Interpretability Guarantees with Merlin-Arthur Classifiers

Stephan Wäldchen[1] **Kartikey Sharma**[1] **Berkant Turan**[1,2]
Max Zimmer[1,2] **Sebastian Pokutta**[1,2]

[1]Zuse Institute Berlin
[2]Technische Universität Berlin

Abstract

We propose an interactive multi-agent classifier that provides provable interpretability guarantees even for complex agents such as neural networks. These guarantees consist of lower bounds on the mutual information between selected features and the classification decision. Our results are inspired by the Merlin-Arthur protocol from Interactive Proof Systems and express these bounds in terms of measurable metrics such as soundness and completeness. Compared to existing interactive setups, we rely neither on optimal agents nor on the assumption that features are distributed independently. Instead, we use the relative strength of the agents as well as the new concept of Asymmetric Feature Correlation which captures the precise kind of correlations that make interpretability guarantees difficult. We evaluate our results on two small-scale datasets where high mutual information can be verified explicitly.

1 Introduction

Safe deployment of *Neural Network* (NN) based AI systems in high-stakes applications requires that their reasoning be subject to human scrutiny. The field of *Explainable AI* (XAI) has thus put forth a number of interpretability approaches, among them saliency maps [Mohseni et al., 2021], mechanistic interpretability [Olah et al., 2018] and self-explaining networks [Alvarez-Melis and Jaakkola, 2018]. These have had some successes, such as detecting biases in established datasets [Lapuschkin et al., 2019]. However, these approaches are motivated primarily by heuristics and come without any theoretical guarantees. Thus, their success cannot be verified. It has also been demonstrated for numerous XAI-methods that they can be manipulated by a clever design of the NNs [Slack et al., 2021, 2020, Anders et al., 2020, Dimanov et al., 2020]. On the other hand, formal approaches run into complexity barriers when applied to NNs and require an exponential amount of time to guarantee useful properties [Macdonald et al., 2020, Ignatiev et al., 2019]. This makes any "right to explanation," as in the EU's *GDPR* [Goodman and Flaxman, 2017], unenforceable.

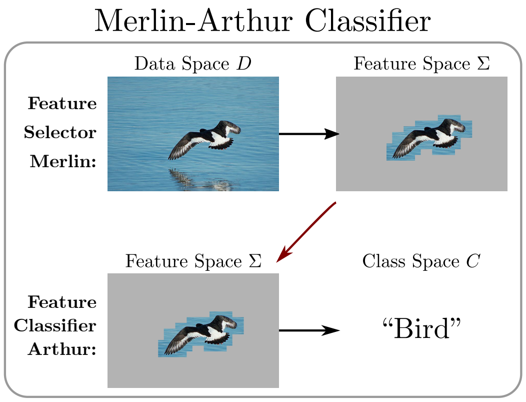


Figure 1: The Merlin-Arthur classifier consists of two interactive agents that communicate over an exchanged feature. This feature serves as an interpretation of the classification.

In this work, we design a classifier that guarantees feature-based interpretability under reasonable assumptions. For this, we connect classification to the *Merlin-Arthur protocol* [Arora and Barak, 2009] from *Interactive Proof Systems* (IPS), see Figure 1. Our setup consists of a *feature classifier* called Arthur (acting as a verifier) and two *feature selectors* referred to as Merlin and Morgana (acting as provers). Merlin and Morgana choose features from the input and send them to Arthur. Merlin aims to send features that

Proceedings of the 27th International Conference on Artificial Intelligence and Statistics (AISTATS) 2024, Valencia, Spain. PMLR: Volume 238. Copyright 2024 by the author(s).

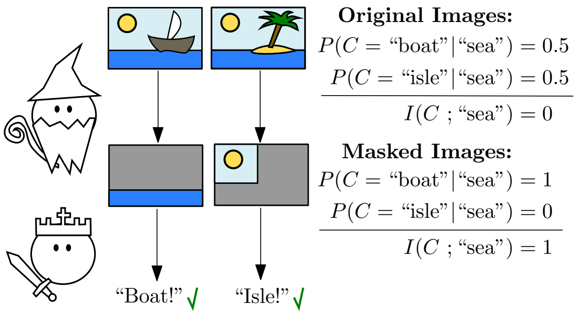

Original Images:
$P(C = \text{“boat”} | \text{“sea”}) = 0.5$
$P(C = \text{“isle”} | \text{“sea”}) = 0.5$
$I(C\,;\text{“sea”}) = 0$
Masked Images:
$P(C = \text{“boat”} | \text{“sea”}) = 1$
$P(C = \text{“isle”} | \text{“sea”}) = 0$
$I(C\,;\text{“sea”}) = 1$
“Boat!” ✓
“Isle!” ✓

Figure 2: Illustration of “cheating” behaviour. In the original dataset, the features “sea” and “sky” appear equally in both classes “boat” and “island”. In the partial images created by Merlin, the “sea” feature appears only in “boat” images and the “sky” feature only for “islands”. Thus, these features now strongly indicate the class of the image. This allows Merlin to communicate the correct class with uninformative features — in contrast to our concept of an interpretable classifier.

cause Arthur to correctly classify the underlying data point. Morgana instead selects features to convince Arthur of the wrong class. Arthur does not know who sent the feature and is allowed to say “Don't know!” if he cannot discern the class. In this context, we can then translate the concepts of *completeness* and *soundness* from IPS to our setting. Completeness describes the probability that Arthur classifies correctly based on features from Merlin. Soundness is the probability that Arthur does not get fooled by Morgana, thus either giving the correct class or answering “Don't know!”. These two quantities can be measured on a test dataset and are used to lower bound the information contained in features selected by Merlin.

1.1 Related Work

Formal approaches to interpretability, such as mutual information [Chen et al., 2018] or Shapley values Frye et al. [2020], generally make use of partial inputs to the classifier. These partial inputs are realised by considering distributions over inputs conditioned on the given information. However, modelling these distributions is difficult for non-synthetic data. This has been pursued practically by training a generative model as in Chattopadhyay et al. [2022]. But as of yet there is no approach that provides a bound on the quality of these models. We discuss these approaches and their challenges in greater detail in Appendix A.2.

Interactive classification in form of a prover-verifier setting has emerged as a way to design inherently interpretable classifiers [Lei et al., 2016, Bastings et al., 2019]. In this setup, the feature selector chooses a feature from a data point and presents it to the classifier who decides the class, see Figure 2. The classification accuracy is meant to guarantee the informativeness of the exchanged features. However, it was noted by Yu et al. that the selector and the classifier can cooperate to achieve high accuracy while communicating over uninformative features, see Figure 2 for an illustration of this “cheating”. Thus, one cannot hope to bound the information content of features via accuracy alone. Chang et al. include an adversarial selector to prevent the cheating. The reasoning is that any “cheating” strategy can be exploited by the adversary to fool the classifier into stating the wrong class, see Figure 3 for an illustration. Anil et al. investigate scenarios in which the three-player setup converges to an equilibrium of perfect completeness and soundness. However, this work assumes that a perfect strategy exists and can be reached through training. For many classification problems, such as the ones we explore in our experimental section, no strategies are perfectly sound and complete when the size of the certificate is limited.

Alternative adversarial setups have been proposed in Yu et al. [2019] and Irving et al. [2018], but without information bounds. We discuss these ideas in detail in Appendix A.4 and show via counterexamples why these formulations cannot straightforwardly yield bounds similar to ours.

An additional theoretical focus has been the learnability of interpretations Goldwasser et al. [2021], Yadav et al. [2022], Poulis and Dasgupta [2017]. In this work, we do not focus on the question of learnability. We instead propose to evaluate soundness and completeness directly on the test dataset, as state-of-the-art models are too complex to guarantee generalisation from a realistic number of training samples.

