Decision Trees vs. Ensembles in Regression-Based Offline RL

Interpretability–Performance Trade-offs and Return-to-Go Effects

Rareş Polenciuc – R.Polenciuc@student.tudelft.nl
Supervisors: Dr. Anna Lukina, Daniël Vos
EEMCS, Delft University of Technology, The Netherlands

TU Delft

1. Introduction

Reinforcement Learning?

- Agent interacts with environment (robot, game, etc.)
- Takes actions → Gets rewards → Learns better policy
- **Problem**: What if interaction is dangerous/expensive?
  - e.g. Medical treatment decisions, Autonomous vehicles, Industrial control systems

2. Offline RL

Learning from Pre-Collected Data

Offline RL Solution:

- Train policies from existing datasets
- No live interaction needed
- Use historical data (logs, previous experiments)

Current State-of-the-Art:

- Complex ensemble methods (e.g., GBDT: 1000+ trees)
- High performance but **complete black boxes**

3. The Interpretability Problem

Why This Matters for Critical Applications

The Dilemma:

- **High Performance:** Complex ensembles → Opaque decisions
- **Interpretability:** Simple models → Explainable but weak?

Example:

- Healthcare: "Why did the AI recommend this treatment?"

Our Research Question: *What are the interpretability-performance trade-offs when using CART trees for offline reinforcement learning compared to GBDT ensembles?*

4. Our Approach

Offline RL as Supervised Learning

Offline RL as Regression: Input = (State, Return-to-Go, Timestep)
Output = Action

Models Compared:

- **XGBoost:** Ensemble (1000 trees)
- **CART:** Single tree (interpretable when small / medium)
- **M-CART:** One tree per action

Interpretability Tiers: CART: Small/Medium (interpretable) to Large/Unbounded. M-CART: Small/Medium/Large.

5. Overall Performance

Decision Trees vs. Ensembles

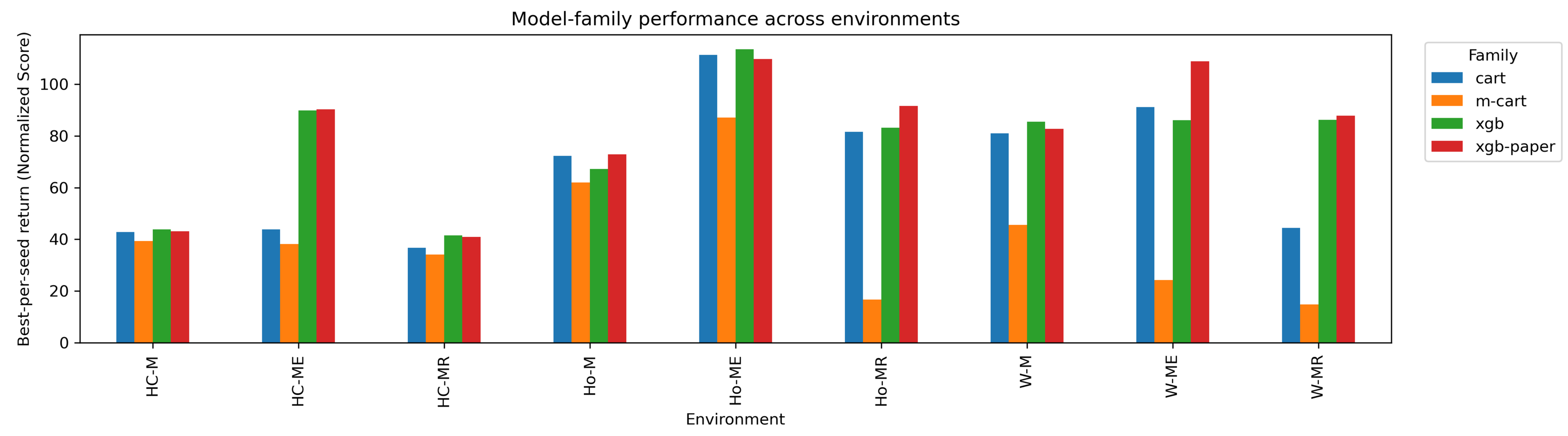

Figure: CART and M-CART show competitive results in several tasks.

