ZUSE INSTITUTE BERLIN

Merlin-Arthur Classifiers: Formal Interpretability with Interactive Black Boxes

DMV 2022

Kartikey Sharma

15th September 2022

Results are joint work with...

Stephan Wäldchen
Zuse Institute Berlin

Sebastian Pokutta
Zuse Institute Berlin

Max Zimmer
Zuse Institute Berlin

1. Introduction
2. Theoretical Framework
3. Experiments

Explainable AI

Motivation

- Neural Networks form a key part of AI
- Outcomes difficult to explain

Consequences

- Existence of hidden biases and vulnerabilities
- Lower trust

Our Contribution

- Intepretable classification system with theoretical guarantees on features.

Explainable AI

Motivation

- Neural Networks form a key part of AI
- Outcomes difficult to explain

Consequences

- Existence of hidden biases and vulnerabilities
- Lower trust

Our Contribution

- Intepretable classification system with theoretical guarantees on features.

Explainable AI

Motivation

- Neural Networks form a key part of AI
- Outcomes difficult to explain

Consequences

- Existence of hidden biases and vulnerabilities
- Lower trust

Our Contribution

- Intepretable classification system with theoretical guarantees on features.

Existing Literature

Heuristic Approaches

- Saliency maps (Mohseni, Zarei, and Ragan 2021), Mechanistic interpretability (Olah et al. 2018).
- Their success cannot be verified. Can be manipulated by a clever design of the NN (Slack, Hilgard, Lakkaraju, et al. 2021; Slack, Hilgard, Jia, et al. 2020; Anders et al. 2020).

Formal Approaches

- Can run into complexity problems, require an exponential amount of time (Macdonald et al. 2020; Ignatiev, Narodytska, and Marques-Silva 2019).

Existing Literature

Heuristic Approaches

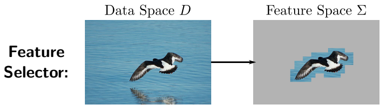
- Saliency maps (Mohseni, Zarei, and Ragan 2021), Mechanistic interpretability (Olah et al. 2018).
- Their success cannot be verified. Can be manipulated by a clever design of the NN (Slack, Hilgard, Lakkaraju, et al. 2021; Slack, Hilgard, Jia, et al. 2020; Anders et al. 2020).

Formal Approaches

- Can run into complexity problems, require an exponential amount of time (Macdonald et al. 2020; Ignatiev, Narodytska, and Marques-Silva 2019).

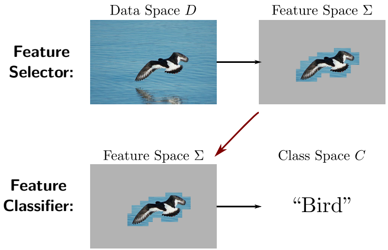
Background

- **Task:** For $\mathbf{x} \in D$, select feature $\phi \in \Sigma$.
- ϕ should have high mutual information to the class $c(\mathbf{x}) \in C$
- Can we lower-bound $I(c(\mathbf{x}); \text{“}\mathbf{x} \text{ contains } \phi\text{”})$?
- **Problem:** Would require model of data manifold with bound on error
- **Idea:** Retrain on selected features
- **Problem:** Cheating!



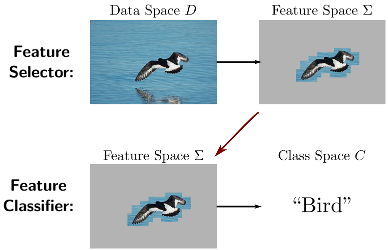
Background

- **Task:** For $\mathbf{x} \in D$, select feature $\phi \in \Sigma$.
- ϕ should have high mutual information to the class $c(\mathbf{x}) \in C$
- Can we lower-bound $I(c(\mathbf{x}); \text{“}\mathbf{x} \text{ contains } \phi\text{”})$?
- **Problem:** Would require model of data manifold with bound on error
- **Idea:** Retrain on selected features
- **Problem:** Cheating!



