Learning Tree Interpretation from Object Representation for Deep Reinforcement Learning

Guiliang Liu, Xiangyu Sun, Oliver Schulte, and Pascal Poupart

Presenter: Francis Kiwon, Department of Statistics and Actuarial Science

23 March 2022

Introduction

- DQN learns to play as well as a professional gamer does (Mnih et al., 2015)
- Limitation: Deep neural networks are "black boxes"
  - Deep RL (DRL) agents are promising, but we do not know what strategies they adopt
- Interpreting DRL models enhances trust and complies with regulations
  - "A right to explanation" established by the EU's General Data Protection Regulation
- We want to **explain**, or **interpret**:
  - How important is each input feature?
  - How does it actually influence the agent's decisions?
  - What and how much did an agent learn from each input?

Introduction

- Previous works focused on visualization of the pointwise importance of low-level *input features*
- However, we want to reveal a *global* causal relationship between targets and high-dimensional inputs
- Let's build transparent trees which "mimic" the DRL model, but:
  - Numerous splits keep us from understanding the "accurate" decision rules
  - Any constraints on the tree complexity leads to the limited performance
- Following the **Information Bottleneck** principle, we learn:
  - The compressed and hidden features which best represent the raw inputs
  - The simplest mimic tree based on such features from the representation

What is “Interpretability”?

Definition by Murdoch et al. (2019)

“The extraction of relevant knowledge from a machine learning model, concerning relationships either contained in data or learned by the model."

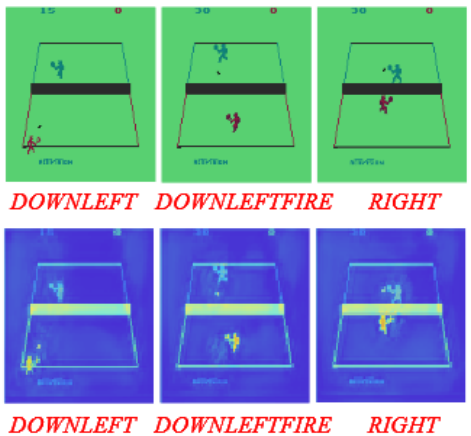
- We may desire the following characteristics for interpretations:
  1. **Predictive Accuracy**: Did our model learn a good approximation?
  2. **Descriptive Accuracy**: Does interpretations “truthfully” represent the actual relationship learned by the model?
  3. **Relevancy**: Does the interpretation provide insight into a chosen domain problem?
  4. **Simplicity**: Can we easily understand it?
  5. **Consistency**: Do different models produce similar predictions and interpretations given the same data?

Types and Scopes of Interpretation (Alharin et al., 2020)

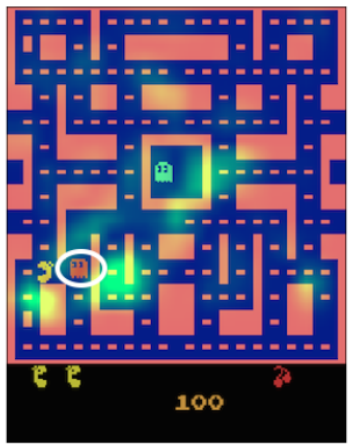
- Interpretation can be either post-hoc, intrinsic, or both:
  1. **Post-Hoc**: Explain the learned rules given the original model architecture
  2. **Intrinsic**: Replace the original model with a transparent alternative
- As a result of interpretation, we can explain either of:
  1. **Local**, or prediction-level decisions of the model at a specific input
  2. Its **global**, or dataset-level strategy in taking actions
- There are various means of delivering interpretations: Graphs, Saliency Maps, Natural Language, Mathematical Expressions, etc.

