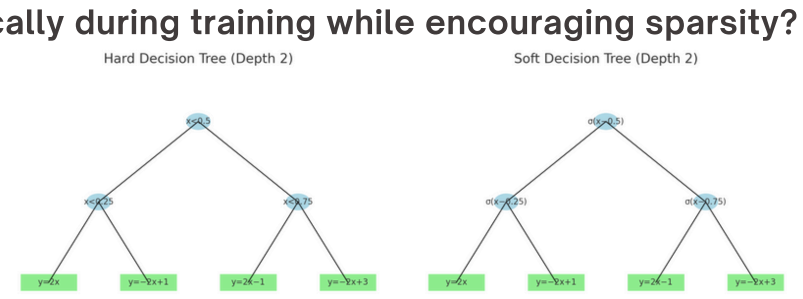
SPLIT-PO : Sparse Piecewise-Linear Interpretable Tree Policy Optimization

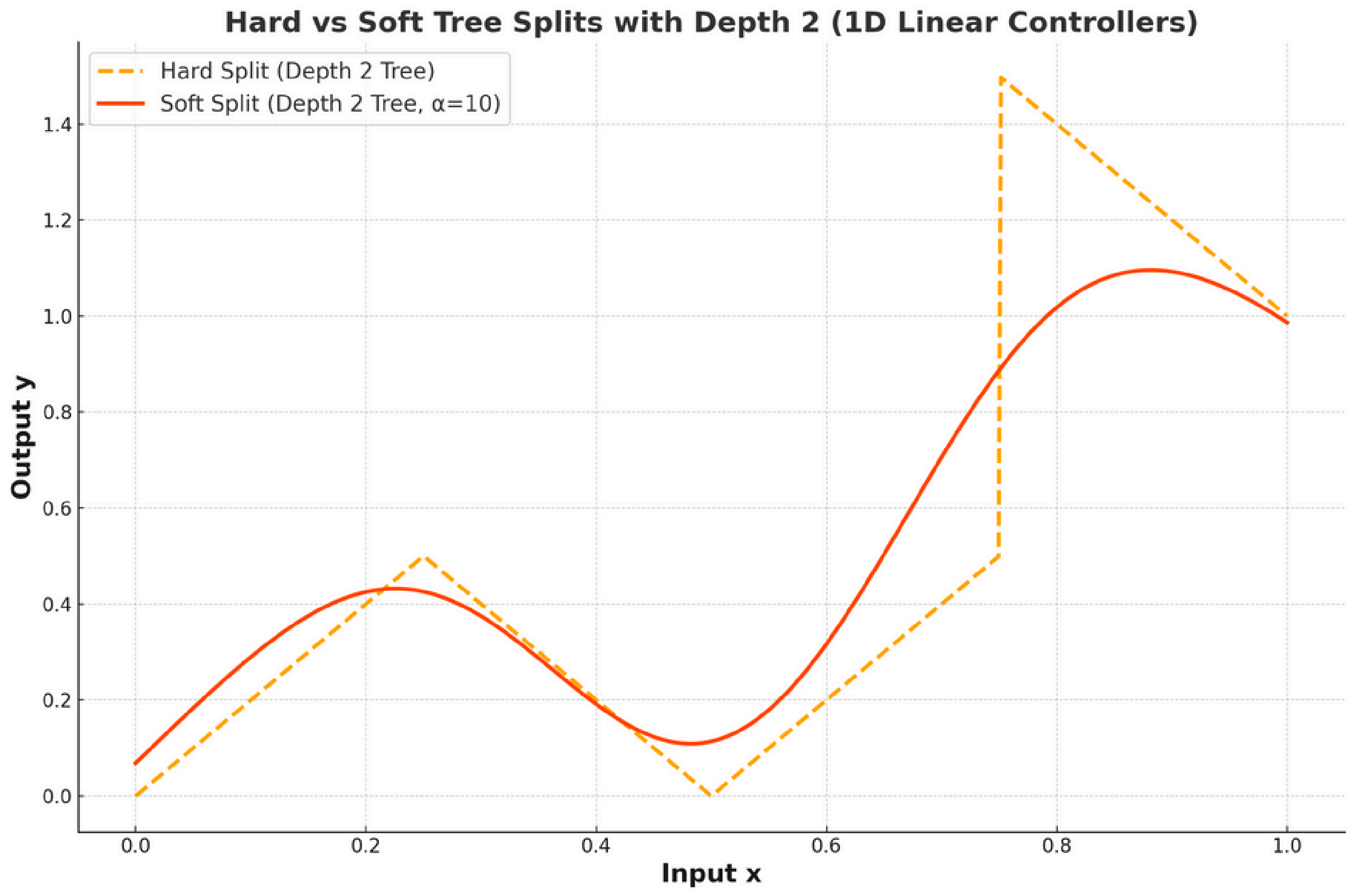
Author : Ernesto Hellouin de Menibus
Supervisor : Daniël Vos
Responsible Professor : Anna Lukina
Delft University of Technology

