Learning Interpretable, High-Performing Policies for Continuous Control

Rohan Paleja*, Yaru Niu*, Andrew Silva, Chace Ritchie, Sugju Choi, and Matthew Gombolay
rohan.paleja@gatech.edu, yarun@andrew.cmu.edu, matthew.gombolay@gatech.edu
Georgia Tech | Robotics & Intelligent Machines | CORE Robotics Lab

Paper | Code

Introduction

Gradient-based approaches in reinforcement learning (RL) have achieved tremendous success in learning policies for continuous control problems. While the performance of these approaches enables real-world adoption, these policies lack interpretability, limiting deployability in the safety-critical and legally-regulated domains like autonomous driving.

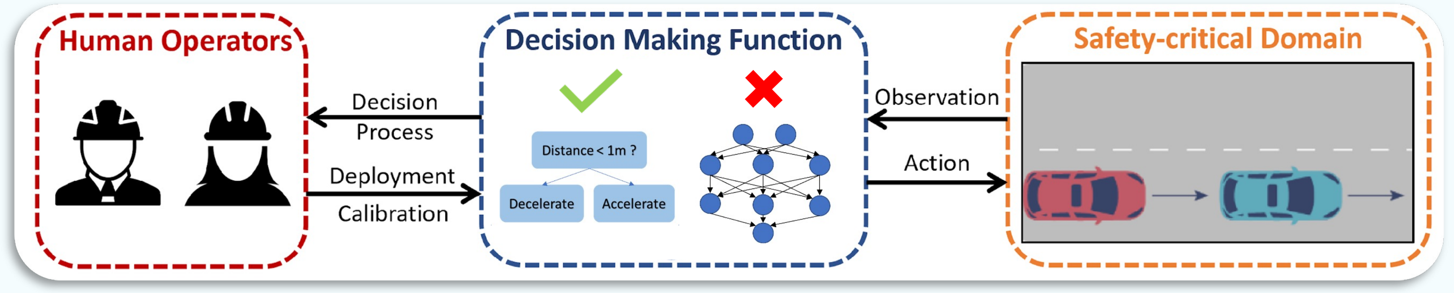


Figure 1. An autonomous vehicle control pipeline. The use of black-box models as a decision-making function does not permit inspection or verification by human operators. It is instead better to utilize white-box approaches that permit insight into a decision-making model.

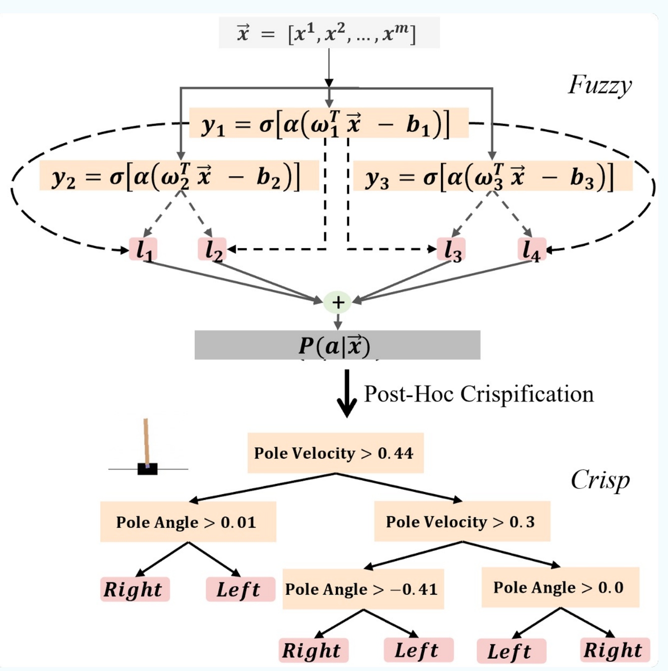
- Interpretable policies can support situational awareness[1], build trust[2], and ensure safety[3]
- In safety-critical and legally-regulated domains, insight into a machine's decision-making process is of utmost importance.

❖ **The ability to optimize sparse logical models, such as decision trees, is one of 10 grand challenges in interpretable machine learning[4].**

Preliminaries

DDTs[5,6,7] models have been trained for discrete action spaces in the past

- Via supervised learning by Suarez and Lutzko and Paleja et al,
- Via Reinforcement learning by Silva et al



Once DDT parameters have been inferred, prior work, applied a post-hoc crispification[6,7] consisting of several argument max operations to produce an interpretable model.

