Training Characteristic Functions with Reinforcement Learning

Stephan Wäldchen, Felix Huber, Sebastian Pokutta
Zuse Institut Berlin

39th International Conference on Machine Learning
July 17 – 23, 2022

How do we interpret black-box Functions?

How do we interpret black-box Functions?

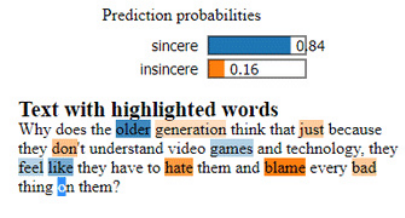
- Core idea: Probe black-box function with different inputs

How do we interpret black-box Functions?

- Core idea: Probe black-box function with different inputs
- Local interpretation (for specific input $\mathbf{x}$): vary features

How do we interpret black-box Functions?

- Core idea: Probe black-box function with different inputs
- Local interpretation (for specific input $\mathbf{x}$): vary features

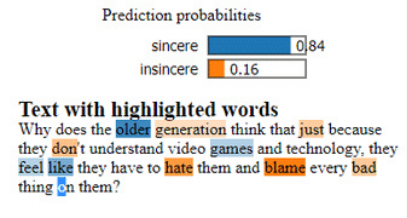


LIME saliency[1]

1) "Explainable ai: A review of machine learning interpretability methods", Linardatos et al. [7]

How do we interpret black-box Functions?

- Core idea: Probe black-box function with different inputs
- Local interpretation (for specific input $\mathbf{x}$): vary features
- ML: Saliency $ Coop. Game Theory: Surplus Attribution

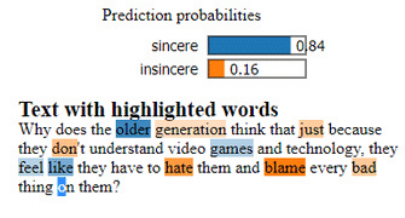


LIME saliency[1]

1) "Explainable ai: A review of machine learning interpretability methods", Linardatos et al. [7]

How do we interpret black-box Functions?

- Core idea: Probe black-box function with di erent inputs
- Local interpretation (for speci c input **x**): vary features
- ML: Saliency $ Coop. Game Theory: Surplus Attribution
- Characteristic function: : $2^{[d]}$! R

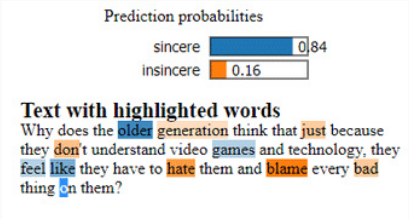


LIME saliency[1]

1) "Explainable ai: A review of machine learning interpretability methods", Linardatos et al. [7]

How do we interpret black-box Functions?

- Core idea: Probe black-box function with di erent inputs
- Local interpretation (for speci c input **x**): vary features
- ML: Saliency $ Coop. Game Theory: Surplus Attribution
- Characteristic function: : $2^{[d]}$! R



LIME saliency[1]

Source: https://clearcode.cc/blog/game-theory-attribution/

How do we interpret black-box Functions?

- Core idea: Probe black-box function with di erent inputs
- Local interpretation (for speci c input **x**): vary features
- ML: Saliency $ Coop. Game Theory: Surplus Attribution
- Characteristic function: : $2^{[d]}$! R
- Prime Implicant Explanation[2] (mostly for : $2^{[d]}$! $f0;1g$)

$$S = \underset{S \quad [d]}{\operatorname{argmin}} jSj \quad \text{s.t.} \quad (S) = ([d])$$

Source: https://clearcode.cc/blog/game-theory-attribution/

1) "Explainable ai: A review of machine learning interpretability methods", Linardatos et al. [7] 2) "A symbolic approach to explaining bayesian network classifiers", Shih et al. [13]

How do we interpret black-box Functions?

