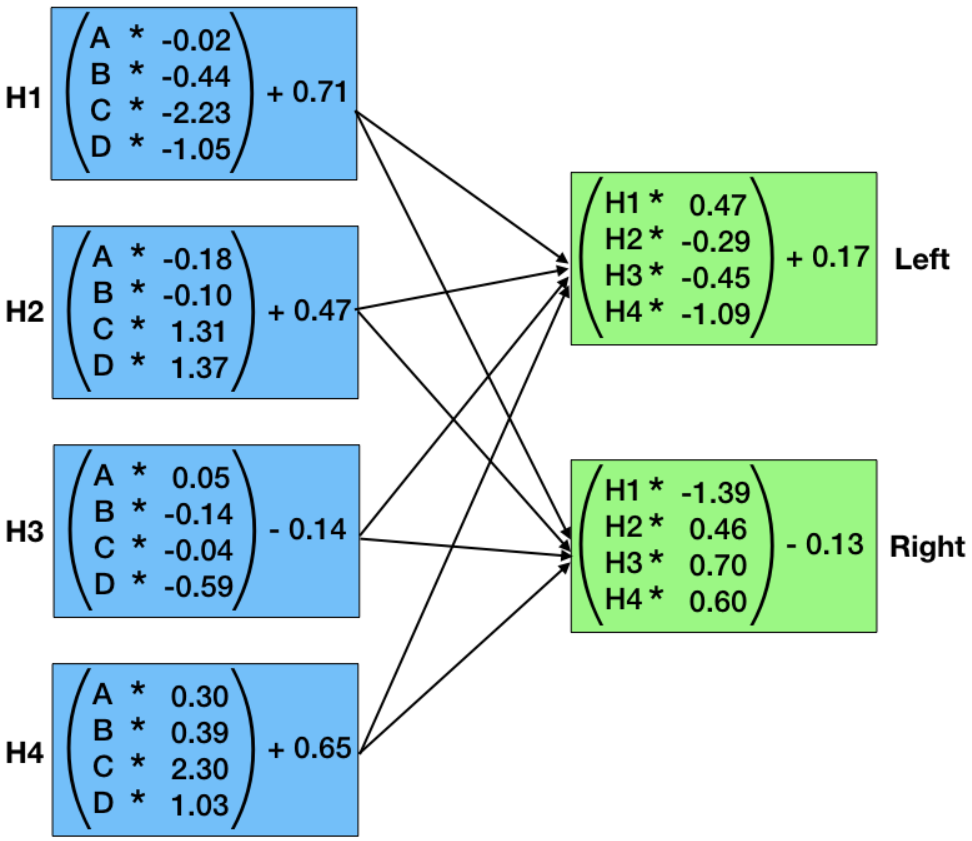


Figure 11: The actual MLP originally intended to go into the user study. Note that it is markedly more complicated than the version given to participants.