Chang et al. introduced prover-verifier games for interpretable classification. The authors show that the best strategy for the provers is to select features with high mutual information with respect to the class, and demonstrate that this setup can be stably trained for large-scale text data. However, these results have three restriction that we resolve in this work: **(i)** The features are assumed to be independently distributed. This is an unrealistic assumption for most dataset, including the ones used by Chang et al., since features are generally correlated. In this regime, simply modelling the data distribution directly is possible. **(ii)** The provers can only select one feature at a time without context. This strategy is unlikely to yield useful rationalizations for most types of data where the importance of a feature strongly depends on the features surrounding it, like images and text. Chang et al. do not keep this restriction for their numerical investigation. **(iii)** The result is not quantitative. Since we cannot expect the agents to play optimally on complex data, we need measures of their performance and how this relates to the mutual information of the features.

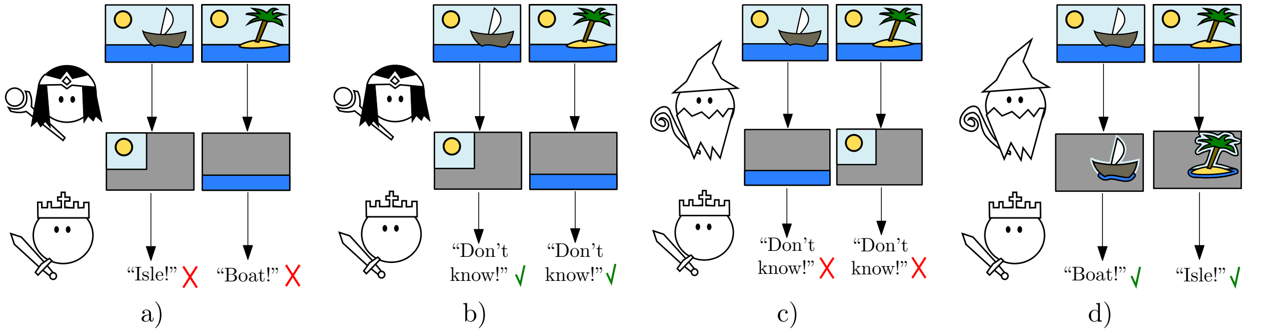


Figure 3: Strategy evolution with Morgana. a) Due to the "cheating" strategy from Figure 2, Arthur expects the "sea" feature for boats and the "sky" for islands. Morgana can exploit this and send the "sky" feature to trick Arthur into classifying a "boat" image as an "island" (and vice versa with "sea"). b) To not be fooled into the wrong class when represented with an ambiguous feature, Arthur refrains from giving a concrete classification. c) Since Arthur does not know who sends the features, he now cannot leverage the uninformative features sent by Merlin. d) Merlin adapts his strategy to only send unambiguous features that cannot be used by Morgana to fool Arthur.

1.2 Contribution

We provide what we believe to be the first quantitative lower bound on the information content of the features in an interpretive setup, eliminating the need to trust a model of the data distribution. Additionally, we improve existing analyses in the following ways:

1. We do not assume our agents to be optimal. In Theorem 2.7, Merlin is allowed to have an arbitrary strategy and in Theorem 2.11, all three players can play suboptimally. We rather rely on the relative strength of Merlin and Morgana for our bound. We also allow our provers to select the features with the context of the full data point.

2. We do not make the assumption that features are independently distributed. Instead, we introduce the concept of Asymmetric Feature Correlation (AFC), which captures the correlations that complicate establishing an information bound. In Theorem 2.7 we circumvent the issue by reducing the dataset, and in Theorem 2.11 we incorporate the AFC explicitly. In Section 4 we discuss why the AFC also matters for other interactive settings.

We numerically demonstrate how the interactive setup prevents a major manipulation that has been demonstrated for other XAI-methods. Finally, we evaluate our theoretical bounds on the MNIST dataset for provers based on Frank-Wolfe optimisers and U-Nets.

2 Theoretical Framework

In this section we develop the theoretical framework for the Merlin-Arthur classifier. What reasonably constitutes a feature strongly depends on the context and prior work often considered subsets of the input as features. W.l.o.g we will stay with this convention for ease of notation. But nothing in our framework relies on these specifics and our theoretical results can be extended to more abstract features as in Chen et al. [2018] or Ribeiro et al. [2018].

We consider abstract datasets $D \subset [0,1]^d$, where D is possibly infinite, e.g., the set of all images of handwritten digits. $\mathcal{D}$ is a distribution on this set. The finite training and test sets, e.g., MNIST, are assumed to be faithful samples from this distribution. Given a vector $\mathbf{x} \in D$, we use $\mathbf{x}_S$ to represent a vector made of the components of $\mathbf{x}$ indexed by the set $S \subseteq \{1, \dots, d\}$.

Definition 2.1. *Given a dataset $D \subset [0,1]^d$, we define the corresponding* partial *dataset D_p as*

$$D_p = \bigcup_{\mathbf{x}\in D} \bigcup_{S\subset[d]} \mathbf{x}_S.$$

Every vector $\mathbf{x} \in D \subset [0,1]^d$ can be uniquely represented as a set $\{(1, x_1), (2, x_2), \dots, (d, x_d)\}$. A partial vector $\mathbf{z} \in D_p$ can then be a subset of $\mathbf{x}$. Thus, $\mathbf{z} \subseteq \mathbf{x}$ indicates that $\mathbf{x}$ contains the feature $\mathbf{z}$. The set D_p might be further restricted to include only connected sets (for image or text data) or only sets of a certain size as in our numerical investigation.

In our theoretical investigation, we restrict ourselves to two classes and assume the existence of a unique class for every data point. These are restrictions that we hope to relax in further research.

Definition 2.2 (Two-class Data Space)**.** *We consider the tuple $\mathfrak{D} = (D, \mathcal{D}, c)$ a* two-class data space *consisting of the dataset $D \subseteq [0,1]^d$, a probability distribution $\mathcal{D}$ along with the ground truth class map $c : D \to \{-1, 1\}$. The* class imbalance *B of a two-class data space is* $\max_{l\in\{-1,1\}} \mathbb{P}_{\mathbf{x}\sim\mathcal{D}}[c(\mathbf{x}) = l]/\mathbb{P}_{\mathbf{x}\sim\mathcal{D}}[c(\mathbf{x}) = -l]$.

We will oftentimes make use of restrictions of the set D and measure $\mathcal{D}$ to a certain class, e.g., $D_l = \{\mathbf{x} \in D \,|\, c(\mathbf{x}) = l\}$ and $\mathcal{D}_l = \mathcal{D}|_{D_l}$.

Definition 2.3 (Feature Selector)**.** *For a given dataset D, we define a* feature selector *as a map $M : D \to D_p$ such that for all $\mathbf{x} \in D$ we have $M(\mathbf{x}) \subseteq \mathbf{x}$. This means that for every data point $\mathbf{x} \in D$ the feature selector M chooses a feature that is present in $\mathbf{x}$. We call $\mathcal{M}(D)$ the space of all feature selectors for a dataset D.*

Definition 2.4 (Feature Classifier)**.** *We define a* feature classifier *for a dataset D as a function $A : D_p \to \{-1, 0, 1\}$. Here, 0 corresponds to the situation where the classifier is unable to identify a correct class. We call the space of all feature classifiers $\mathcal{A}$.*

2.1 Mutual Information, Entropy and Precision

We consider a feature to carry class information if it has high mutual information with the class. For a given feature $\mathbf{z} \in D_p$ and data points $\mathbf{x} \sim \mathcal{D}$ the mutual information is

$$I_{\mathbf{x}\sim\mathcal{D}}(c(\mathbf{x}); \mathbf{z} \subseteq \mathbf{x}) := H_{\mathbf{x}\sim\mathcal{D}}(c(\mathbf{x})) - H_{\mathbf{x}\sim\mathcal{D}}(c(\mathbf{x}) \,|\, \mathbf{z} \subseteq \mathbf{x}).$$

When the conditional entropy $H_{\mathbf{x}\sim\mathcal{D}}(c(\mathbf{x}) \,|\, \mathbf{z} \subseteq \mathbf{x})$ goes to zero, the mutual information becomes maximal and reaches the pure class entropy $H_{\mathbf{x}\sim\mathcal{D}}(c(\mathbf{x}))$ which measures how uncertain we are about the class a priori. A closely related concept is *precision*. Given another data point $\mathbf{y}$ with feature $\mathbf{z}$, precision is defined as $\mathrm{Pr}(\mathbf{z}; \mathbf{y}) := \mathbb{P}_{\mathbf{x}\sim\mathcal{D}}[c(\mathbf{x}) = c(\mathbf{y}) \,|\, \mathbf{z} \subseteq \mathbf{x}]$ and was introduced in the context of interpretability by Ribeiro et al. [2018] and Narodytska et al. [2019]. We extend this definition to a feature selector.