Source: https://clearcode.cc/blog/game-theory-attribution/

- Core idea: Probe black-box function with di erent inputs
- Local interpretation (for speci c input $\mathbf{x}$): vary features
- ML: Saliency $ Coop. Game Theory: Surplus Attribution
- Characteristic function: : $2^{[d]}$! R
- Prime Implicant Explanation[2] (mostly for : $2^{[d]}$! $f0;1g$)

$$S = \underset{S \quad [d]}{\operatorname{argmin}} jSj \quad \text{s.t.} \quad (S) = ([d])$$

- Shapley Values[3] (linear, e cient, symmetric, null-player)

$$_i = \frac{1}{d!} \sum_{2\Pi([d])} ((P_i [fig) \quad (P_i)):$$

1) "Explainable ai: A review of machine learning interpretability methods", Linardatos et al. [7] 2) "A symbolic approach to explaining bayesian network classifiers", Shih et al. [13] 3) "A value for n-person games" Shapley [12]

How do we interpret black-box Functions?

Source: https://clearcode.cc/blog/game-theory-attribution/

- Core idea: Probe black-box function with di erent inputs
- Local interpretation (for speci c input **x**): vary features
- ML: Saliency $ Coop. Game Theory: Surplus Attribution
- Characteristic function: : $2^{[d]}$! R
- Prime Implicant Explanation[2] (mostly for : $2^{[d]}$! $f0;1g$)

$$S = \underset{S \quad [d]}{\operatorname{argmin}} jSj \quad \text{s.t.} \quad (S) = ([d])$$

- Shapley Values[3] (linear, e cient, symmetric, null-player)

$$_i = \frac{1}{d!} \sum_{2\Pi([d])} ((P_i [fig) \quad (P_i)):$$

- Problem: We don't have characteristic functions!

1) "Explainable ai: A review of machine learning interpretability methods", Linardatos et al. [7] 2) "A symbolic approach to explaining bayesian network classifiers", Shih et al. [13] 3) "A value for n-person games" Shapley [12]

Interpretations Rely on a Model of the Data Manifold

Interpretations Rely on a Model of the Data Manifold

- Approach from Lundberg et al[1]:

$$\Phi_{f;\mathbf{x}}(S) = \mathbb{E}_{\mathbf{y}}[f(\mathbf{y}) \mid \mathbf{y}_S = \mathbf{x}_S] = \int f(\mathbf{x}) \, dP[\mathbf{x}_{\bar{S}} \mid \mathbf{x}_S].$$

1) "A unified approach to interpreting model predictions", Lundberg et al, [8]

Interpretations Rely on a Model of the Data Manifold

- Approach from Lundberg et al[1]:

$$\Phi_{f;\mathbf{x}}(S) = \mathbb{E}_{\mathbf{y}}[\ f(\mathbf{y}) \mid \mathbf{y}_S = \mathbf{x}_S] = \int f(\mathbf{x}) \mathrm{d}P[\mathbf{x}_{S^c} \mid \mathbf{x}_S]:$$

- Needs good model of $P[\mathbf{x}_{S^c} \mid \mathbf{x}_S]$!

1) "A unified approach to interpreting model predictions", Lundberg et al, [8]

Interpretations Rely on a Model of the Data Manifold

- Approach from Lundberg et al[1]:

$$\Phi_{f;\mathbf{x}}(S) = \mathbb{E}_{\mathbf{y}}[f(\mathbf{y}) \mid \mathbf{y}_S = \mathbf{x}_S] = \int f(\mathbf{x}) \mathrm{d}P[\mathbf{x}_{S^c} \mid \mathbf{x}_S].$$

- Needs good model of $P[\mathbf{x}_{S^c} \mid \mathbf{x}_S]$!
- Most methods simply approximate with baseline values (sometimes layer-wise)

1) "A unified approach to interpreting model predictions", Lundberg et al, [8]

Interpretations Rely on a Model of the Data Manifold

- Approach from Lundberg et al[1]:

$$\Phi_{i;\mathbf{x}}(S) = \mathrm{E}_{\mathbf{y}}[\ (\mathbf{y})\, j\, \mathbf{y}_S = \mathbf{x}_S] = \int (\mathbf{x}) \mathrm{dP}[\mathbf{x}_{S^c} j \mathbf{x}_S]:$$

- Needs good model of $\mathrm{P}[\mathbf{x}_{S^c} j \mathbf{x}_S]$!
- Most methods simply approximate with baseline values (sometimes layer-wise)
- Change o -manifold behaviour to manipulate: Gradient, Integrated gradients[2;3], LRP[2;4;7], LIME[3;5], DeepShap[3;5], Grad-Cam[7], Shapley-based[6], Counterfactual explanations[8],