An Interpretable and Differentiable Framework for Sparse-Tree Policy Optimization

1 Why Tree Based Policies?

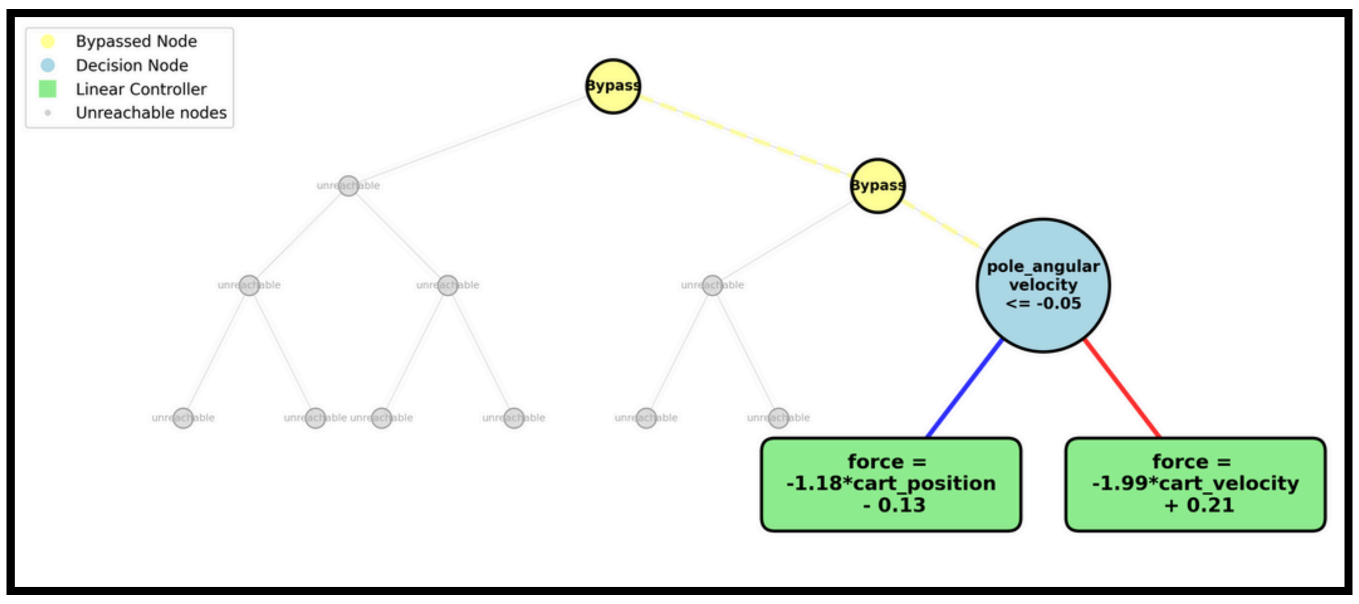
- Deep Neural Networks perform well but are Black-Box.
- Decision Trees offer interpretability but they are **not-differentiable.** Making them unusable with gradient based optimization.
- **Differentiable Decision Trees (DDTs) and Interpretable Continuous Control Trees (ICCTs)** were introduced to allow trees to be differentiable by using soft splits.
- However tree size and strucutre is fixed. This can lead to unecessarly large and less intepretable trees aswell as structural bias.
- **How can we design a reinforcement learning framework that allows differentiable piecewise-linear decision trees to adapt their structure dynamically during training while encouraging sparsity?**