- DDT models in the past have been unable to handle continuous action-spaces.
- Interpretable models generated via post-hoc crispification are not representative of the model learned via Reinforcement Learning

References

[1] Paleja, R.R., Ghuy, M., Arachchige, N.R., Jensen, R., & Gombolay, M.C. The Utility of Explainable AI in Ad Hoc Human-Machine Teaming. NeurIPS 2021.
[2] Bhatt, U., Ravikumar, P., & Moura, J.M. Building Human-Machine Trust via Interpretability. AAAI 2019.
[3] Vasic, M., Petrović, A., Wang, K., Nikolic, M., Singh, R., & Khurshid, S. MoËT: Interpretable and Verifiable Reinforcement Learning via Mixture of Expert Trees. arXiv, 2019.
[4] Rudin, C., Chen, C., Chen, Z., Huang, H., Semenova, L., & Zhong, C. Interpretable Machine Learning: Fundamental Principles and 10 Grand Challenges. arXiv, 2021.
[5] Suárez, A., & Lutsko, J.F. Globally Optimal Fuzzy Decision Trees for Classification and Regression. IEEE TPAMI, 1999.
[6] Silva, A., Gombolay, M.C., Killian, T.W., Jimenez, I.D., & Son, S. Optimization Methods for Interpretable Differentiable Decision Trees Applied to Reinforcement Learning. AISTATS 2020.
[7] Paleja, R., Silva, A., Chen, L., & Gombolay, M. Interpretable and personalized apprenticeship scheduling: Learning interpretable scheduling policies from heterogeneous user demonstrations. NeurIPS 2020.

Interpretable Continuous Control Trees (ICCTs)

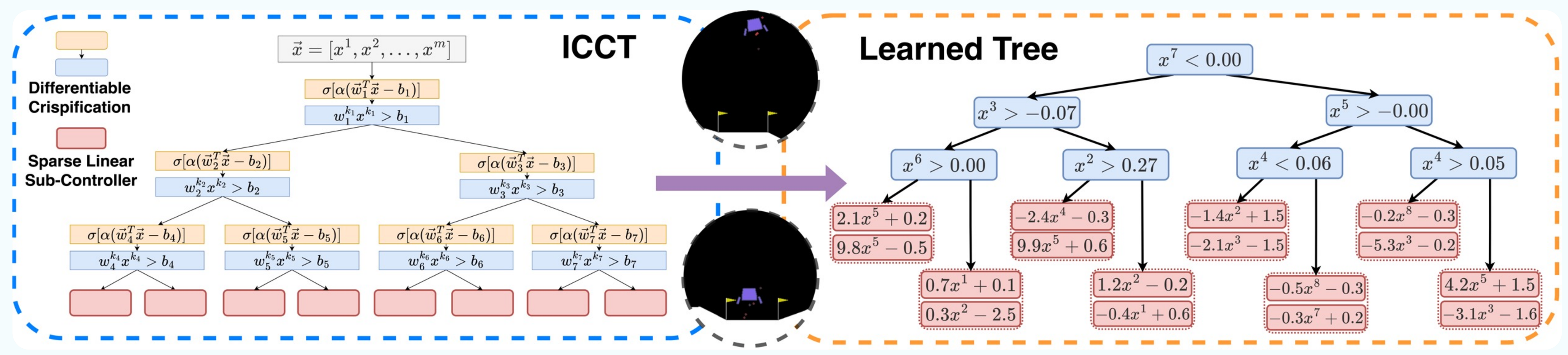


Figure 2. A depiction of our ICCT in its form during training (left) alongside its conversion to an interpretable form (right).

- ICCTs
  - Extend DDTs to continuous action-spaces by maintaining sparse linear controllers at the leaves. Sparsity can be tuned to go from a simple tree with scalar leaves to a multivariate linear controller.
  - Utilize a novel differentiable crispification mechanism directly optimize over a sparse decision-tree representation. Due to this, the interpretable model is consistent with the model being trained.