1) "A unified approach to interpreting model predictions", Lundberg et al, [8] 2) "Fairwashing explanations with off-manifold detergent", Anders et al. [1] 3) "You Shouldn't Trust Me: Learning Models Which Conceal Unfairness From Multiple Explanation Methods", Dimanov et al. [3] 4) "Explanations can be manipulated and geometry is to blame" Dombrowski et al. [4] 5) "Fooling lime and shap: Adversarial attacks on post hoc explanation methods", Slack et al. [14] 6) "Shapley explainability on the data manifold", Frye et al. [5] 7) "Fooling neural network interpretations via adversarial model manipulation" Heo et al. [6] 8) "Counterfactual Explanations Can Be Manipulated" Slack et al. [15]

Interpretations Rely on a Model of the Data Manifold

- Approach from Lundberg et al[1]:

$$\Phi_{i;\mathbf{x}}(S) = \mathrm{E}_{\mathbf{y}}[\ (\mathbf{y}) j\, \mathbf{y}_S = \mathbf{x}_S] = \int (\mathbf{x}) \mathrm{dP}[\mathbf{x}_{S^c} j \mathbf{x}_S]:$$

- Needs good model of $\mathrm{P}[\mathbf{x}_{S^c} j \mathbf{x}_S]$!
- Most methods simply approximate with baseline values (sometimes layer-wise)
- Change o -manifold behaviour to manipulate: Gradient, Integrated gradients[2;3], LRP[2;4;7], LIME[3;5], DeepShap[3;5], Grad-Cam[7], Shapley-based[6], Counterfactual explanations[8],

Image FW AFW

LCG LAFW Sensitivity

- Best Performer RDE creates new features![8]

1) "A unified approach to interpreting model predictions", Lundberg et al, [8] 2) "Fairwashing explanations with off-manifold detergent", Anders et al. [1] 3) "You Shouldn't Trust Me: Learning Models Which Conceal Unfairness From Multiple Explanation Methods", Dimanov et al. [3] 4) "Explanations can be manipulated and geometry is to blame" Dombrowski et al. [4] 5) "Fooling lime and shap: Adversarial attacks on post hoc explanation methods", Slack et al. [14] 6) "Shapley explainability on the data manifold", Frye et al. [5] 7) "Fooling neural network interpretations via adversarial model manipulation" Heo et al. [6] 8) "Counterfactual Explanations Can Be Manipulated" Slack et al. [15] 9) "Interpretable Neural Networks with Frank-Wolfe: Sparse Relevance Maps and Relevance Orderings", Macdonald et al. [9]

Interpretations Rely on a Model of the Data Manifold

- Approach from Lundberg et al[1]:

$$\Phi_{i;\mathbf{x}}(S) = \mathrm{E}_{\mathbf{y}}[\ (\mathbf{y})\, j\, \mathbf{y}_S = \mathbf{x}_S] = \int (\mathbf{x}) \mathrm{dP}[\mathbf{x}_{S^c} j \mathbf{x}_S]:$$

- Needs good model of $\mathrm{P}[\mathbf{x}_{S^c} j \mathbf{x}_S]$!
- Most methods simply approximate with baseline values (sometimes layer-wise)
- Change o -manifold behaviour to manipulate: Gradient, Integrated gradients[2;3], LRP[2;4;7], LIME[3;5], DeepShap[3;5], Grad-Cam[7], Shapley-based[6], Counterfactual explanations[8],

Image FW AFW

LCG LAFW Sensitivity

- Idea: Directly train a characteristic function!

1) "A unified approach to interpreting model predictions", Lundberg et al, [8] 2) "Fairwashing explanations with off-manifold detergent", Anders et al. [1] 3) "You Shouldn't Trust Me: Learning Models Which Conceal Unfairness From Multiple Explanation Methods", Dimanov et al. [3] 4) "Explanations can be manipulated and geometry is to blame" Dombrowski et al. [4] 5) "Fooling lime and shap: Adversarial attacks on post hoc explanation methods", Slack et al. [14] 6) "Shapley explainability on the data manifold", Frye et al. [5] 7) "Fooling neural network interpretations via adversarial model manipulation" Heo et al. [6] 8) "Counterfactual Explanations Can Be Manipulated" Slack et al. [15] 9) "Interpretable Neural Networks with Frank-Wolfe: Sparse Relevance Maps and Relevance Orderings", Macdonald et al. [9]