Previous Works on DRL Interpretations

1. Visualization based on the high-dimensional input state



Masked State-Action Pairs
(Shi et al., 2020)



An Unsuccessful Agent
(Greydanus et al., 2018)

- Attention distributions are not identifiable for local samples
- The interpretations are pointwise, so cannot identify the underlying causality

Previous Works on DRL Interpretations

2. Mimic Learning

- A simple model can learn the complex functions as accurately as deep models (Ba and Caruana, 2014)
- Liu et al. (2018) and Sun et al. (2020) approximated the Q functions using linear trees $\Rightarrow$ Too complex interpretations
- Boz (2002) entertained pruning given the constraint of tree complexity $\Rightarrow$ Too limited performance

The Information Bottleneck (IB; Tishby et al., 1999)

- We want the mimic learner to preserve both of:
  - Information about targets (Descriptive Accuracy / Fidelity)
  - Conciseness of the input data (Simplicity)
- **An Issue**: It may not learn the raw inputs $X := (S, A, R)$
- **Solution**: Learn a latent representation $Z = \{Z_d\}_{d=1}^{D}$ first, and build a mimic model ϕ upon Z
- **Another Issue**: The marginal distribution $p(X)$, and therefore the posterior $p(Z|X)$ are both intractable in practice
- **Solution**: Do variational approximation!

"Represent And Mimic" (RAMi) Framework

- RAMi separately interprets:
  - Input features with their interpretable latent representations
  - Decision rules with a transparent mimic tree
- The knowledge of a DRL model is distilled to a mimic tree, which will learn post-hoc interpretations
- We want to mimic action advantages defined as $y = Q(s, a) - V(s)$
- The mimic learner lets us understand *when* an action outperform others by y
- By Theorem, we maximize the lower bound of the following function:

Evidence Lower Bound + **Minimum Description Length** + **Entropy Regularizer**

Identifiable Multi-Object Network (IMONet)

Our IMONet adopts the following two frameworks:

1. **Variational Auto-Encoders** (VAE; Kingma and Welling, 2014)

   1. *Approximate:* $q(Z|X) \approx p(Z|X)$ ***2.*** *Encode:* $X \rightarrow q(Z|X)$

   3. *Sample & Decode:* $Z_{\sim q(Z|X)} \rightarrow p_d(X|Z) \Rightarrow$ ***4.*** *Reconstruct:* $\tilde{X}$

- In IMONet, we assume $Z_1, \cdots, Z_D$ independent i.e., $p(Z) = \prod_{d=1}^{D} p(Z_d)$

  $\Longrightarrow$ Each Z_d is disentangled and identifiable

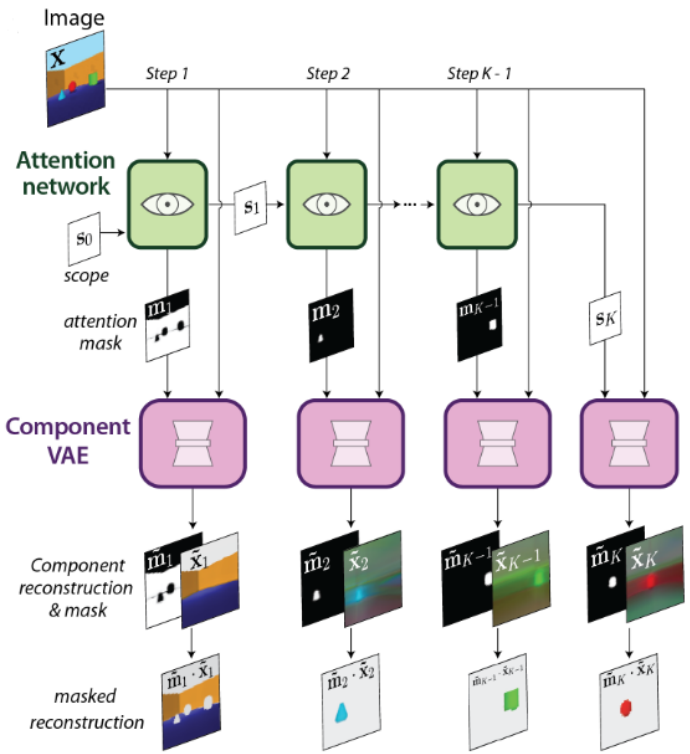
  $\Longrightarrow$ We can model causal relations between Z and Y

- A good approximation of $p(Z|X)$ minimizes $\mathcal{D}_{KL}[q(z|x_n)||p_0(z)]$

  $\Longrightarrow$ It then maximizes the Evidence Lower Bound and fidelity!