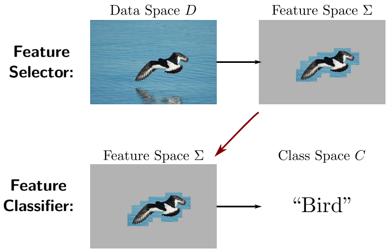
Background

- **Task:** For $\mathbf{x} \in D$, select feature $\phi \in \Sigma$.
- ϕ should have high mutual information to the class $c(\mathbf{x}) \in C$
- Can we lower-bound $I(c(\mathbf{x}); \text{“}\mathbf{x} \text{ contains } \phi\text{”})$?
- **Problem:** Would require model of data manifold with bound on error
- **Idea:** Retrain on selected features
- **Problem:** Cheating!

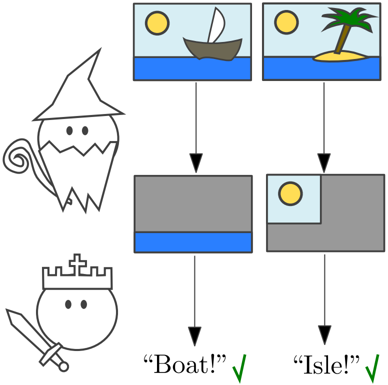


Background

- **Task:** For $\mathbf{x} \in D$, select feature $\phi \in \Sigma$.
- ϕ should have high mutual information to the class $c(\mathbf{x}) \in C$
- Can we lower-bound $I(c(\mathbf{x}); \text{“}\mathbf{x} \text{ contains } \phi\text{”})$?
- **Problem:** Would require model of data manifold with bound on error
- **Idea:** Retrain on selected features
- **Problem:** Cheating!



Cheating



Original Images:

$$P(C = \text{“boat”} \mid \text{“sea”}) = 0.5$$
$$P(C = \text{“isle”} \mid \text{“sea”}) = 0.5$$

$$I(C\,;\,\text{“sea”}) = 0$$

Masked Images:

$$P(C = \text{“boat”} \mid \text{“sea”}) = 1$$
$$P(C = \text{“isle”} \mid \text{“sea”}) = 0$$

$$I(C\,;\,\text{“sea”}) = 1$$

Methodology: Merlin-Arthur Classification

- Based on Merlin-Arthur protocols from Interactive Proof Systems
- Cooperative feature selector / Prover (Merlin): M
- Adversarial feature selector / Prover (Morgana): $\widehat{M}$
- Classifier / Verifier (Arthur): A
- Arthur should leverage Merlin but not be misled by Morgana

Methodology: Merlin-Arthur Classification

- Based on Merlin-Arthur protocols from Interactive Proof Systems
- Cooperative feature selector / Prover (Merlin): M
- Adversarial feature selector / Prover (Morgana): $\widehat{M}$
- Classifier / Verifier (Arthur): A
- Arthur should leverage Merlin but not be misled by Morgana

Methodology: Merlin-Arthur Classification

- Based on Merlin-Arthur protocols from Interactive Proof Systems
- Cooperative feature selector / Prover (Merlin): M
- Adversarial feature selector / Prover (Morgana): $\widehat{M}$
- Classifier / Verifier (Arthur): A
- Arthur should leverage Merlin but not be misled by Morgana

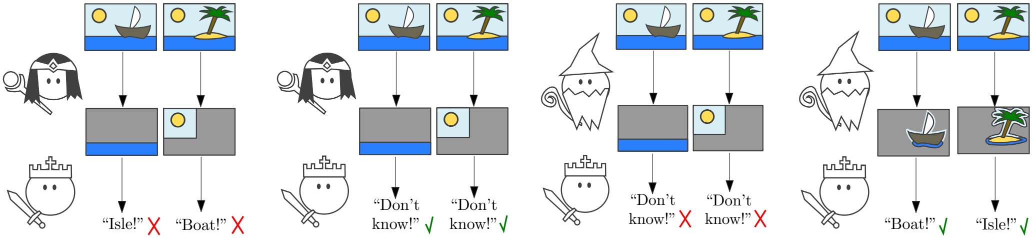
Methodology: Merlin-Arthur Classification

- Based on Merlin-Arthur protocols from Interactive Proof Systems
- Cooperative feature selector / Prover (Merlin): M
- Adversarial feature selector / Prover (Morgana): $\widehat{M}$
- Classifier / Verifier (Arthur): A
- Arthur should **leverage Merlin** but not be misled by Morgana