Definition 2.5 (Average Precision)**.** *For a given two-class data space $\mathfrak{D}$ and a feature selector $M \in \mathcal{M}(D)$, we define the* average precision *of M wrt. $\mathcal{D}$ as*

$$\mathrm{Pr}_{\mathcal{D}}(M) := \mathbb{E}_{\mathbf{y}\sim\mathcal{D}}[\mathbb{P}_{\mathbf{x}\sim\mathcal{D}}[c(\mathbf{x}) = c(\mathbf{y}) \,|\, M(\mathbf{y}) \subseteq \mathbf{x}]].$$

The average precision $\mathrm{Pr}_{\mathcal{D}}(M)$ can be used to bound the *average* conditional entropy of Merlin's features, defined as

$$\begin{aligned} H_{\mathbf{x},\mathbf{y}\sim\mathcal{D}}(c(\mathbf{x}) \,|\, M(\mathbf{y}) \subseteq \mathbf{x}) :=& \\ \mathbb{E}_{\mathbf{y}\sim\mathcal{D}}[H_{\mathbf{x}\sim\mathcal{D}}(c(\mathbf{x}) \,|\, M(\mathbf{y}) \subseteq \mathbf{x})],& \quad (1) \end{aligned}$$

and accordingly the average mutual information, see Appendix B.1. Using this, we can lower-bound the mutual information as follows:

$$\begin{aligned} \mathbb{E}_{\mathbf{y}\sim\mathcal{D}}[I_{\mathbf{x}\sim\mathcal{D}}(c(\mathbf{x}); M(\mathbf{y}) \subseteq \mathbf{x})]& \\ \geq H_{\mathbf{x}\sim\mathcal{D}}(c(\mathbf{x})) - H_b(\mathrm{Pr}_{\mathcal{D}}(M)).& \quad (2) \end{aligned}$$

When the precision goes to 1, the binary entropy $H_b(p) = -p\log(p) - (1-p)\log(1-p)$ goes to 0 and the mutual information becomes maximal. Our results are easier to state in terms of $\mathrm{Pr}_{\mathcal{D}}(M)$, because of the infinite slope of the binary entropy.

We can connect $\mathrm{Pr}_{\mathcal{D}}(M)$ back to the precision of any feature selected by M in the following way.

Lemma 2.6. *Given $\mathfrak{D} = (D, \mathcal{D}, c)$, $M \in \mathcal{M}(D)$ and $\delta \in [0, 1]$. Let $\mathbf{x}, \mathbf{y} \sim \mathcal{D}$, then with probability $1 - \delta^{-1}(1 - \mathrm{Pr}_{\mathcal{D}}(M))$, $M(\mathbf{y})$ is a feature s.t.*

$$\mathbb{P}_{\mathbf{x}\sim\mathcal{D}}[c(\mathbf{x}) = c(\mathbf{y}) \,|\, M(\mathbf{y}) \subseteq \mathbf{x}] \geq 1 - \delta.$$

The proof follows directly from Markov's inequality, see Appendix B.1. We will now introduce a new framework that will allow us to prove bounds on $\mathrm{Pr}_{\mathcal{D}}(M)$ and thus assure feature quality. For I and H, we will leave the dependence on the distribution implicit when it is clear from context.

2.2 Merlin-Arthur Classification

For a feature classifier A (Arthur) and two feature selectors M (Merlin) and $\widehat{M}$ (Morgana) we define

$$E_{M,\widehat{M},A} := \left\{ x \in D \;\middle|\; \begin{array}{l} A(M(\mathbf{x})) \neq c(\mathbf{x}) \vee \\ A\Big(\widehat{M}(\mathbf{x})\Big) = -c(\mathbf{x}) \end{array} \right\} \quad (3)$$

as the set of data points for which Merlin fails to convince Arthur of the correct class or Morgana is able to trick him into returning the wrong class, in short, the set of points where Arthur fails. We can now state the following theorem connecting the competitive game between Arthur, Merlin and Morgana to the class conditional entropy.

Theorem 2.7. *[Min-Max] Let $M \in \mathcal{M}(D)$ be a feature selector and let*

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

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

$$\mathrm{Pr}_{\mathcal{D}'}(M) = 1, \quad \textit{thus} \quad H_{\mathbf{x},\mathbf{y}\sim\mathcal{D}'}(c(\mathbf{y}) \mid M(\mathbf{y}) \subseteq \mathbf{x}) = 0.$$

The proof is in Appendix B. This theorem states that if Merlin's strategy allows Arthur to classify almost perfectly, i.e., small ϵ_M, then there exists a set that covers almost the entire original dataset and on which the class entropy conditioned on the selected features is zero. Note that these guarantees are for the set D' and not the original set D. A bound for the set D, such as $\mathrm{Pr}_{\mathcal{D}}(M) \geq 1 - \epsilon_M$, is complicated by a factor we call *asymmetric feature correlation (AFC)*, and which we explain in Section 2.3.

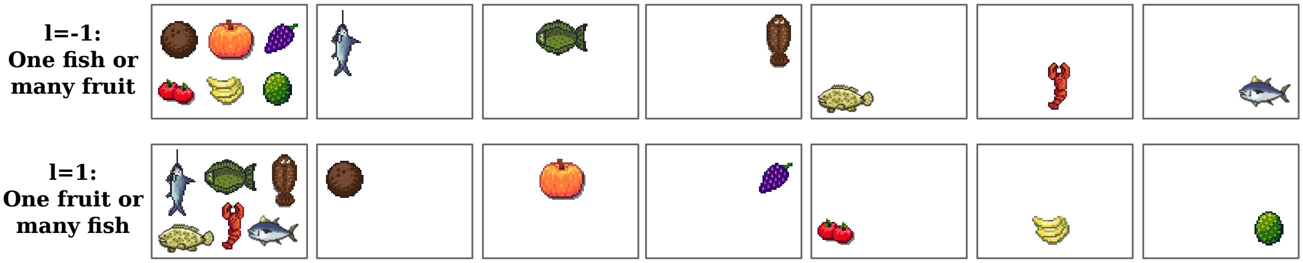


Figure 4: Example of a dataset with an AFC $\kappa = 6$. The "fruit" features are concentrated in one image for class $l = -1$ but spread out over six images for $l = 1$ (vice versa for the "fish" features). Each individual feature is not indicative of the class as it appears exactly once in each class. Nevertheless, Arthur and Merlin can exchange "fruits" to indicate "$l = 1$" and "fish" for "$l = -1$". The images where this strategy fails or can be exploited by Morgana are the two images on the left. Applying Theorem 2.7, we get $\epsilon_M = \frac{1}{7}$ and the set D' corresponds to all images with a single feature. Restricted to D', the features determine the class completely.

This bound is tight, and we provide an example of a dataset and Merlin that achieve it in Figure 4. The example shows a rather unintuitive image dataset with classes "One fish or many fruit" and "One fruit or many fish". Merlin selects a fish feature for the first and a fruit feature for the second class. The best strategy for Arthur is then to accept these features as proof for the respective class. The only images where this strategy fails are the two images with many fish or fruit, leading to a small ϵ_M no matter what Morgana does. Note, however, that each individual feature appears once in each class, which means the precision is 0.5 and the conditional entropy is 1. On the other hand, when we restrict the dataset to all images with only a single fruit or fish as D', then covers almost the whole dataset and restricted to D' the features determine the class completely. This illustrates why the restriction to D' is necessary: It allows us to connect the informativeness of a set of features (e.g., each fish feature) to the informativeness of each single feature.