- **XGBoost Baseline:** Our reproduction aligns with prior work.
- XGBoost wins most tasks.
- **But CART is competitive in many cases**
- M-CART shows decent performance in specific tasks

6. Complexity vs. Interpretability

What are the Tradeoffs?

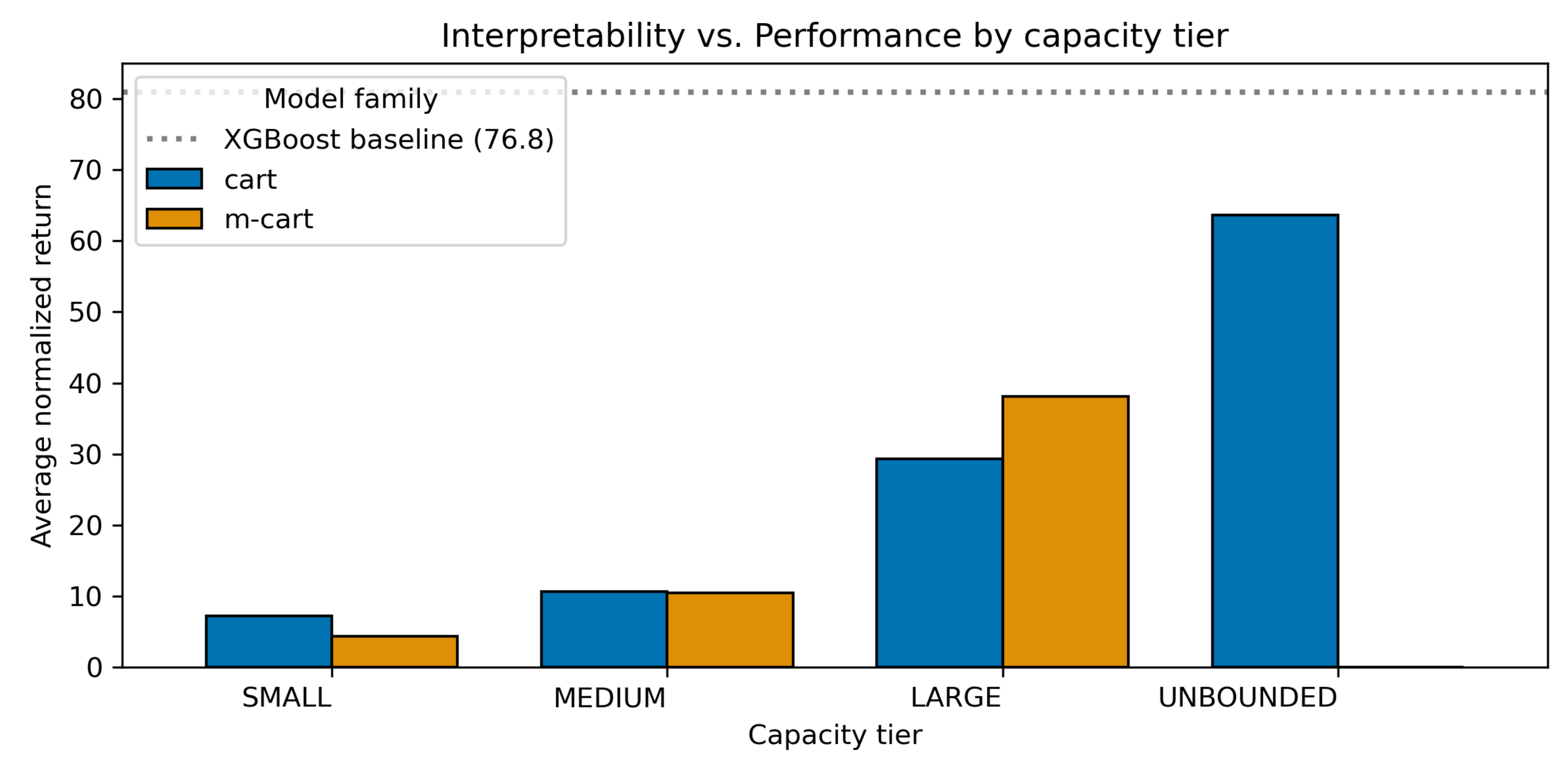

Figure: Performance improves with complexity, but interpretability drops.