Identifiable Multi-Object Network (IMONet)

2. **Multi-Object Network** (MONet; Burgess et al., 2019)



Schematic of MONet

1. *Decompose:*
$s = (s_1, \ldots, s_K)|(m_1, \ldots, m_K)$

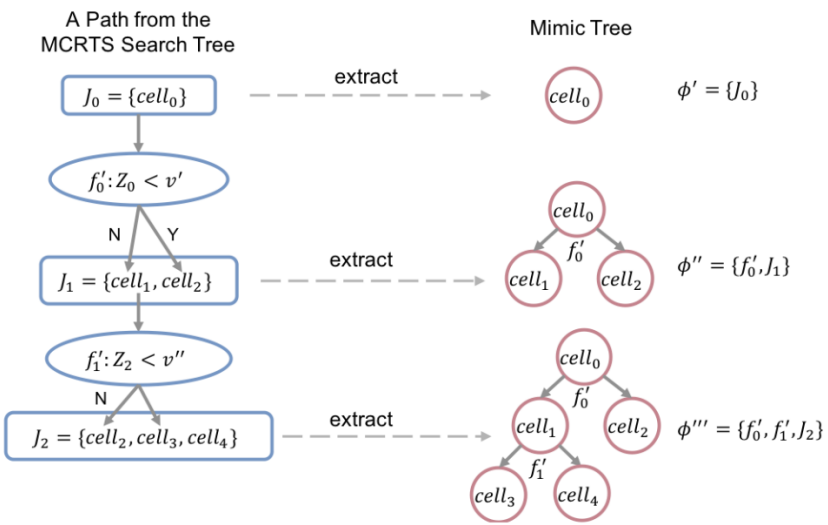
2. *Encode-Sample-Decode*:*
Conditional VAE on $s_k|m_k, a, r$,
employing a factored prior
$p(Z|A, R) = \prod_{d=1}^{D} p(Z_d|A, R)$

3. *Reconstruct:* $\tilde{m}_k \cdot \tilde{s}_k$

*See also Sohn et al. (2015)

Monte Carlo Regression Tree Search (MCRTS)

- MCRTS minimizes the IB-Mimimum Description Length



MCRTS constructs a *search tree*;
An edge refers to a split in the selected *mimic tree*

- Extract z_i, which collects the vectors of D-dimensional latent features from K objects for the ith instance
- Construct $\langle z_i, a_i, r_i; y_i \rangle$ as inputs and store them at the root node
- Partition the instances in a parent by a split f to two cells in children
- Record the number of visits and the estimate of Q at f
  $\Rightarrow$ MCRTS therefore learns a compact distribution $p(\Phi|Z)$, where $\phi \in \Phi$

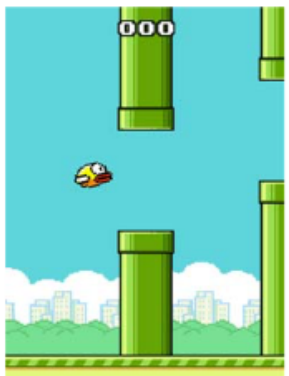
Monte Carlo Regression Tree Search (MCRTS)

1. Search: Run M plays from a starting node
   - At each mth play and lth layer, select the split $f_{l,m}$ that maximize the upper confidence bound:

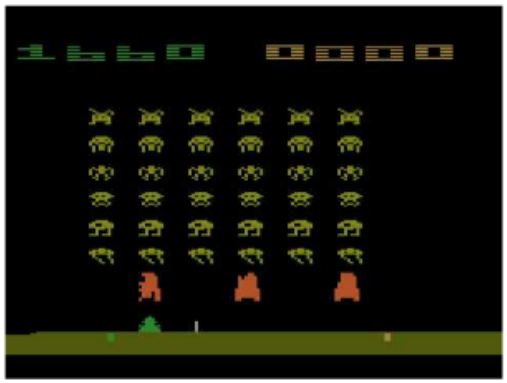
$$\mathrm{UCB} = \mathrm{argmax}_f \left[Q^{MC}_{m-1}(J_l, f) + c\sqrt{\log(m-1)} / \left\{ NV_{m-1}(J_l, f) + 1 \right\} \right]$$

   - Augment the previous estimate more for a less visited node
   - Control the exploration with a constant c
2. Evaluate: Evaluate the selected leaf node J with reward r^{MC}
3. Expand: Expand the leaf with children
4. Update: $Q^{MC}_m = \left(Q^{MC}_{m-1} + r^{MC} \right) / \left(NV\, m - 1 + 1 \right)$
5. Move: Select the split with the highest NV_M and set the starting node to the connected child

Experiment: Environments



Flappy Bird



Space Invaders

Assault

- Flappy Bird: 0.1 reward per step, +1: Pass, -1: Interference
  - The pillars, or states are *randomly* generated
- Space Invaders and Assault: +1 per kill

Experiment: Implementation

1. Train a DRL agent for each environment
   - Flappy Bird: DQN (Chen, 2015)
   - Space Invaders and Assault: A3C (Mnih et al., 2016)
2. Collect the $N = 50,000$ pairs of $(\langle s_n, a_n, r_n \rangle, y_n)$
   - An ϵ-greedy Policy with $\epsilon = 0.01$
   - Train-Validation-Test Split: 80-10-10
3. Train the tree-based baseline mimic methods with the *raw* input data
   - **MCRTS**, **CART** (Breiman, 1984; Timofeev, 2004), **VIPER** (Bastani, Pu, and Solar-Lezama, 2018), **M5** (Quinlan et al., 1992), **Linear Model Trees** (**LMT**; Liu et al., 2018; Sun et al., 2020)
4. Compare their performance based on the latent representation learned by **IMONet**

Experiment: Latent Traversals

1. Get a random sample of size 1000
2. Average all the latent features of the sampled images generated by IMONet
3. Traverse each latent feature $Z_{k,d}$, having other $KD - 1$ values fixed
4. Observe the variations of generated images