2.3 Asymmetric Feature Correlation

AFC describes a possible quirk of datasets, where a set of features is strongly concentrated in a few data points in one class and spread out over almost all data points in another. We give an illustrative example in Figure 4. If a data space $\mathfrak{D}$ has a large AFC κ, Merlin can use features that individually appear equally in both classes (low precision) to indicate the class where they are spread over almost all points. Morgana can only fool Arthur in the other class where these features are highly concentrated, thus only in a few data points. This ensures a small ϵ_M even with uninformative features.

For a given set of features $F \subset D_p$, we define the set

$$F^* := \{\mathbf{x} \in D \mid \exists\, \mathbf{z} \in F : \mathbf{z} \subseteq \mathbf{x}\},$$

i.e., all data points that contain a feature from F.

Definition 2.8 (Asymmetric feature correlation)**.** *Let $(D, \mathcal{D}, c)$ be a two-class data space, then the asymmetric feature correlation κ is defined as*

$$\kappa = \max_{l\in\{-1,1\}} \max_{F\subset D_p} \mathbb{E}_{\mathbf{y}\sim\mathcal{D}_l|_{F^*}} \left[\max_{\substack{\mathbf{z}\in F\\ s.t.\ \mathbf{z}\subseteq\mathbf{y}}} \kappa_l(\mathbf{z}, F) \right]$$

with

$$\kappa_l(\mathbf{z}, F) = \frac{\mathbb{P}_{\mathbf{x}\sim\mathcal{D}_{-l}}[\mathbf{z} \subseteq \mathbf{x} \mid \mathbf{x} \in F^*]}{\mathbb{P}_{\mathbf{x}\sim\mathcal{D}_l}[\mathbf{z} \subseteq \mathbf{x} \mid \mathbf{x} \in F^*]}.$$

We derive this expression in more detail in Appendix B.3, but give an intuition here. The probability $\mathbb{P}_{\mathbf{x}\sim\mathcal{D}_l}[\mathbf{z} \subseteq \mathbf{x} \mid \mathbf{x} \in F^*]$ for $\mathbf{z} \in F$ is a measure of how correlated the features are. If all features appear in the same data points this quantity takes a maximal value of 1 for each $\mathbf{z}$. If no features share the same data point the value is minimally $\frac{1}{|F|}$ for the average $\mathbf{z}$. The $\kappa_l(\mathbf{z}, F)$ thus measures the difference in correlation between the two classes. In the example in Figure 4 the worst-case F for $l = -1$ correspond to the "fish" features and $\kappa_l(\mathbf{z}, F) = 6$ for each feature. To take an expectation over the features $\mathbf{z}$ requires a distribution, so we take the distribution of data points that have a feature from F, i.e., $\mathbf{y} \sim \mathcal{D}_l|_{F^*}$, and select the worst-case feature from each data point. Then we maximise over class and the possible feature sets F. Since in Figure 4, the "fish" and "fruit" features are the worst case for each class respectively, we arrive at an AFC of 6.

Though it is difficult to calculate the AFC for complex datasets, we show that it can be bounded above by the maximum number of features per data point in D.

Lemma 2.9. *Let $\mathfrak{D}$ be a two-class data space with AFC of κ. Let $K = \max_{\mathbf{x}\in D} |\{\mathbf{z} \in D_p \mid \mathbf{z} \subseteq \mathbf{x}\}|$ be the maximum number of features per data point. Then $\kappa \leq K$.*

We prove this in Appendix B. K depends on the type of features one considers, e.g., for image data a rectangular cutout of given size, $K \sim d$, when any subset of pixels is allowed, then $K \sim 2^d$. See also Appendix B for an example dataset with an exponentially large AFC.

2.4 Realistic Algorithms and Relative Success Rate

In Theorem 2.7, we make use of a perfect Morgana. For complex classifiers this implies exhaustive search, which is indeed possible for low-dimensional data often used in recruitment and criminal justice, where interpretability is crucial. Consider the UCI Census Income dataset Dua and Graff [2017] with 14 dimensions. When restricting features to a maximal size of seven, the search space is at most $\binom{14}{7} = 3432$, well within range for exhaustive search. Contrary to this, modelling the UCI data distribution explicitly is still an involved task, and when done incorrectly, leads to incorrect explanations Frye et al. [2020].

However, we also aim to apply our setup to high-dimensional datasets, where exhaustive search is not possible. It turns out we can relax the requirement for Morgana to play optimally in two important ways: (i) She only has to find the features that can also be found by Merlin (ii) She only has to do so with a success rate comparable to Merlin.

Definition 2.10 (Relative Success Rate)**.** *Let $\mathfrak{D} = (D, \mathcal{D}, c)$ be a two-class data space. Let $A \in \mathcal{A}$ and $M, \widehat{M} \in \mathcal{M}(D)$. Then the relative success rate α of $\widehat{M}$ with respect to A, M and $\mathfrak{D}$ is defined as*

$$\alpha := \min_{l \in \{-1,1\}} \frac{\mathbb{P}_{\mathbf{x}\sim\mathcal{D}_{-l}}\left[A(\widehat{M}(\mathbf{x})) = l \,|\, \mathbf{x} \in F_l^*\right]}{\mathbb{P}_{\mathbf{x}\sim\mathcal{D}_l}[A(M(\mathbf{x})) = l \,|\, \mathbf{x} \in F_l^*]},$$

where $F_l^* := \{\mathbf{x} \in D \,|\, \exists \mathbf{z} \subseteq \mathbf{x} : \; \mathbf{z} \in M(D_l), A(\mathbf{z}) = l\}$.

The set F_l is the set of all features that Merlin uses in class l to successfully convince Arthur, and F_l^* is the set of all data points containing such a feature. Thus, we only evaluate Morgana's performance on data points where, in principle, she can identify one of Merlin's features. The question is then how the context of the other features makes this computationally easier or harder. We discuss this idea in more depth in Appendix B and give a worst-case example in Figure 13. We argue that realistically, we can assume a large α when using an algorithm for Morgana that is at least as powerful as the one for Merlin. Together with the AFC, this allows us to state the following theorem.

Theorem 2.11. *Let $\mathfrak{D} = (D, \mathcal{D}, c)$ be a two-class data space with AFC of κ and class imbalance B. Let $A \in \mathcal{A}$, and $M, \widehat{M} \in \mathcal{M}(D)$ such that $\widehat{M}$ has a relative success rate of α with respect to A, M and $\mathfrak{D}$. Define*

1. *Completeness:*

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

2. *Soundness*

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

Then it follows that

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

The proof is provided in Appendix B. This bound is asymptotically tight. Consider again the example in Figure 4, but generalised to n fish and fruit features instead of six. Then we have $\mathrm{Pr}_{\mathcal{D}}(M) = \frac{1}{2}$, $\epsilon_c = \frac{1}{n+1}$, $\epsilon_s = \frac{1}{n+1}$, $\kappa = n$, $B = 1$ and $\alpha = 1$, since both Merlin and Morgana will always succeed in finding a feature to convince Arthur if it exists in the data point. Then we have

$$\frac{1}{2} \geq 1 - \frac{1}{n+1} - \frac{n\frac{1}{n+1}}{1 - \frac{1}{n+1} + n\frac{1}{n+1}} = \frac{1}{2} - \frac{1}{n+1},$$

which approaches equality as the number of features grows larger.