ICCT Key Elements

Decision Node Crispification

Translates a fuzzy decision node to be conditioned upon a single feature.

$$\text{Node\_Crisp}(\sigma(\alpha(\vec{w}_i^T\vec{x} - b_i))) \rightarrow \sigma(\alpha(w_i^k x^k - b_i))$$

Decision Outcome Crispification

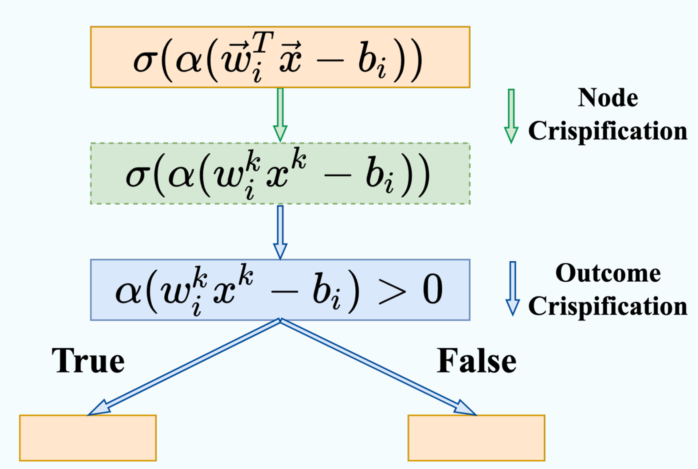
Translates the intermediate decision node representation to output a Boolean.

$$\text{Outcome\_Crisp}(\sigma(\alpha(w_i^k x^k - b_i))) \rightarrow \mathbb{1}(\alpha(w_i^k x^k - b_i) > 0)$$

Sparse Linear Leaf Controllers

Translates leaf nodes to condition upon only e features.

$$l_d^* \triangleq (\vec{u}_d \circ \vec{\beta}_d)^T(\vec{u}_d \circ \vec{x}) + \vec{u}_d^T\vec{\phi}_d$$



Algorithm 4 Differentiable Argument Max Function for Crispification: DIFF_ARGMAX$(\cdot)$

Input: Logits $\vec{q}$
Output: One-Hot Vector $\vec{h}$

1: $\vec{h}_{soft} \leftarrow f(\vec{q})$ — $f(\cdot)$ is a Softmax function
2: $\vec{h}_{hard} \leftarrow \text{ONE\_HOT}(\text{ARGMAX}(f(\vec{q})))$ (step 1 for $g(\cdot)$)
3: $\vec{h} = \vec{h}_{hard} + \vec{h}_{soft} - \text{STOP\_GRAD}(\vec{h}_{soft})$ (step 2 for $g(\cdot)$)

Algorithm to maintain gradients in gradient-diminishing operations via the straight-through trick

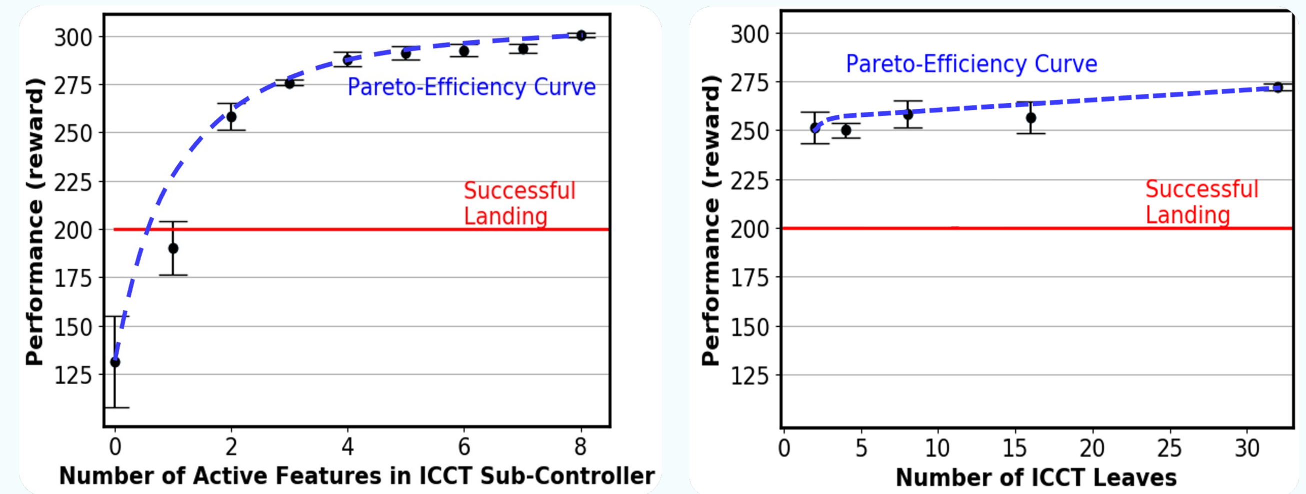
Results

Worst to Best:

Method	Common Continuous Control Problems		Autonomous Driving Problems			
	Inverted Pendulum	Lunar Lander	Lane Keeping	Single-Lane Ring	Multi-Lane Ring	Figure-8
DT	155.0 ± 0.9 256 leaves (766 params)	-285.5 ± 15.6 256 leaves (1022 params)	-359.0 ± 11.0 256 leaves (766 params)	$\mathbf{123.2 \pm 0.03}$ 32 leaves (94 params)	503.2 ± 24.8 256 leaves (1022 params)	831.1 ± 1.1 256 leaves (766 params)
DT w\ DAgger	776.6 ± 54.2 32 leaves (94 params)	184.7 ± 17.3 32 leaves (126 params)	$\mathbf{395.2 \pm 13.8}$ 16 leaves (46 params)	121.5 ± 0.01 16 leaves (46 params)	1249.4 ± 3.4 31 leaves (122 params)	$\mathbf{1113.8 \pm 9.5}$ 16 leaves (46 params)
CDDT-Crisp	5.0 ± 0.0 2 leaves (5 params)	-451.6 ± 97.3 8 leaves (37 params)	-43526.0 ± 15905.0 16 leaves (61 params)	68.1 ± 18.7 16 leaves (61 params)	664.5 ± 192.6 16 leaves (77 params)	322.9 ± 47.1 16 leaves (61 params)
ICCT-static	$\mathbf{984.0 \pm 10.4}$ 32 leaves (125 params)	$\mathbf{192.4 \pm 10.7}$ 32 leaves (157 params)	374.2 ± 55.8 16 leaves (61 params)	120.5 ± 0.5 16 leaves (61 params)	$\mathbf{1271.7 \pm 4.1}$ 16 leaves (77 params)	1003.8 ± 27.2 16 leaves (61 params)
ICCT-1-feature	1000.0 ± 0.0 8 leaves (45 params)	190.1 ± 13.7 8 leaves (69 params)	437.6 ± 7.0 16 leaves (93 params)	121.6 ± 0.5 16 leaves (93 params)	1269.6 ± 10.7 16 leaves (141 params)	1072.4 ± 37.1 16 leaves (93 params)
ICCT-2-feature	1000.0 ± 0.0 4 leaves (29 params)	258.4 ± 7.0 8 leaves (101 params)	458.5 ± 6.3 16 leaves (125 params)	$\mathbf{121.9 \pm 0.5}$ 16 leaves (125 params)	1280.4 ± 7.3 16 leaves (205 params)	$\mathbf{1088.6 \pm 21.6}$ 16 leaves (125 params)
ICCT-3-feature	$\mathbf{1000.0 \pm 0.0}$ 2 leaves (17 params)	275.8 ± 1.5 8 leaves (133 params)	448.8 ± 3.0 16 leaves (157 params)	120.8 ± 0.5 16 leaves (157 params)	$\mathbf{1280.8 \pm 7.7}$ 16 leaves (269 params)	1048.7 ± 46.7 16 leaves (157 params)
ICCT-L1-sparse	1000.0 ± 0.0 4 leaves (29 params)	265.2 ± 4.3 8 leaves (165 params)	465.5 ± 4.3 16 leaves (253 params)	121.5 ± 0.3 16 leaves (765 params)	1275.3 ± 6.7 16 leaves (2189 params)	993.2 ± 14.6 16 leaves (509 params)
ICCT-complete	1000.0 ± 0.0 2 leaves (13 params)	$\mathbf{300.5 \pm 1.2}$ 8 leaves (165 params)	$\mathbf{476.6 \pm 3.1}$ 16 leaves (253 params)	120.7 ± 0.5 16 leaves (765 params)	1248.6 ± 3.6 16 leaves (2189 params)	994.1 ± 29.1 16 leaves (509 params)
CDDT-controllers Crisp	84.0 ± 10.4 2 leaves (13 params)	-126.6 ± 53.5 8 leaves (165 params)	-39826.4 ± 21230.0 16 leaves (253 params)	97.9 ± 12.0 16 leaves (765 params)	639.62 ± 160.4 16 leaves (2189 params)	245.5 ± 48.5 16 leaves (509 params)
MLP-Lower	1000.0 ± 0.0 79 params	231.6 ± 49.8 110 params	474.7 ± 5.8 127 params	$\mathbf{121.8 \pm 0.6}$ 151 params	646.4 ± 151.2 221 params	868.4 ± 100.9 103 params
MLP-Upper	1000.0 ± 0.0 121 params	288.7 ± 2.8 222 params	467.9 ± 8.5 407 params	121.8 ± 0.3 709 params	1239.5 ± 4.2 3266 params	1077.7 ± 31.1 1021 params
MLP-Max	1000.0 ± 0.0 67329 params	$\mathbf{298.5 \pm 0.7}$ 68610 params	$\mathbf{478.2 \pm 6.7}$ 69377 params	121.7 ± 0.4 77569 params	1011.9 ± 141.3 83458 params	$\mathbf{1104.3 \pm 9.4}$ 73473 params
CDDT	$\mathbf{1000.0 \pm 0.0}$ 2 leaves (8 params)	226.4 ± 44.5 8 leaves (86 params)	464.7 ± 5.4 16 leaves (226 params)	120.9 ± 0.5 16 leaves (706 params)	$\mathbf{1248.0 \pm 6.4}$ 16 leaves (1036 params)	1033.2 ± 24.1 16 leaves (466 params)
CDDT-controllers	1000.0 ± 0.0 2 leaves (16 params)	289.0 ± 2.4 8 leaves (214 params)	469.7 ± 11.1 16 leaves (418 params)	120.1 ± 0.3 16 leaves (1410 params)	1243.8 ± 3.6 16 leaves (2092 params)	1010.9 ± 25.7 16 leaves (914 params)