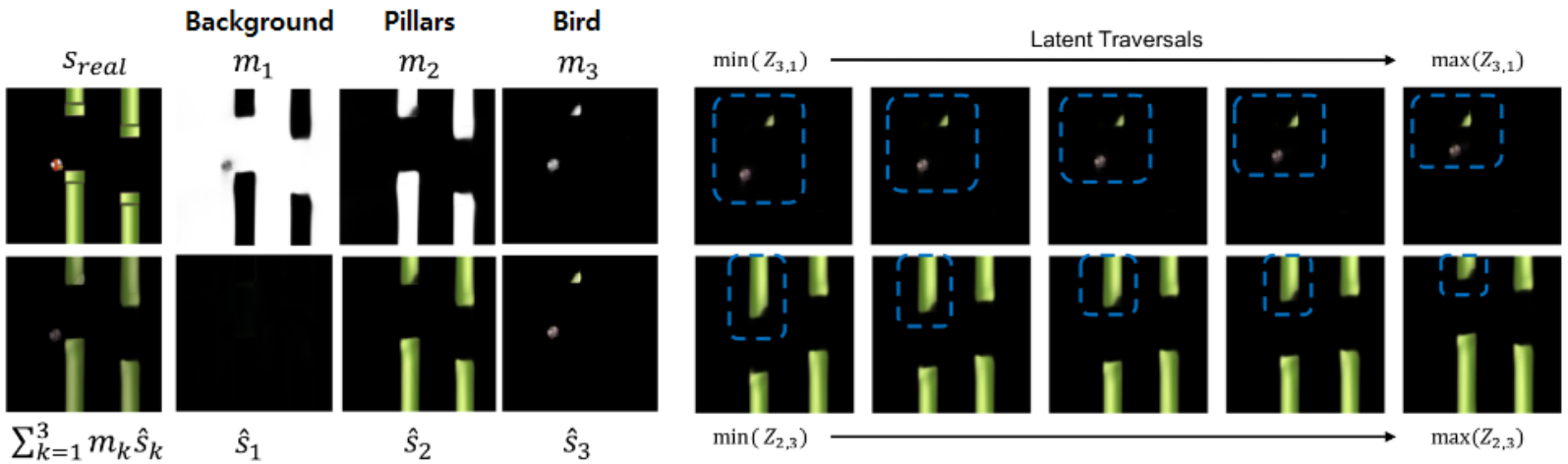


Figure 1: Visualized IMONet Outputs in a Flappy Bird Experiment

Results

	Flappy Bird			Space Invaders			Assault		
Method	VR	VR-PL	Leaf	VR	VR-PL	Leaf	VR	VR-PL	Leaf
Cart	8.51E-2	8.43E-5	1007	4.96E-2	7.02E-5	705	4.79E-2	7.46E-5	642
VIPER	8.57E-2	1.88E-4	453	4.63E-2	8.80E-5	525	5.28E-2	8.09E-5	653
M5-RT	**9.59E-2**	8.37E-5	1144	4.54E-2	2.92E-5	1558	4.37E-2	2.73E-5	1605
M5-MT	9.56E-2	1.55E-4	612^{w_+}	1.60E-2	1.23E-5	1303^{w_+}	3.42E-2	2.54E-5	1351^{w_+}
GM-LMT	8.99E-2	2.99E-4	303^{w_+}	2.07E-2	8.32E-5	249^{w_+}	5.55E-2	1.83E-4	307^{w_+}
VR-LMT	8.46E-2	5.36E-4	157^{w_+}	2.65E-2	1.61E-4	166^{w_+}	5.80E-2	1.98E-4	291^{w_+}
VAE+CART	7.25E-2	3.44E-4	212	3.99E-2	7.86E-5	507	5.15E-2	1.16E-4	448
VAE+VIPER	7.63E-2	5.32E-4	143	4.12E-2	9.89E-5	417	4.57E-2	1.29E-4	356
VAE+GM-LMT	6.35E-2	3.51E-4	180^{w_+}	3.39E-2	2.75E-4	123^{w_+}	4.20E-2	1.44E-5	293^{w_+}
VAE+VR-LMT	7.95E-2	5.12E-4	154^{w_+}	3.52E-2	2.08E-4	171^{w_+}	5.10E-2	1.99E-4	258^{w_+}
VAE+MCRTS	7.83E-2	1.27E-3	61	4.82E-2	5.66E-4	85	6.58E-2	7.75E-4	85
IMONet+CART	8.23E-2	4.02E-4	204	5.21E-2	1.38E-4	375	5.67E-2	1.81E-4	315
IMONet+VIPER	8.50E-2	4.48E-4	191	5.26E-2	1.69E-4	313	6.05E-2	1.90E-4	319
IMONet+GM-LMT	7.87E-2	3.74E-4	212^{w_+}	4.79E-2	3.23E-4	149^{w_+}	5.45E-2	2.15E-4	256^{w_+}
IMONet+VR-LMT	8.21E-2	7.16E-4	115^{w_+}	4.54E-2	3.79E-4	120^{w_+}	6.03E-2	2.27E-4	268^{w_+}
IMONet+MCRTS	8.53E-2	**1.37E-3**	**62**	**5.37E-2**	**7.08E-4**	**76**	**7.53E-2**	**9.07E-4**	**83**

Figure 2: Regression Performance

- VR, VR-PL: Variance Reduction, – per Leaf
- RT, MT, GM: Regression-Tree, Model-Tree, Gaussian Mixture
- w_+: Each leaf node has an extra linear model
- MCRTS trained with the raw data is intractable