2 Sparse Piecewise-Linear Interpretable Tree Policy Optimization (SPLIT-PO)

- **Dynamic Sparse Structure:** Learns which nodes to keep or bypass during training
- **Sparse Controllers:** Uses top-k feature selction for each leaf (only few input features are used)
- **Crisp + Differentiable:** Straight-through estimators allow hard splits in forward pass, gradients in backwards
- **End-to-end training:** with actor critic RL (DDPG, SAC)



Example learned tree for Inverted Pendulum (*k=1*): Bypassed nodes are yellow; green leaves show sparse linear controllers.

Gated Node Activation

$$\hat{g}_i = \text{step}(\sigma(g_i)) + (\sigma(g_i) - \text{detach}(\sigma(g_i)))$$

Path Probability to Leaf l

$$P_\ell(x) = \prod_{(i,d_i)\in \text{path}(\ell)} \begin{cases} \hat{g}_i \cdot \hat{s}_i(x) & \text{if } d_i = 0 \\ (1-\hat{g}_i) + \hat{g}_i \cdot (1-\hat{s}_i(x)) & \text{if } d_i = 1 \end{cases}$$

Total Policy Objective

$$\mathcal{L}_{\text{total}} = \underbrace{\mathcal{L}_{\text{actor}}}_{\text{Policy Gradient}} + \underbrace{\mathcal{L}_{\text{critic}}}_{\text{TD Error}} + \underbrace{\lambda \sum_{i\in\mathcal{N}} \sigma(g_i)}_{\text{Gate Regularization}}$$

Note : $\hat{s}_i(x)$ is the ICCT-style crispified split: sigmoid forward, gradient backwards.

SPLIT-PO introduces learnable gates to dynamically prune the tree. Paths are computed using gated splits, and a regularization term encourages sparsity.

3 Results

- **Selected results highlighting key performance and interpretability trade-offs**
- **SPLIT-PO matches or exceeds baseline performance with orders-of-magnitude fewer parameters and compact, interpretable tree policies.**
- **It almost always makes trees smaller than ICCT and uses less then 1% of the number of parameters as the MLP baseline.**

Model	Env	Reward	Leaves	Params
SPLIT-PO (k=2)	Inverted pendulum	1000 ± 0	2	73
ICCT (k=2)	Inverted pendulum	1000 ± 0	4	30
MLP	Inverted pendulum	1000 ± 0	n/a	67,586
SPLIT-PO (k=k)	Lunar Lander	285.20 ± 21.03	1	221
ICCT (k=k)	Lunar Lander	279.00 ± 18.44	8	214
MLP	Lunar Lander	287.43 ± 14.23	n/a	69,124

4 Key Takeaways - Future Work - Limitations

Key Contributions

- **Interpretable tree policies** trained end-to-end using actor-critic RL
- **Dynamic sparsity:** gates learn which nodes to keep or bypass during training
- **Sparse linear controllers** with 1-k features per leaf
- **Small trees :** Loss function promotes smaller, more interpretable policies.

Limitations

- **Sample inefficient compared to MLPs,** needs more training steps
- **Fixed-depth limitation:** tree depth still needs to be pre-set
- **Difficulties in high dimensional environments**

Future Work

- **Improve sample efficiency** using imitation learning or warm starts
- **Support image-based observations** via feature extractors
- **Extend to discrete/hybrid action spaces**
- Explore **verifiability** and **formal guarantees** of resulting tree policies