Setup: Connect Four with hidden colour information

Setup: Connect Four with hidden colour information

- Abstract Games: complex, yet low-dimensional

1) “Proximal policy optimization algorithms”, Schulman et al [11] 2) “Reinforcement Learning for Two-Player Zero-Sum Games”, Crespo [2]

Setup: Connect Four with hidden colour information

- Abstract Games: complex, yet low-dimensional

1) \Proximal policy optimization algorithms", Schulman et al [11] 2) \Reinforcement Learning for Two-Player Zero-Sum Games", Crespo [2]

Setup: Connect Four with hidden colour information

- Abstract Games: complex, yet low-dimensional
- Every turn t: $p_h \sim U([0; p_h^{max}])$

1) \Proximal policy optimization algorithms", Schulman et al [11] 2) \Reinforcement Learning for Two-Player Zero-Sum Games", Crespo [2]

Setup: Connect Four with hidden colour information

- Abstract Games: complex, yet low-dimensional
- Every turn t: $p_h \sim U([0; p_h^{max}])$
- Hide both p_h at t colour features at random

1) \Proximal policy optimization algorithms", Schulman et al [11] 2) \Reinforcement Learning for Two-Player Zero-Sum Games", Crespo [2]

Setup: Connect Four with hidden colour information

- Abstract Games: complex, yet low-dimensional
- Every turn t: $p_h \sim U([0; p_h^{max}])$
- Hide both p_h at t colour features at random

1) \Proximal policy optimization algorithms", Schulman et al [11] 2) \Reinforcement Learning for Two-Player Zero-Sum Games", Crespo [2]

Setup: Connect Four with hidden colour information

- Abstract Games: complex, yet low-dimensional
- Every turn t: $p_h \sim U([0; p_h^{max}])$
- Hide both colour features at random
- Train agent with Proximal Policy Optimisation (PPO)[1,2]

1) \Proximal policy optimization algorithms", Schulman et al [11] 2) \Reinforcement Learning for Two-Player Zero-Sum Games", Crespo [2]

Setup: Connect Four with hidden colour information

- Abstract Games: complex, yet low-dimensional
- Every turn t: $p_h \sim U([0; p_h^{max}])$
- Hide both p_h at colour features at random
- Train agent with Proximal Policy Optimisation (PPO)[1,2]

FI: Full Information.

1) \Proximal policy optimization algorithms", Schulman et al [11] 2) \Reinforcement Learning for Two-Player Zero-Sum Games", Crespo [2]

Setup: Connect Four with hidden colour information

- Abstract Games: complex, yet low-dimensional
- Every turn t: $p_h \sim U([0; p_h^{max}])$
- Hide both colour features at random
- Train agent with Proximal Policy Optimisation (PPO)[1,2]

FI: Full Information.

PI-50: with $p_h \sim U([0; 0.5])$

1) \Proximal policy optimization algorithms", Schulman et al [11] 2) \Reinforcement Learning for Two-Player Zero-Sum Games", Crespo [2]

Setup: Connect Four with hidden colour information

- Abstract Games: complex, yet low-dimensional
- Every turn t: $p_h \sim U([0; p_h^{max}])$
- Hide both colour features at random
- Train agent with Proximal Policy Optimisation (PPO)[1,2]

FI: Full Information.

PI-50: with $p_h \sim U([0; 0.5])$

PI-100: with $p_h \sim U([0; 1])$

1) \Proximal policy optimization algorithms", Schulman et al [11] 2) \Reinforcement Learning for Two-Player Zero-Sum Games", Crespo [2]

Setup: Connect Four with hidden colour information

- Abstract Games: complex, yet low-dimensional
- Every turn t: $p_h \sim U([0; p_h^{max}])$
- Hide both colour features at random
- Train agent with Proximal Policy Optimisation (PPO)[1,2]

FI: Full Information.

PI-50: with $p_h \sim U([0; 0.5])$

PI-100: with $p_h \sim U([0; 1])$

1) \Proximal policy optimization algorithms", Schulman et al [11] 2) \Reinforcement Learning for Two-Player Zero-Sum Games", Crespo [2]

Setup: Connect Four with hidden colour information

- Abstract Games: complex, yet low-dimensional
- Every turn t: $p_h \sim U([0; p_h^{max}])$
- Hide both colour features at random
- Train agent with Proximal Policy Optimisation (PPO)[1,2]

FI: Full Information.