Methodology: Merlin-Arthur Classification

- Based on Merlin-Arthur protocols from Interactive Proof Systems
- Cooperative feature selector / Prover (Merlin): M
- Adversarial feature selector / Prover (Morgana): $\widehat{M}$
- Classifier / Verifier (Arthur): A
- Arthur should **leverage Merlin** but **not be misled by Morgana**

Cheating with Morgana



Average Precision

Definition

Completeness: $\min_{l \in \{-1,1\}} \mathbb{P}_{\mathbf{x} \sim \mathcal{D}_l}[A(M(\mathbf{x})) = c(\mathbf{x})] \geq 1 - \epsilon_c,$

Soundness: $\max_{l \in \{-1,1\}} \mathbb{P}_{\mathbf{x} \sim \mathcal{D}_l}\left[A\left(\widehat{M}(\mathbf{x})\right) = -c(\mathbf{x})\right] \leq \epsilon_s.$

Definition

Given a feature selector $M \in \mathcal{M}(\mathcal{D})$, the average precision of $\mathcal{M}$ with respect to the data distribution $\mathcal{D}$ is

$$Q_{\mathcal{D}}(M) := \mathbb{E}_{\mathbf{x} \sim D}[\mathbb{P}_{\mathbf{y} \sim D}[c(\mathbf{y}) = c(\mathbf{x}) | \mathbf{y} \in M(\mathbf{x})]]$$

Average Precision

Definition

$$\text{Completeness:} \quad \min_{l \in \{-1,1\}} \mathbb{P}_{\mathbf{x} \sim \mathcal{D}_l}[A(M(\mathbf{x})) = c(\mathbf{x})] \geq 1 - \epsilon_c,$$

$$\text{Soundness:} \quad \max_{l \in \{-1,1\}} \mathbb{P}_{\mathbf{x} \sim \mathcal{D}_l}\left[A\left(\widehat{M}(\mathbf{x})\right) = -c(\mathbf{x})\right] \leq \epsilon_s.$$

Definition

Given a feature selector $M \in \mathcal{M}(\mathcal{D})$, the average precision of $\mathcal{M}$ with respect to the data distribution $\mathcal{D}$ is

$$Q_{\mathcal{D}}(M) := \mathbb{E}_{\mathbf{x} \sim \boldsymbol{D}}[\mathbb{P}_{\mathbf{y} \sim \boldsymbol{D}}[c(\mathbf{y}) = c(\mathbf{x}) | \mathbf{y} \in M(\mathbf{x})]]$$

Performance of Merlin

Classifier: A, Feature Selectors: Merlin M and Morgana $\widehat{M}$

$$E_{M,\widehat{M},A} := \left\{x \in D \,\middle|\, A(M(\mathbf{x})) \neq c(\mathbf{x}) \vee A\left(\widehat{M}(\mathbf{x})\right) = -c(\mathbf{x})\right\},$$

Theorem (Wäldchen 2022+)

Let $M \in \mathcal{M}(D)$ be a feature selector and let

$$\epsilon_M = \min_{A \in \mathcal{A}} \max_{\widehat{M} \in \mathcal{M}} \mathbb{P}_{\mathbf{x} \sim \mathcal{D}}\left[\mathbf{x} \in E_{M,\widehat{M},A}\right].$$

Then there exists a set $D' \subset D$ with $\mathbb{P}_{\mathbf{x} \sim \mathcal{D}}[\mathbf{x} \in D'] \geq 1 - \epsilon_M$ such that for $\mathcal{D}' = \mathcal{D}|_{D'}$ we have

$$Q_{\mathcal{D}'}(M) = 1 \quad \text{and thus} \quad H_{\mathbf{x},\mathbf{y} \sim \mathcal{D}'}(c(\mathbf{y}) \mid \mathbf{y} \in M(\mathbf{x})) = 0.$$

Performance of Merlin

Classifier: A, Feature Selectors: Merlin M and Morgana $\widehat{M}$

$$E_{M,\widehat{M},A} := \left\{x \in D \,\middle|\, A(M(\mathbf{x})) \neq c(\mathbf{x}) \vee A\left(\widehat{M}(\mathbf{x})\right) = -c(\mathbf{x})\right\},$$

Theorem (Wäldchen 2022+)