Interpretability-Performance Tradeoff

We conduct an ablation to look at the interpretability-performance tradeoff of our ICCTs, assessing the change in performance as we vary the number of leaves in our tree and varying the number of active features in our leaf controllers.



- Lakkaraju et al.[8] states that decision trees are interpretable because of their simplicity and that there is a cognitive limit on how complex a model can be while also being understandable.
- Our model architecture allows us to study the tradeoff in interpretability and performance.
  - In Lunar Lander, we find as model complexity increases, there is a slight gain in performance and a large decrease in interpretability.

The Pareto-Efficiency curves provides insight into the interpretability-performance tradeoff for ICCT tree depth.

Conclusion

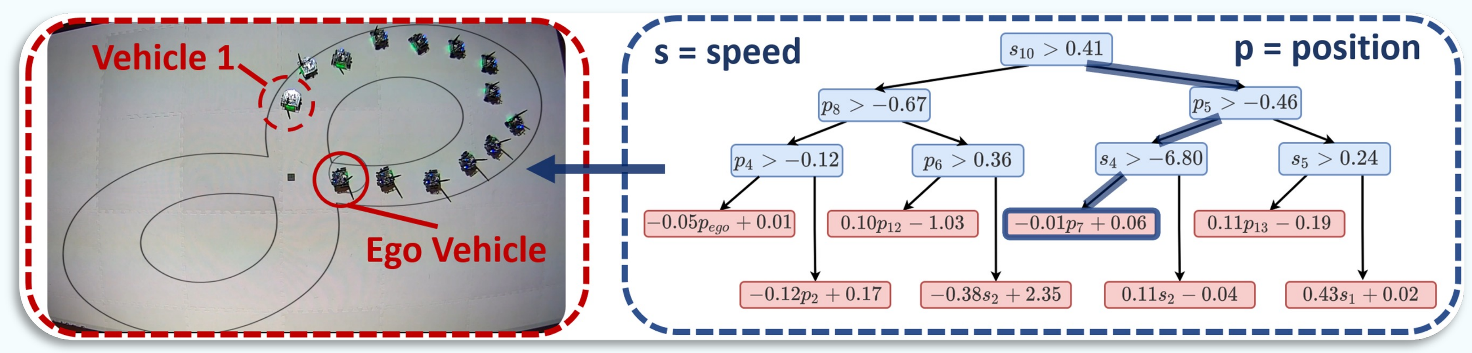


Figure 3. ICCTs controlling a vehicle in a 14-car physical robot demonstration within a Figure-8 traffic scenario.

We present a framework to allow for interpretable reinforcement learning in continuous action spaces, addressing one of the grand challenges for interpretable ML by providing a technique for direct optimization for sparse, logical models.

Authors

Rohan Paleja | Yaru Niu | Andrew Silva | Chace Ritchie | Sugju Choi | Matthew Gombolay