PI-50: with $p_h \sim U([0; 0.5])$

PI-100: with $p_h \sim U([0; 1])$

1) \Proximal policy optimization algorithms", Schulman et al [11] 2) \Reinforcement Learning for Two-Player Zero-Sum Games", Crespo [2]

Interpretability with Characteristic Functions

1) \Deep Neural Network Training with Frank-Wolfe", Pokutta et al. [10]

Interpretability with Characteristic Functions

- Let $t \geq 2$ [42]

1) \Deep Neural Network Training with Frank-Wolfe", Pokutta et al. [10]

Interpretability with Characteristic Functions

I Let t 2 [42], x 2 $[0; 1]^3$ 6 7

1) \Deep Neural Network Training with Frank-Wolfe", Pokutta et al. [10]

Interpretability with Characteristic Functions

I Let $t \in [42]$, $x \in [0;1]^{3\ 6\ 7}$, $S \in [t]$

1) \Deep Neural Network Training with Frank-Wolfe", Pokutta et al. [10]

Interpretability with Characteristic Functions

I Let $t \in [42]$, $x \in [0;1]^{3 \times 6 \times 7}$, $S \in [t]$ and let $x^{(S)}$ be state with colour feature of S^c hidden

1) \Deep Neural Network Training with Frank-Wolfe", Pokutta et al. [10]

Interpretability with Characteristic Functions

- Let $t \in [42]$, $x \in [0;1]^{3 \times 6 \times 7}$, $S \in [t]$ and let $x^{(S)}$ be state with colour feature on S^c hidden
- Let furthermore $a = \operatorname{argmax}_a P(a;x)$

1) \Deep Neural Network Training with Frank-Wolfe", Pokutta et al. [10]

Interpretability with Characteristic Functions

- Let $t \in [42]$, $x \in [0;1]^{3 \times 6 \times 7}$, $S \in [t]$ and let $x^{(S)}$ be state with colour feature on S^c hidden
- Let furthermore $a = \operatorname{argmax}_a P(a;x)$
- We define $\pi^{\text{pol}} : 2^{[t]} \to [0;1]$ and $\pi^{\text{val}} : 2^{[t]} \to [-1;1]$ as

$$\pi^{\text{pol}}(S) = P(a;x^{(S)}) \quad \text{and} \quad \pi^{\text{val}}(S) = V(x^{(S)}):$$

1) \Deep Neural Network Training with Frank-Wolfe", Pokutta et al. [10]

Interpretability with Characteristic Functions

- Let $t \in [42]$, $x \in [0;1]^{3 \times 6 \times 7}$, $S \in [t]$ and let $x^{(S)}$ be state with colour feature on S^c hidden
- Let furthermore $a = \operatorname{argmax}_a P(a;x)$
- We define $\pi_{\text{pol}} : 2^{[t]} \to [0;1]$ and $\pi_{\text{val}} : 2^{[t]} \to [-1;1]$ as

$$\pi_{\text{pol}}(S) = P(a;x^{(S)}) \quad \text{and} \quad \pi_{\text{val}}(S) = V(x^{(S)}):$$

1) \Deep Neural Network Training with Frank-Wolfe", Pokutta et al. [10]

Interpretability with Characteristic Functions

- Let $t \in [42]$, $x \in [0;1]^{3 \times 6 \times 7}$, $S \in [t]$ and let $x^{(S)}$ be state with colour feature on S^c hidden
- Let furthermore $a = \operatorname{argmax}_a P(a; x)$
- We define $\nu_{\text{pol}} : 2^{[t]} \to [0;1]$ and $\nu_{\text{val}} : 2^{[t]} \to [-1;1]$ as

$$\nu_{\text{pol}}(S) = P(a; x^{(S)}) \quad \text{and} \quad \nu_{\text{val}}(S) = V(x^{(S)}):$$

- We can approximate the Shapley sum by sampling from $\mathcal{U}([t])$:

$$\Phi_i = \frac{1}{t!} \sum_{\pi \in \Pi([t])^{\text{sample}}} \left(\nu(P_i^\pi \cup \{i\}) - \nu(P_i^\pi) \right):$$

1) \Deep Neural Network Training with Frank-Wolfe", Pokutta et al. [10]