Let $M \in \mathcal{M}(D)$ be a feature selector and let

$$\epsilon_M = \min_{A \in \mathcal{A}} \max_{\widehat{M} \in \mathcal{M}} \mathbb{P}_{\mathbf{x} \sim \mathcal{D}}\left[\mathbf{x} \in E_{M,\widehat{M},A}\right].$$

Then there exists a set $D' \subset D$ with $\mathbb{P}_{\mathbf{x} \sim \mathcal{D}}[\mathbf{x} \in D'] \geq 1 - \epsilon_M$ such that for $\mathcal{D}' = \mathcal{D}|_{D'}$ we have

$$Q_{\mathcal{D}'}(M) = 1 \quad \text{and thus} \quad H_{\mathbf{x},\mathbf{y} \sim \mathcal{D}'}(c(\mathbf{y}) \mid \mathbf{y} \in M(\mathbf{x})) = 0.$$

Average Precision Bound

Theorem (Wäldchen 2022+)

Let $\mathfrak{D} = ((D, \sigma, \mathcal{D}), c, \Sigma)$ be a two-class data space with AFC of κ and class imbalance B. Let $A \in \mathcal{A}$, M and $\widehat{M} \in \mathcal{M}(D)$ such that $\widehat{M}$ has a context impact of α with respect to A, M and $\mathfrak{D}$. Then it follows that

$$Q_{\mathcal{D}}(M) \geq 1 - \epsilon_c - \frac{\alpha\kappa\epsilon_s}{1 - \epsilon_c + \alpha\kappa\epsilon_s B^{-1}}.$$

Corollary

$$\mathbb{E}_{\mathbf{x}\sim\boldsymbol{D}}[I_{\mathbf{y}\sim\mathcal{D}}(c(\mathbf{y}); \mathbf{y} \in M(\mathbf{x}))] \geq H_{\mathbf{y}\sim\mathcal{D}}(c(\mathbf{y})) - H_b(Q_{\mathcal{D}}(M)).$$

Average Precision Bound

Theorem (Wäldchen 2022+)

Let $\mathfrak{D} = ((D, \sigma, \mathcal{D}), c, \Sigma)$ be a two-class data space with AFC of κ and class imbalance B. Let $A \in \mathcal{A}$, M and $\widehat{M} \in \mathcal{M}(D)$ such that $\widehat{M}$ has a context impact of α with respect to A, M and $\mathfrak{D}$. Then it follows that

$$Q_{\mathcal{D}}(M) \geq 1 - \epsilon_c - \frac{\alpha\kappa\epsilon_s}{1 - \epsilon_c + \alpha\kappa\epsilon_s B^{-1}}.$$

Corollary

$$\mathbb{E}_{\mathbf{x}\sim \boldsymbol{D}}[I_{\mathbf{y}\sim\mathcal{D}}(c(\mathbf{y}); \mathbf{y} \in M(\mathbf{x}))] \geq H_{\mathbf{y}\sim\mathcal{D}}(c(\mathbf{y})) - H_b(Q_{\mathcal{D}}(M)).$$

Experimental Setup

- MNIST Dataset
- **Models**:
  - Merlin and Morgana (Feature Selectors): FW-Classifier and U-Net
  - Arthur (Classifier): Convolutional Neural Network
- **Training process**:
  - Alternate between gradients steps for the masks and for the classifier
  - Alternate between epochs over masked images and regular images

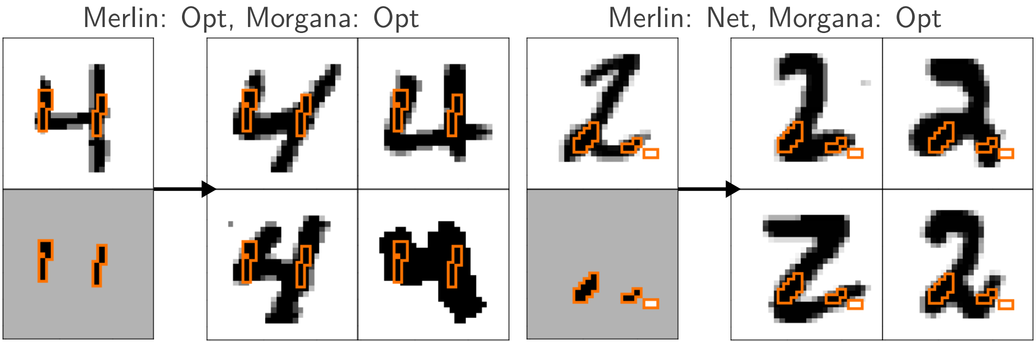
Experimental Setup

- MNIST Dataset
- **Models**:
  - Merlin and Morgana (Feature Selectors): FW-Classifier and U-Net
  - Arthur (Classifier): Convolutional Neural Network
- **Training process**:
  - Alternate between gradients steps for the masks and for the classifier
  - Alternate between epochs over masked images and regular images