The core assumption we make when comparing our lower bound with the measured average precision in Section 3 is the following:

Assumption 2.12. *The AFC κ of $\mathfrak{D}$ and the relative success rate α of $\widehat{M}$ w.r.t. A, M, $\mathfrak{D}$ are $\mathcal{O}(1)$.*

Currently, we cannot confirm whether a dataset exhibits a small AFC. However, we conjecture that, even in cases where a dataset may include a feature set that realises large AFC, identifying such a set poses a computationally challenging task for Merlin. We leave this issue open for further investigation.

Finitely Sampled and Biased Dataset We usually have access to only finitely many samples of a dataset. Additionally, the observed samples can be biased as compared to the true distribution. We prove bounds for both cases in Appendix B.5. We show that any exchanged feature is either informative, or it is incorrectly represented in the dataset—thus highlighting the bias!

In conclusion, the theoretical results presented show that (i) For optimal feature selectors and classifiers, we can guarantee highly informative features without the need to model the data distribution, see Theorem 2.7. (ii) For suboptimal agents we can still assure informative features as long as the success probability of Morgana is comparable to the one of Merlin, see Theorem 2.11. (iii) We can certify feature quality with measurable quantities soundness and completeness.

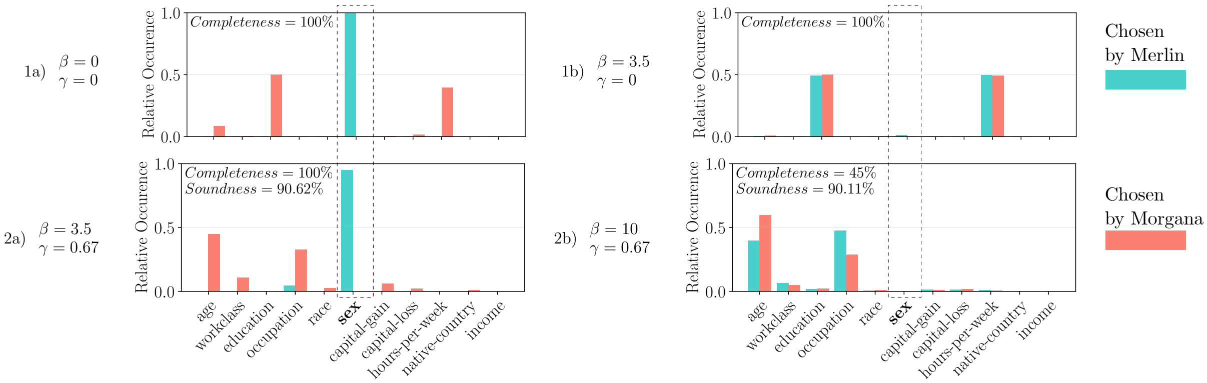


Figure 5: The objective is hiring only men while hiding "sex" as the explanation. 1. No soundness is required ($\gamma = 0$). a) Merlin has no punishment for showing "sex" ($\beta = 0$). He sends Arthur the "sex" feature and they discriminate successfully (high completeness). b) Merlin is incentivised not to use "sex" ($\beta > 0$). He successfully communicates the "sex" to Arthur via different features, here "hours per week" and "education". Morgana can exploit this strategy with the same features switched. 2. High soundness is now required ($\gamma = 0.67$). Merlin either a) shows the "sex" feature despite the punishment ($\beta = 3.5$) and achieves high completeness, or b) hides the "sex" feature ($\beta = 10$) and reduces completeness to below 50%, ceasing the discrimination.

3 Numerical Implementation

Let us describe how to train the agents Arthur, Merlin, and Morgana in a general n-class interactive learning setting for image data of dimension d, where $n, d \in \mathbb{N}$. We explain in Appendix A.3 why we chose a multi-class neural network for Arthur and compare with the approaches of Chang et al. [2019] and Anil et al. [2021]. The training process for tabular data is equivalent, a detailed overview is provided in the appendix.[1]

Arthur is modelled by a feed-forward neural network. He returns a probability distribution over his possible answers, so let $A : [0,1]^d \to [0,1]^{(n+1)}$, corresponding to the probabilities of stating a class or "Don't know". The provers select a set S of at most k pixels from the image via a *mask* $\mathbf{s} \in B_k^d$, where B_k^d is the space of k-sparse binary vectors of dimension d. A masked image $\mathbf{s} \cdot \mathbf{x}$ has all its pixels outside of S set to a baseline or a random value. We define the Merlin-loss L_M as the cross-entropy loss with regard to the correct class, whereas the Morgana-loss $L_{\widehat{M}}$ considers the total probability of either answering the correct class or the "I don't know" option, so

$$L_M(A, \mathbf{x}, \mathbf{s}) = -\log\big(A(\mathbf{s} \cdot \mathbf{x})_{c(\mathbf{x})}\big) \quad \text{and}$$
$$L_{\widehat{M}}(A, \mathbf{x}, \mathbf{s}) = -\log\big(A(\mathbf{s} \cdot \mathbf{x}))_0 + A(\mathbf{s} \cdot \mathbf{x})_{c(\mathbf{x})}\big).$$

Arthur's total loss is then $L = \mathbb{E}_{\mathbf{x} \sim \mathcal{D}}[L(\mathbf{x})]$, where

$$L(\mathbf{x}) = (1-\gamma) L_M(A, \mathbf{x}, M(\mathbf{x})) + \gamma L_{\widehat{M}}(A, \mathbf{x}, \widehat{M}(\mathbf{x})),$$

and $\gamma \in [0,1]$ is a tunable parameter. In our experiments, we choose $\gamma > 0.5$ since we always want to ensure good soundness. Note that Merlin wants to minimise L_M, whereas Morgana aims to maximise $L_{\widehat{M}}$. In an ideal world, they would solve

$$\begin{aligned} M(\mathbf{x}) &= \operatorname*{argmin}_{\mathbf{s} \in B_k^d} L_M(A, \mathbf{x}, \mathbf{s}) \quad \text{and} \\ \widehat{M}(\mathbf{x}) &= \operatorname*{argmax}_{\mathbf{s} \in B_k^d} L_{\widehat{M}}(A, \mathbf{x}, \mathbf{s}). \end{aligned} \tag{4}$$

The above solutions can be obtained either by solving the optimisation problem (Frank-Wolfe solver [Macdonald et al., 2022]) or by using U-Nets to predict the solutions. We describe the training algorithm in Algorithm 1: For N epochs we iterate over the dataset and alternately train Arthur on masked images and on the original, unmasked images. The update steps for Merlin and Morgana (steps 8 and 9) only apply when the feature selectors are realised by U-Nets.

Algorithm 1 Merlin-ArthurTraining

```
 1: Input: dataset: $D_{\text{train}}$, Epochs: $N$, $\gamma$
 2: Output: Classifier Network (A), Optional: Masking Networks Merlin ($M$) and Morgana ($\widehat{M}$)
 3: for $i \in [N]$ do
 4:     for $\mathbf{x}_j, l_j \in D_{\text{train}}$ do
 5:         $\mathbf{s}_M \leftarrow M(\mathbf{x}_j, l_j), \mathbf{s}_{\widehat{M}} \leftarrow \widehat{M}(\mathbf{x}_j, l_j)$
 6:         $L_A(\mathbf{x}_j, l_j) = (1-\gamma) L_M(A(\mathbf{s}_M \cdot \mathbf{x}_j), l_j) + \gamma L_{\widehat{M}}(A(\mathbf{s}_{\widehat{M}} \cdot \mathbf{x}_j), l_j)$
 7:         $\theta_A \leftarrow \theta_A - \alpha \nabla_\theta L_A(\mathbf{x}_j, l_j)$
 8:         $\theta_M \leftarrow \theta_M - \alpha \nabla_\theta L_M(A(M(\mathbf{x}_j) \cdot \mathbf{x}_j), l_j)$
 9:         $\theta_{\widehat{M}} \leftarrow \theta_{\widehat{M}} - \alpha \nabla_\theta L_{\widehat{M}}(A(\widehat{M}(\mathbf{x}_j) \cdot \mathbf{x}_j), l_j)$
10:     end for
11:     for $\mathbf{x}_j, l_j \in D_{\text{train}}$ do
12:         $\theta_A \leftarrow \theta_A - \alpha \nabla_\theta L(A(\mathbf{x}_j), l_j))$
13:     end for
14: end for
```

[1] The code is available at `https://github.com/ZIB-IOL/merlin-arthur-classifiers`.