Interpretability with Characteristic Functions

- Let $t \in [42]$, $x \in [0;1]^{3 \cdot 6 \cdot 7}$, $S \in [t]$ and let x_S be state with colour feature on S^c hidden
- Let furthermore $a = \operatorname{argmax}_a P(a;x)$
- We define $\nu_{pol} : 2^{[t]} \to [0;1]$ and $\nu_{val} : 2^{[t]} \to [-1;1]$ as

$$\nu_{pol}(S) = P(a;x_S) \quad \text{and} \quad \nu_{val}(S) = V(x_S):$$

- We can approximate the Shapley sum by sampling from $\mathcal{U}([t])$:

$$\Phi_i = \frac{1}{t!} \sum_{S \in ([t])^{\text{sample}}} \left(\nu(P_i \cup [i]) - \nu(P_i)\right):$$

- $P[|\Phi_i - \Phi_i| \leq 1 - \delta]$ $(0.01, 0.01)$-approximation $\approx 26\,500$ samples (Hoeffding)

1) "Deep Neural Network Training with Frank-Wolfe", Pokutta et al. [10]

Interpretability with Characteristic Functions

- Let $t \in [42]$, $x \in [0;1]^{3 \times 6 \times 7}$, $S \in [t]$ and let $x^{(S)}$ be state with colour feature of S^c hidden
- Let furthermore $a = \operatorname{argmax}_a P(a; x)$
- We define $\nu^{pol} : 2^{[t]} \to [0;1]$ and $\nu^{val} : 2^{[t]} \to [-1;1]$ as

$$\nu^{pol}(S) = P(a; x^{(S)}) \quad \text{and} \quad \nu^{val}(S) = V(x^{(S)}):$$

- We can approximate the Shapley sum by sampling from $\mathcal{U}([t])$:

$$\phi_i = \frac{1}{t!} \sum_{S \in ([t])^{sample}} \left(\nu(P_i \cup [f \setminus i]) - \nu(P_i \setminus [f \setminus i])\right):$$

- $P(|\phi_i - \hat{\phi}_i| \leq 1) \geq 0.99$ $(0.01, 0.01)$-approximation $\approx 26\,500$ samples (Hoeffding)
- Calculate PIE with Frank-Wolfe optimiser solving convex relaxation of[1]

$$S = \operatorname*{argmin}_{j \in \mathbb{R}^{|[t]|}} \|\phi([t]) - \nu(S)\|_2^2:$$

1) "Deep Neural Network Training with Frank-Wolfe", Pokutta et al. [10]

Example Saliencies for di erent Methods

Information-Performance Comparison

Round-Robin Tournament

Round-Robin Tournament

Round-Robin Tournament

Limitations and Outlook

Limitations and Outlook

- So far works only for certain abstract games (Connect Four, Hex, Go, ...)

Limitations and Outlook

- So far works only for certain abstract games (Connect Four, Hex, Go, ...)
- Filter out illegal moves with model-based approaches (see e.g. AlphaGo)

Limitations and Outlook

- So far works only for certain abstract games (Connect Four, Hex, Go, ...)
- Filter out illegal moves with model-based approaches (see e.g. AlphaGo)
- Further extension to real-world tasks (high-dimensional, high redundancy) is challenging

Limitations and Outlook

- So far works only for certain abstract games (Connect Four, Hex, Go, ...)
- Filter out illegal moves with model-based approaches (see e.g. AlphaGo)
- Further extension to real-world tasks (high-dimensional, high redundancy) is challenging
- Our approach is should be used primarily for evaluation of saliency methods

Limitations and Outlook

- So far works only for certain abstract games (Connect Four, Hex, Go, ...)
- Filter out illegal moves with model-based approaches (see e.g. AlphaGo)
- Further extension to real-world tasks (high-dimensional, high redundancy) is challenging
- Our approach is should be used primarily for evaluation of saliency methods
- Shapley sampling su ered from unstable policy layer for large hidden information

Limitations and Outlook

- So far works only for certain abstract games (Connect Four, Hex, Go, ...)
- Filter out illegal moves with model-based approaches (see e.g. AlphaGo)
- Further extension to real-world tasks (high-dimensional, high redundancy) is challenging
- Our approach is should be used primarily for evaluation of saliency methods
- Shapley sampling su ered from unstable policy layer for large hidden information
- Train value function instead