Results

- The combination of IMONet and MCRTS presents a promising performance with significantly fewer leaves
- The object representation learned by IMONet outperformed others thanks to identifiability
- MCRTS considers the tree's performance at a global level, and can maintain the simple mimic tree's fidelity
- Some trees built from raw inputs may outperform in terms of other metrics such as RMSE, but their size is far larger than our model

Results

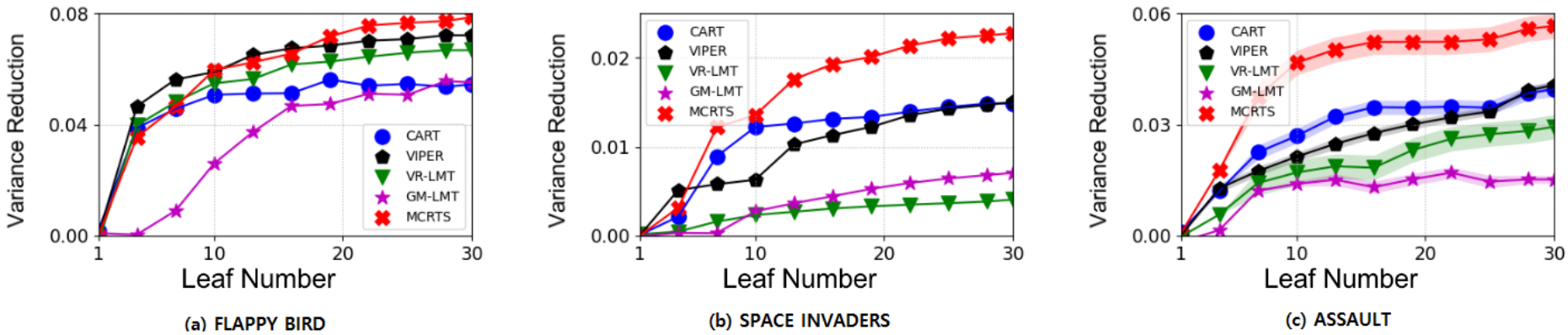


Figure 3: Leaf-by-Leaf Regression Performance based on the latent features from IMONet

- If we constrain the number of leaves, MCTRS dominates
- MCRTS looks ahead to the future cumulative rewards instead of local influence
- The selected split is well-explored, and therefore more efficient than a greedy one with extra linear regressors at leaf nodes

Results

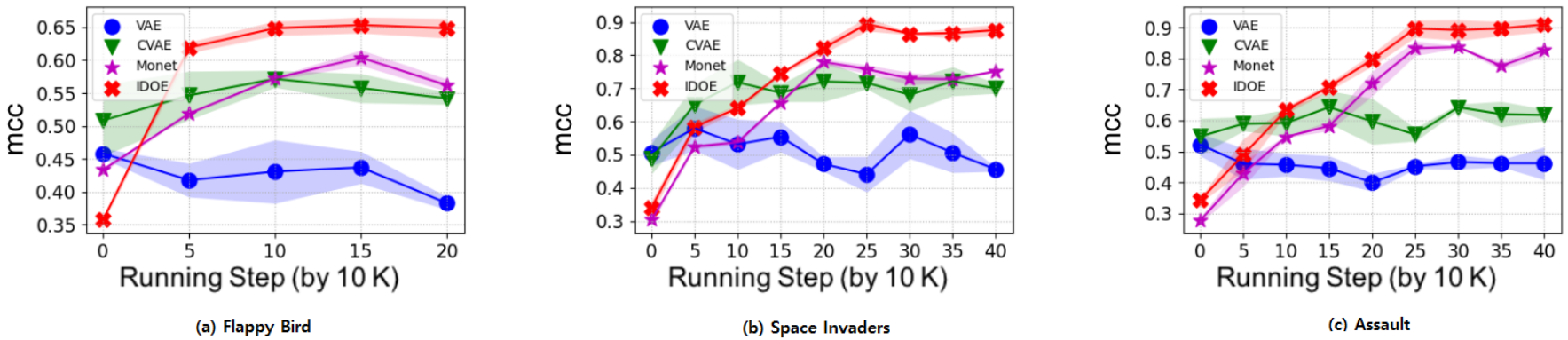


Figure 4: Mean Correlation Coefficients (MCC) for Different Variational Encoders

- MCC measures the latent features from one model differs enough than those from the other
- Conditioning variables (action, reward) and an object network together further improve the identifiability of latent features