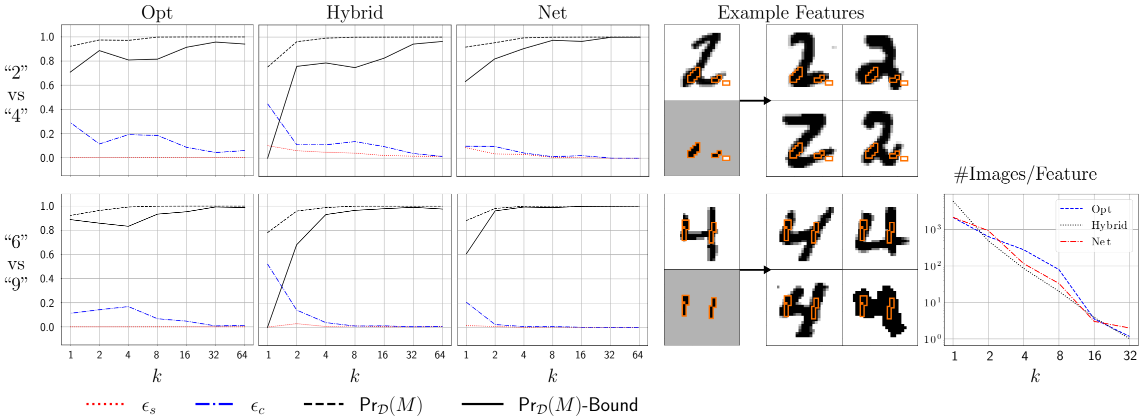


Figure 6: *Left:* For four different setups of Merlin and Morgana, we compare the lower bound on $\mathrm{Pr}_{\mathcal{D}}(M)$ with the experimental results on the MNIST dataset. The top row is for the labels {"2", "4"}, and the bottom row for {"6", "9"}. The bound is tight for large masks, but loosens sharply for very small mask sizes k. *Middle:* Examples of the features selected by Merlin for two images. For the "4" feature, there are 13 images in MNIST that share it, all of them of class "4" (we show four here). For the "2" there are 16 images, all of them in "2". *Right:* The average number of images found for a feature selected by Merlin. These images were used to estimate $\mathrm{Pr}_{\mathcal{D}}(M)$.

3.1 Preventing Manipulation

XAI algorithms without guarantees cannot exclude the possibility for manipulation. Indeed, arbitrarily changing the interpretation by slightly modifying the classifier has been demonstrated for many XAI approaches. Slack et al. fool LIME and ShAP by making use of the fact that these methods sample off-manifold evaluations of the classifier. We are robust against this approach, since the Merlin-Arthur classifier only takes on-manifold inputs, see Appendix C.1.3 for further details. Dimanov et al., Heo et al. and Anders et al. optimise manipulated classifier networks to give the desired explanations by penalising any deviation. They introduce a penalty term in the network objective that measures the distance between the produced and the desired relevance scores. The equivalent for our setup is to put a penalty on Merlin to hide the true (potentially biased) explanations in the exchanged features. Our theory predicts this scheme to be unsuccessful, and we confirm numerically that either: (i) The bias becomes visible, (ii) Morgana can exploit the setup, i.e., soundness is low or (iii) the completeness, i.e., prediction accuracy, drops.

We demonstrate this on a modified UCI Census Income dataset[2]. We train a classifier to discriminate by "sex" — one of the remaining 12 features. We change the Merlin-loss to

$$L'_M(A, \mathbf{x}, \mathbf{s}) := L_M(A, \mathbf{x}, \mathbf{s}) + \beta \|\mathbf{s}_{\text{sex}}\|,$$

where $\beta > 0$, to train Merlin to hide the sensitive "sex" feature. We then train the Merlin-Arthur classifier to accept only men and set $k = 1$ to only send one feature. We compare a setup where soundness is ignored to one where we choose γ such that a soundness of at least 90% is achieved.

The results are illustrated in Figure 5. When soundness is ignored ($\gamma = 0$), setting $\beta = 3.5$ causes Merlin to hide the "sex" feature and communicate "sex" via selecting "education" for men, "working hours" for women. When soundness is required ($\gamma = 0.67$), Merlin starts to reveal the "sex" feature again. Increasing the penalty to $\beta = 10$, causes him to hide "sex", but now the prediction accuracy reduces to 0.45, i.e., close to chance and thus ceasing the discrimination. When high soundness is required, Merlin is unable to both discriminate and hide his bias.

In Appendix C.1.3, we apply this approach to different post-hoc XAI methods and show that they are indeed susceptible to manipulation.

3.2 Evaluation of Theoretical Bounds

Low-dimensional datasets serve to evaluate our mutual information bounds. For small features there are multiple data points that contain them and the ground truth information of the feature can be estimated. We evaluate Assumption 2.12 on the MNIST dataset restricted to two classes for three setups of feature selectors, one with Frank-Wolfe optimisers (**Opt**), one where Merlin is a U-Net and Morgana an optimiser (**Hybrid**), and with U-Nets for both (**Net**). We want to stress that this is not a comparison to other XAI methods, which do not generally make predic-

[2] For this experiment, we have removed the "relationship" and "marital status" features, since they strongly correlate with sex. See Appendix C.1 for experiments that include them.

tions about the precision of the highlighted features. Standard classifiers might be easier to train or achieve higher accuracy compared to a Merlin-Arthur classifier. But they are not interpretable, as there are no post-hoc methods robust to manipulation.

In Figure 6, the lower bound is tight for larger masks, but drops off when k is small. One reason is that for small masks, Arthur tends to give up on one of the classes, while keeping the completeness in the other class high. Regularising Arthur to maintain equal completeness is a potential solution. When Merlin and Morgana are realised by the same method (both optimisers or NNs), the bound is the tightest. In our hybrid approach, the bound is pessimistic since Merlin is at a disadvantage. He needs to learn on the training set to select good features, whereas Morgana can optimise directly on the test set. In Appendix C Figure 17, we show error bars sampled over 10 training runs. Our lower bound is always below the empirical estimate, which is evidence that Assumption 2.12 is correct. This must be evaluated more extensively on different datasets.

4 Discussion and Limitations

We can draw a connection between soundness and Adversarial Robustness [Goodfellow et al., 2015]. Consider the generation of adversarial examples :

$$\boldsymbol{\delta}^* = \operatorname*{argmin}_{\|\boldsymbol{\delta}\| \leq \epsilon} L(\mathbf{x} + \boldsymbol{\delta}).$$

The intuition is that minuscule changes to the input, imperceptible to humans, should not change the classifier decision. Likewise, the intuition behind soundness, i.e., robustness with respect to Morgana, is that hiding parts of an object should not convince the classifier of a different class. At most, one could hide the whole object, which is reflected in the “Don't know!” option. In this sense, soundness should be expected of classifiers that generalise to partially hidden objects.

AFC seems to be more generally relevant to interactive interpretability, even in different setups than ours. In Yu et al. [2019], a prover sends part of an image to a cooperative classifier and the rest to an adversarial classifier. The goal is to allow correct classification for the cooperator and prevent it for the adversary. However, as in our example in Figure 4, the prover can use completely uninformative features (“fish” and “fruit”) and the adversary is unable to exploit this except for a small number of image, inversely proportional to the AFC. This means the AFC would need to be part of an information bound, assuming it is formulated in terms of the accuracy of the cooperator and adversary on the whole dataset. For details, see Appendix A.4.2.