- **Small/Medium (truly interpretable):** Only 13-14% of XGBoost performance
- **To get competitive performance:** Need depths >10 levels
- **At depth >10:** Not really interpretable anymore
- **M-CART advantage:** Better performance than single trees when size-constrained

7. RTG Sensitivity

Return-to-Go Effects

RTG = Return-To-Go = target reward

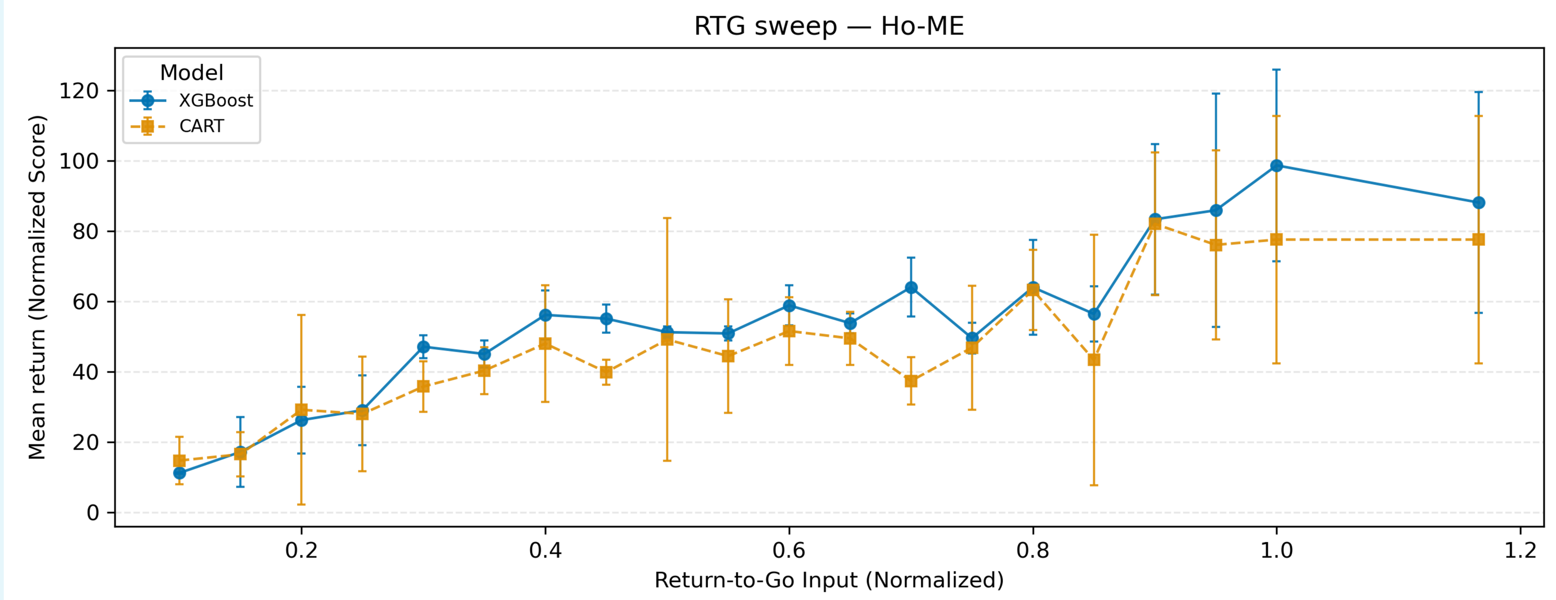

Figure: Tiny RTG changes = huge action differences.

Behavioral Fragility with RTG:

- CART policies: **high sensitivity to RTG input** (ρ).
- Small ρ shifts → **abrupt, erratic performance changes**.

Implications for Interpretability:

- Traditional DT interpretability (fixed state → fixed action) is broken by dynamic RTG.
- **Key Takeaway:** Requires full RTG tracing; same states → different actions with ρ variations.

8. Key Takeaways

What This Means for Interpretable AI

a. **Performance vs. Interpretability:** Fundamental trade-off.
b. **Complexity:** Good performance needs non-interpretable complexity
c. **RTG Problem:** Dynamic inputs break traditional interpretability

Broader Impact:

- Simple model architecture $\neq$ interpretable behavior
- Need new approaches for truly auditable AI systems

9. Future Work

Next Steps

- Alternative conditioning signals
- Hierarchical decision structures

Ultimate Goal: Auditable RL for critical applications