Experimental Setup

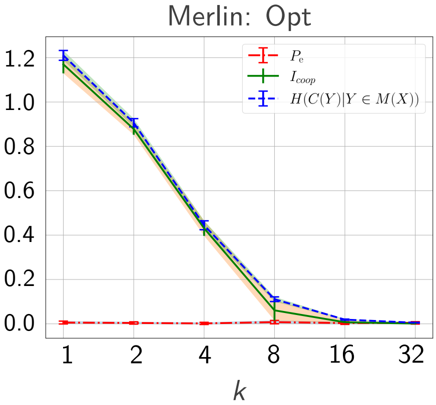
- MNIST Dataset
- **Models**:
  - Merlin and Morgana (Feature Selectors): FW-Classifier and U-Net
  - Arthur (Classifier): Convolutional Neural Network
- **Training process**:
  - Alternate between gradients steps for the masks and for the classifier
  - Alternate between epochs over masked images and regular images

Selected Features



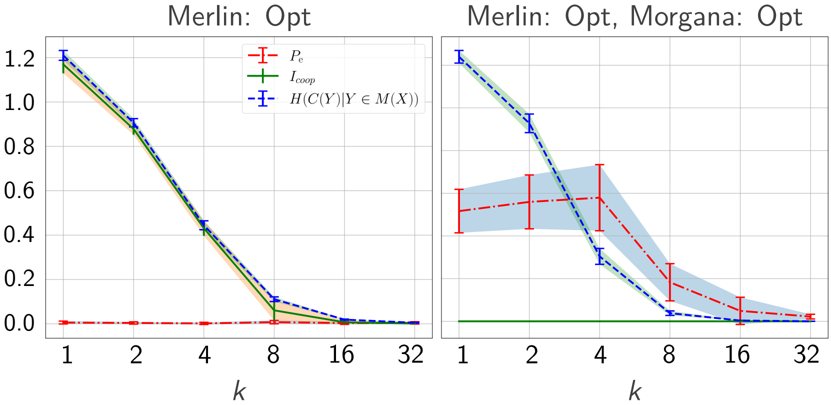
Key Point: Merlin features which tend to be unique to the class when Morgana is present.

Experimental Results



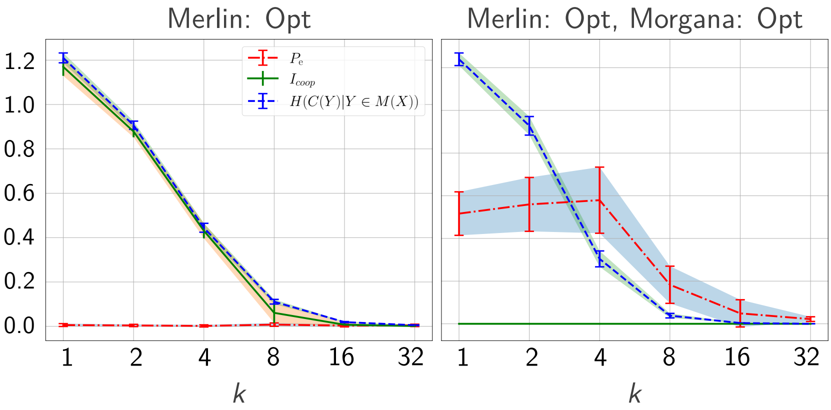
Key Point: Very low error rates even at small pixel sizes for Merlin only. Errors increase when Morgana is present.