High completeness and soundness can be mandated for commercial classifiers, e.g., in the context of hiring decisions with past decisions by the Merlin-Arthur classifier as ground truth. An auditor uses their own Morgana to verify sufficient soundness. If Arthur is sound, the features selected by Merlin are verifiably the basis of the hiring decisions and can be inspected for protected attributes, e.g., race, sex or attributes that strongly correlate with them [Mehrabi et al., 2021]. This hinges on the fact, as explained in in Section 2.4 in terms of the relative success rate, whether the Morgana used by the auditor is computationally as powerful as the Merlin used by hiring department.

However, simply identifying features with high mutual information does not necessarily point to causal mechanisms, since they can include spurious correlations. While the adversary prevents such correlations in the masks, they might still be present in the original data. In the UCI dataset, the removed features “marital status” and “relationship” are correlated with sex and thus can be used by Merlin to communicate “sex” to Arthur when included, see Appendix C.1. It is up to society to determine whether the exchanged features constitute discrimination. This is a problem shared generally by interpretability tools, however, there has been progress to adapt interactive classification to find causal features Chang et al. [2020].

In future work, we aim to move beyond the restriction to the deterministic two-class case. We discuss the training stability of the three-player game and numerical challenges in Appendix C.2.2.

5 Conclusion

We extend the framework of adversarial interactive classifiers to provide quantitative mutual information bounds on the exchanged features in terms of the measurable criteria completeness and soundness. We also move beyond the common assumptions of optimally playing agents and of feature independence. Instead, we consider the relative strength of the provers and introduce Asymmetric Feature Correlation, which captures the relevant aspect of the feature dependence. Finally, we evaluate our results on the UCI Census Income and MNIST datasets. Our experiments show that the Merlin-Arthur classifier can prevent manipulation that is successful for other XAI methods, and that our theory matches well with our numerics.

Acknowledgement Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany´s Excellence Strategy – The Berlin Mathematics Research Center MATH+ (EXC-2046/1, project ID: 390685689).

References

K. Aas, M. Jullum, and A. Løland. Explaining individual predictions when features are dependent: More accurate approximations to Shapley values. *Artificial Intelligence*, 298:103502, 2021.

C. Agarwal and A. Nguyen. Explaining image classifiers by removing input features using generative models. In *Computer Vision – ACCV 2020*, pages 101–118. Springer International Publishing, 2021.

D. Alvarez-Melis and T. S. Jaakkola. Towards robust interpretability with self-explaining neural networks. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, page 7786–7795. Curran Associates Inc., 2018.

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.

C. Anil, G. Zhang, Y. Wu, and R. Grosse. Learning to give checkable answers with prover-verifier games. *arXiv preprint arXiv:2108.12099*, 2021.

S. Arora and B. Barak. *Computational complexity: a modern approach.* Cambridge University Press, 2009.

J. Bastings, W. Aziz, and I. Titov. Interpretable neural predictions with differentiable binary variables. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2963–2977. Association for Computational Linguistics, 2019.

G. Blanc, J. Lange, and L.-Y. Tan. Provably efficient, succinct, and precise explanations. *Advances in Neural Information Processing Systems*, 34:6129–6141, 2021.

J. S. Bridle. Training stochastic model recognition algorithms as networks can lead to maximum mutual information estimation of parameters. In *Proceedings of the 2nd International Conference on Neural Information Processing Systems*, NIPS'89, page 211–217. MIT Press, 1989.

C.-H. Chang, E. Creager, A. Goldenberg, and D. Duvenaud. Explaining image classifiers by counterfactual generation. *arXiv preprint arXiv:1807.08024*, 2018.

S. Chang, Y. Zhang, M. Yu, and T. Jaakkola. A game theoretic approach to class-wise selective rationalization. *Advances in neural information processing systems*, 32, 2019.

S. Chang, Y. Zhang, M. Yu, and T. Jaakkola. Invariant rationalization. In *International Conference on Machine Learning*, pages 1448–1458. PMLR, 2020.

A. Chattopadhyay, S. Slocum, B. D. Haeffele, R. Vidal, and D. Geman. Interpretable by design: Learning predictors by composing interpretable queries. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(6):7430–7443, 2022.

J. Chen, L. Song, M. Wainwright, and M. Jordan. Learning to explain: An information-theoretic perspective on model interpretation. In *International Conference on Machine Learning*, pages 883–892. PMLR, 2018.

P. Dabkowski and Y. Gal. Real time image saliency for black box classifiers. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6970–6979, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

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*, volume 2560 of *CEUR Workshop Proceedings*, pages 63–73. CEUR-WS.org, 2020.

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. *Advances in neural information processing systems*, 32: 13589–13600, 2019.

D. Dua and C. Graff. UCI machine learning repository, 2017. URL `http://archive.ics.uci.edu/ml`.

R. M. Fano. Transmission of information: A statistical theory of communications. *American Journal of Physics*, 29(11):793–794, 1961.

R. C. Fong and A. Vedaldi. Interpretable explanations of black boxes by meaningful perturbation. In *Proceedings of the IEEE international conference on computer vision*, pages 3429–3437, 2017.

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.

S. Goldwasser, G. N. Rothblum, J. Shafer, and A. Yehudayoff. Interactive proofs for verifying machine learning. In *12th Innovations in Theoretical Computer Science Conference (ITCS 2021)*, volume 185 of *LIPIcs*, pages 1–19. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2021.

I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. In *3rd International Conference on Learning Representations, ICLR 2015*, 2015.

B. Goodman and S. Flaxman. European Union regulations on algorithmic decision-making and a "right to explanation". *AI magazine*, 38(3):50–57, 2017.

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.

A. Ignatiev, N. Narodytska, and J. Marques-Silva. Abduction-based explanations for machine learning models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 1511–1519, 2019.

G. Irving, P. Christiano, and D. Amodei. AI Safety via debate. *arXiv preprint arXiv:1805.00899*, 2018.

Y. Izza, A. Ignatiev, and J. Marques-Silva. On explaining decision trees. *arXiv preprint arXiv:2010.11034*, 2020.

M. Jaggi. Revisiting Frank-Wolfe: Projection-free sparse convex optimization. In *International Conference on Machine Learning*, pages 427–435. PMLR, 2013.

J. Kleinberg and E. Tardos. *Algorithm design*. Pearson Education India, 2006.

N. Kokhlikyan, V. Miglani, M. Martin, E. Wang, B. Alsallakh, J. Reynolds, A. Melnikov, N. Kliushkina, C. Araya, S. Yan, and O. Reblitz-Richardson. Captum: A unified and generic model interpretability library for PyTorch, 2020.

S. Lapuschkin, S. Wäldchen, A. Binder, G. Montavon, W. Samek, and K.-R. Müller. Unmasking clever hans predictors and assessing what machines really learn. *Nature communications*, 10(1):1–8, 2019.

T. Lei, R. Barzilay, and T. Jaakkola. Rationalizing neural predictions. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 107–117. Association for Computational Linguistics, 2016.

S. Liu, B. Kailkhura, D. Loveland, and Y. Han. Generative counterfactual introspection for explainable deep learning. In *2019 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pages 1–5. IEEE, 2019.

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

J. Macdonald and S. Wäldchen. A complete characterisation of ReLu-Invariant Distributions. In *International Conference on Artificial Intelligence and Statistics*, pages 1457–1484. PMLR, 2022.

J. Macdonald, S. Wäldchen, S. Hauch, and G. Kutyniok. A rate-distortion framework for explaining neural network decisions. *arXiv preprint arXiv:1905.11092*, 2019.

J. Macdonald, S. Wäldchen, S. Hauch, and G. Kutyniok. Explaining neural network decisions is hard. In *XXAI Workshop, 37th ICML*, 2020.

J. Macdonald, M. Besancon, and S. Pokutta. Interpretable neural networks with Frank-Wolfe: Sparse relevance maps and relevance orderings. In *International Conference on Machine Learning*, pages 14699–14716. PMLR, 2022.