Limitations and Outlook

- So far works only for certain abstract games (Connect Four, Hex, Go, ...)
- Filter out illegal moves with model-based approaches (see e.g. AlphaGo)
- Further extension to real-world tasks (high-dimensional, high redundancy) is challenging
- Our approach is should be used primarily for evaluation of saliency methods
- Shapley sampling su ered from unstable policy layer for large hidden information
- Train value function instead
- Q-Learning could be a more stable approach

Conclusion:

Conclusion:

- Interpretability relies on a good model of the data distribution

Conclusion:

- Interpretability relies on a good model of the data distribution
- We can design proxy-task where we know the distribution via abstract games with missing information

Conclusion:

- Interpretability relies on a good model of the data distribution
- We can design proxy-task where we know the distribution
  via abstract games with missing information
- Use these tasks to evaluate saliency methods without going o -manifold

Conclusion:

- Interpretability relies on a good model of the data distribution
- We can design proxy-task where we know the distribution via abstract games with missing information
- Use these tasks to evaluate saliency methods without going o -manifold

Thank You!

Conclusion:

- Interpretability relies on a good model of the data distribution
- We can design proxy-task where we know the distribution via abstract games with missing information
- Use these tasks to evaluate saliency methods without going o -manifold

Thank You!

Contact: waeldchen@zib.de

Paper: Training Characteristic Functions with Reinforcement Learning: XAI-methods play Connect Four
S Wäldchen, F Huber, S Pokutta
arXiv preprint arXiv:2202.11797

References i

C. Anders, P. Pasliev, A.-K. Dombrowski, K.-R. Müller, and P. Kessel.
Fairwashing explanations with off-manifold detergent.
In International Conference on Machine Learning, pages 314–323. PMLR, 2020.

J. Crespo.
Reinforcement learning for two-player zero-sum games.
Master's thesis, Tecnico Lisboa, https://fenix.tecnico.ulisboa.pt/downloadFile/1689244997260153/81811-joao-crespo_dissertacao.pdf, 2019.

B. Dimanov, U. Bhatt, M. Jamnik, and A. Weller.
You shouldn't trust me: Learning models which conceal unfairness from multiple explanation methods.
In SafeAI@ AAAI, 2020.

References ii

A.-K. Dombrowski, M. Alber, C. J. Anders, M. Ackermann, K.-R. Müller, and P. Kessel.
Explanations can be manipulated and geometry is to blame.
arXiv preprint arXiv:1906.07983, 2019.

C. Frye, D. de Mijolla, T. Begley, L. Cowton, M. Stanley, and I. Feige.
Shapley explainability on the data manifold.
arXiv preprint arXiv:2006.01272, 2020.

J. Heo, S. Joo, and T. Moon.
Fooling neural network interpretations via adversarial model manipulation.
Advances in Neural Information Processing Systems, 32:2925–2936, 2019.

P. Linardatos, V. Papastefanopoulos, and S. Kotsiantis.
Explainable ai: A review of machine learning interpretability methods.
Entropy, 23(1):18, 2020.

References iii

S. M. Lundberg and S.-I. Lee.
A uni ed approach to interpreting model predictions.
In Proceedings of the 31st international conference on neural information processing systems, pages 4768{4777, 2017.

J. Macdonald, M. Besancon, and S. Pokutta.
Interpretable neural networks with frank-wolfe: Sparse relevance maps and relevance orderings.
arXiv preprint arXiv:2110.08105, 2021.

S. Pokutta, C. Spiegel, and M. Zimmer.
Deep neural network training with frank-wolfe.
arXiv preprint arXiv:2010.07243, 2020.

J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov.
Proximal policy optimization algorithms.
arXiv preprint arXiv:1707.06347, 2017.

References iv

L. S. Shapley.
17. A value for n-person games.
Princeton University Press, 2016.

A. Shih, A. Choi, and A. Darwiche.
A symbolic approach to explaining bayesian network classi ers.
arXiv preprint arXiv:1805.03364, 2018.

D. Slack, S. Hilgard, E. Jia, S. Singh, and H. Lakkaraju.
Fooling lime and shap: Adversarial attacks on post hoc explanation methods.
In Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society, pages 180{186, 2020.

D. Slack, S. Hilgard, H. Lakkaraju, and S. Singh.
Counterfactual explanations can be manipulated.
arXiv preprint arXiv:2106.02666, 2021.

Appendix

Ground Truth Comparison: Winning Move

Tournament: Standard Deviation and Illegal Move Rate