Experimental Results



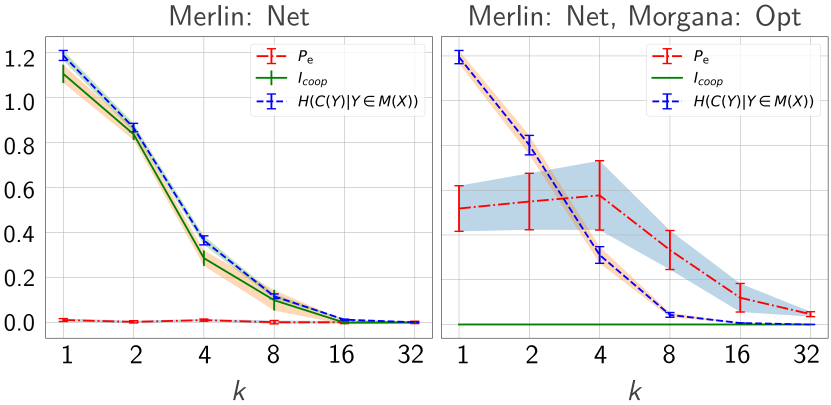
Key Point: Very low error rates even at small pixel sizes for Merlin only. Errors increase when Morgana is present.

Experimental Results

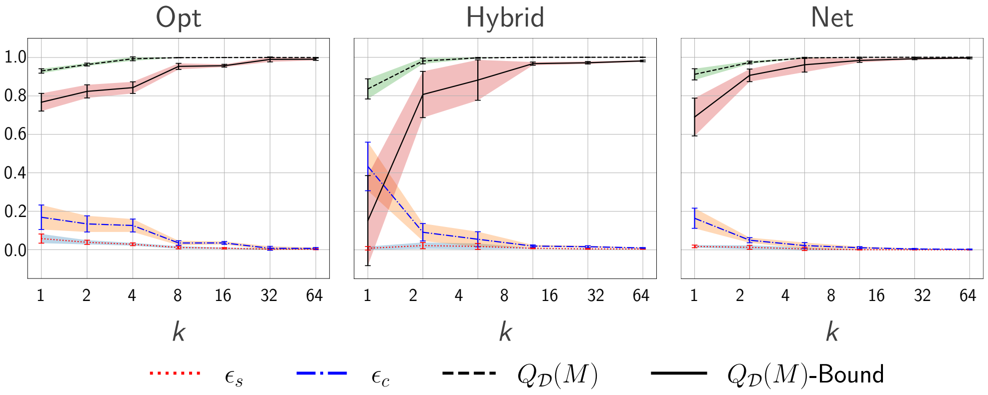


Key Point: Very low error rates even at small pixel sizes for Merlin only. Errors increase when Morgana is present.

Experimental Results



Key Point: Very low error rates even at small pixel sizes for Merlin only. Errors increase when Morgana is present.



Key Point: As mask size increases the bound becomes tighter and the completeness and soundness increase

Conclusion

- We provide an interpretable classification framework inspired by interactive proof systems.
- We achieve guarantees on the mutual information of the features with the class be expressing it in terms of measurable criteria such as completeness and soundness.
- We evaluate our results numerically on the MNIST data set. We observe high quality features which also demonstrate good agreement between our theoretical bounds and the experimental quality of the exchanged features.

Thank you for your attention!

References

- Anders, Christopher et al. (2020). "Fairwashing explanations with off-manifold detergent". In: *International Conference on Machine Learning*. PMLR, pp. 314–323.
- Ignatiev, Alexey, Nina Narodytska, and Joao Marques-Silva (2019). "Abduction-based explanations for machine learning models". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33. 01, pp. 1511–1519.
- Macdonald, Jan et al. (2020). "Explaining neural network decisions is hard". In: *XXAI Workshop, 37th ICML*.
- Mohseni, Sina, Niloofar Zarei, and Eric D Ragan (2021). "A multidisciplinary survey and framework for design and evaluation of explainable AI systems". In: *ACM Transactions on Interactive Intelligent Systems (TiiS)* 11.3-4, pp. 1–45.
- Olah, Chris et al. (2018). "The building blocks of interpretability". In: *Distill* 3.3, e10.

References

Slack, Dylan, Sophie Hilgard, Emily Jia, et al. (2020). "Fooling lime and shap: Adversarial attacks on post hoc explanation methods". In: *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, pp. 180–186.

Slack, Dylan, Sophie Hilgard, Himabindu Lakkaraju, et al. (2021). "Counterfactual Explanations Can Be Manipulated". In: *arXiv preprint arXiv:2106.02666*.