Interpretability of the IMONet+MCRTS Mimic Tree

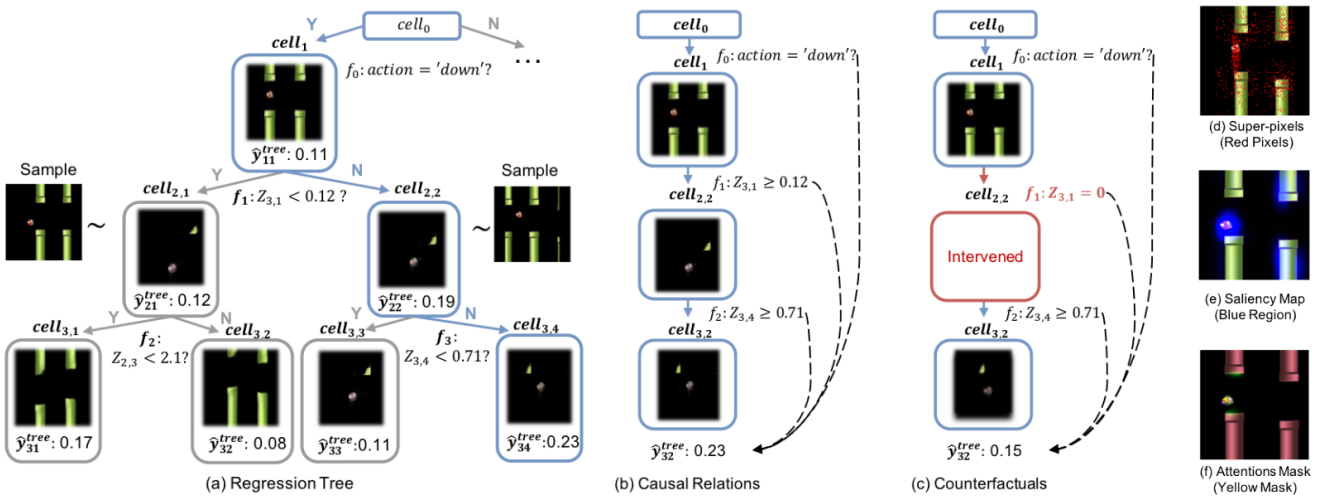


Figure 5: Mimic Tree. f_l: the lth split; Solid / Dash Lines: Path / Causality

- Causal Relation: When the bird goes "down" and it is closer to the upper pillar i.e., $Z_{3,1} \geq 0.12$, the advantage is maximized
- Counterfactual: If the bird is far enough from the upper pillar i.e., $Z_{3,1} = 0$, the advantage decreases

Concluding Remarks

- The IB principle led to the development of the framework which jointly optimizes the fidelity and the simplicity of the mimic tree
- Utilizing the conditional VAE, IMONet converts state features to an identifiable latent representation which captures the independent factors of variation for the masked objects
- MCRTS learns a compact distribution over the collection of mimic trees, and decrease the complexity of the optimal mimic tree which minimizes the IB-minimum description length
- The nature of MCRTS, which conducts multiple simulations for searching the optimal mimic tree, increases the computational cost
- The empirical evaluation involved illustrative examples and human evaluation, because it is generally hard to theoretically justify or numerically quantify the level of interpretability

Works Not Cited in the Original Paper

- W. J. Murdoch, C. Singh, K. Kumbiera, R. Abbasi-Asl, and B. Yu (2019). Definitions, methods, and applications in interpretable machine learning. *PNAS* (116), 22071–80.
- A. Alharin, T.-N. Doan, and M. Sartipi (2020). Reinforcement learning interpretation methods: A survey. *IEEE Access*(8), 171058-77.