Training Algorithm

Data: Dataset: D, Epochs: N, γ

Result: Classifier (A), Optional: Masking Networks Merlin (M) and Morgana ($\widehat{M}$)

for $i \in [N]$ **do**

  for $\mathbf{x}_j, \mathbf{y}_j \in D$ **do**

    $\mathbf{s}_M \leftarrow M(\mathbf{x}_j, \mathbf{y}_j), \mathbf{s}_{\widehat{M}} \leftarrow \widehat{M}(\mathbf{x}_j, \mathbf{y}_j)$ `M, M can be optimiser or NN`

    $A \leftarrow \arg\min_A (1-\gamma) L_M(A(s_M \cdot x_j), y_j) + \gamma L_{\widehat{M}}(A(s_{\widehat{M}} \cdot x_j), y_j)$ `Update classifier using masked images`

    $M \leftarrow \arg\min L_M(A(M(\mathbf{x}_j) \cdot x_j), y_j)$ `Update only if M is a NN`

    $\widehat{M} \leftarrow \arg\max L_{\widehat{M}}(A(\widehat{M}(\mathbf{x}_j) \cdot x_j), y_j)$ `Update only if M is a NN`

  end

end

for $\mathbf{x}_j, \mathbf{y}_j \in D$ **do**

  $A \leftarrow \arg\min_A L(A(x_j), y_j))$ `Update classifier using regular images`

end

Training Algorithm

Data: Dataset: D, Epochs: N, γ
Result: Classifier (A), Optional: Masking Networks Merlin (M) and Morgana ($\widehat{M}$)
for $i \in [N]$ **do**
  for $\mathbf{x}_j, \mathbf{y}_j \in D$ **do**
    $\mathbf{s}_M \leftarrow M(\mathbf{x}_j, \mathbf{y}_j), \mathbf{s}_{\widehat{M}} \leftarrow \widehat{M}(\mathbf{x}_j, \mathbf{y}_j)$ `M, M̂ can be optimiser or NN`
    $A \leftarrow \arg\min_A (1-\gamma) L_M(A(s_M \cdot x_j), y_j) + \gamma L_{\widehat{M}}(A(s_{\widehat{M}} \cdot x_j), y_j)$ `Update classifier using masked images`
    $M \leftarrow \arg\min L_M(A(M(\mathbf{x}_j) \cdot x_j), y_j)$ `Update only if M is a NN`
    $\widehat{M} \leftarrow \arg\max L_{\widehat{M}}(A(\widehat{M}(\mathbf{x}_j) \cdot x_j), y_j)$ `Update only if M̂ is a NN`
  end
end
for $\mathbf{x}_j, \mathbf{y}_j \in D$ **do**
  $A \leftarrow \arg\min_A L(A(x_j), y_j))$ `Update classifier using regular images`
end

Training Algorithm

Data: Dataset: D, Epochs: N, γ
Result: Classifier (A), Optional: Masking Networks Merlin (M) and Morgana ($\widehat{M}$)
for $i \in [N]$ **do**
  for $\mathbf{x}_j, \mathbf{y}_j \in D$ **do**
    $\mathbf{s}_M \leftarrow M(\mathbf{x}_j, \mathbf{y}_j), \mathbf{s}_{\widehat{M}} \leftarrow \widehat{M}(\mathbf{x}_j, \mathbf{y}_j)$ `M, M̂ can be optimiser or NN`
    $A \leftarrow \arg\min_A (1-\gamma) L_M(A(s_M \cdot x_j), y_j) + \gamma L_{\widehat{M}}(A(s_{\widehat{M}} \cdot x_j), y_j)$ `Update classifier using masked images`
    $M \leftarrow \arg\min L_M(A(M(\mathbf{x}_j) \cdot x_j), y_j)$ `Update only if M is a NN`
    $\widehat{M} \leftarrow \arg\max L_{\widehat{M}}(A(\widehat{M}(\mathbf{x}_j) \cdot x_j), y_j)$ `Update only if M̂ is a NN`
  end
end
for $\mathbf{x}_j, \mathbf{y}_j \in D$ **do**
  $A \leftarrow \arg\min_A L(A(x_j), y_j))$ `Update classifier using regular images`
end