J. Marques-Silva, T. Gerspacher, M. C. Cooper, A. Ignatiev, and N. Narodytska. Explanations for monotonic classifiers. In *International Conference on Machine Learning*, pages 7469–7479. PMLR, 2021.

N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan. A survey on bias and fairness in machine learning. *ACM Computing Surveys (CSUR)*, 54(6):1–35, 2021.

S. Mertes, T. Huber, K. Weitz, A. Heimerl, and E. André. This is not the texture you are looking for! introducing novel counterfactual explanations for non-experts using generative adversarial learning. *arXiv preprint arXiv:2012.11905*, 2020.

S. Mohseni, N. Zarei, and E. D. Ragan. A multidisciplinary survey and framework for design and evaluation of explainable ai systems. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 11(3-4):1–45, 2021.

N. Narodytska, A. Shrotri, K. S. Meel, A. Ignatiev, and J. Marques-Silva. Assessing heuristic machine learning explanations with model counting. In *International Conference on Theory and Applications of Satisfiability Testing*, pages 267–278. Springer, 2019.

C. Olah, A. Satyanarayan, I. Johnson, S. Carter, L. Schubert, K. Ye, and A. Mordvintsev. The building blocks of interpretability. *Distill*, 3(3):e10, 2018.

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

S. Poulis and S. Dasgupta. Learning with feature feedback: from theory to practice. In *Artificial Intelligence and Statistics*, pages 1104–1113. PMLR, 2017.

M. T. Ribeiro, S. Singh, and C. Guestrin. "Why should I trust you?" Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016.

M. T. Ribeiro, S. Singh, and C. Guestrin. Anchors: High-precision model-agnostic explanations. In *Proceedings of the AAAI conference on artificial intel-*

ligence, volume 32 of *AAAI'18*, pages 1527–1535. AAAI Press, 2018.

O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

K. Roth, A. Lucchi, S. Nowozin, and T. Hofmann. Stabilizing training of generative adversarial networks through regularization. In *Advances in neural information processing systems*, volume 30 of *NIPS'17*, page 2015–2025, 2017.

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 classifiers. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, page 5103–5111, 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, A. Hilgard, H. Lakkaraju, and S. Singh. Counterfactual explanations can be manipulated. *Advances in neural information processing systems*, 34:62–75, 2021.

S. Waeldchen, J. Macdonald, S. Hauch, and G. Kutyniok. The computational complexity of understanding binary classifier decisions. *Journal of Artificial Intelligence Research*, 70:351–387, 2021.

S. Wäldchen. Hardness of deceptive certificate selection. In L. Longo, editor, *Explainable Artificial Intelligence*, pages 415–427. Springer Nature Switzerland, 2023.

S. Wäldchen, S. Pokutta, and F. Huber. Training characteristic functions with reinforcement learning: XAI-methods play Connect Four. In *International Conference on Machine Learning*, pages 22457–22474. PMLR, 2022.

M. Wiatrak, S. V. Albrecht, and A. Nystrom. Stabilizing generative adversarial networks: A survey. *arXiv preprint arXiv:1910.00927*, 2019.

C. Yadav, M. Moshkovitz, and K. Chaudhuri. A learning-theoretic U-Net framework for certified auditing of machine learning models. *arXiv preprint arXiv:2206.04740*, 2022.

M. Yu, S. Chang, Y. Zhang, and T. Jaakkola. Rethinking cooperative rationalization: Introspective extraction and complement control. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4094–4103. Association for Computational Linguistics, 2019.

Supplementary Material: Interpretability Guarantees with Merlin-Arthur Classifiers

A Conceptual Overview

Formal interpretability faces two hurdles: a complexity barrier, as well as a modelling problem. Here, we further explain these challenges and how we overcome them. Furthermore, we also compare our architecture in detail with that proposed in Chang et al. [2019] and Anil et al. [2021] and provide an explanation for the differences in Appendix A.3. For two alternative interactive classification setups with an adversary, the debate setup [Irving et al., 2018] and the adversarial classifier setup Yu et al. [2019], Dabkowski and Gal [2017], we show that they cannot be used to derive bounds with the same generality as in our work. Instead, it would require stronger assumptions on either the classifier or the data space to exclude our counterexamples, see Appendix A.4.

A.1 Computational Complexity

Prime Implicant Explanations [Shih et al., 2018], a concept from logical abduction, can be efficiently computed for simple classifiers like decision trees [Izza et al., 2020] and monotonic functions [Marques-Silva et al., 2021]. This concept has been extended to NNs [Ignatiev et al., 2019] in the form of *probabilistic* prime implicants, which correspond to features with high precision. However, it has been shown that even approximating small implicants within any non-trivial factor is NP-hard [Waeldchen et al., 2021] for networks of two layers or more. In Blanc et al. [2021], the authors construct an algorithm that circumvents these hardness results by further relaxation of the problem. While this is a noteworthy theoretical breakthrough, the polynomial bound on the feature size grows so quickly with the dimension of the data space that the algorithm does not guarantee useful features for real-world data. For reasonably sized images, one would get guarantees only for features that cover the whole image.

We circumvent the hardness of this problem using a method that is very typical of Deep Learning: Use a heuristic and verify success afterwards! Our approach can be put alongside the regular training of classifiers, which is a theoretically hard problem as well. A heuristic like Stochastic Gradient Descent is not a priori guaranteed to produce a capable classifier. However, we can check the success of the procedure by evaluating the accuracy on a test dataset. In our case, training the Merlin-Arthur classifier is not guaranteed to converge to an equilibrium with informative features. But we can check whether this is the case via the test dataset, where soundness and completeness take the role of the accuracy.

A.2 Modelling the True Data Distribution

We introduce Merlin-Arthur classification as it provides us a way to measure the feature quality via the completeness and soundness values over a test dataset. This would not be necessary if we could directly measure the feature quality over the dataset (though it would still be faster than measuring every individual feature). The reason we need the Merlin-Arthur setup is that for general datasets the conditional entropy

$$H_{\mathbf{y}\sim\mathcal{D}}(c(\mathbf{y})\,|\,\mathbf{z}\subseteq\mathbf{y}) = H_{\mathbf{y}\sim\mathcal{D}|_{\mathbf{z}\subseteq\mathbf{y}}}(c(\mathbf{y})),$$

is difficult to measure, since we do not generally know the conditional distribution $\mathcal{D}|_{\mathbf{z}\subseteq\mathbf{y}}$. This measurement is possible for MNIST for small features since the dataset is very simple. However, for more complex data, a feature which is large enough to be indicative of the class will in all likelihood not appear more than once in the same dataset. We will now discuss some existing approaches that aim to approximate the conditional data distribution and what problems they face.

Modelling the conditional data distribution has been pursued in the context of calculating Shapley values. These are different interpretability method based on *characteristic functions* from cooperative game theory that assign

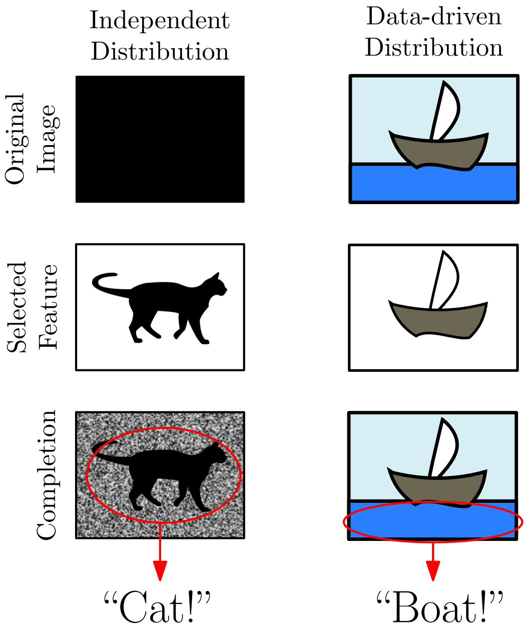


Figure 7: Different failure modes of unrepresentative distributions. *Left:* Independent, random inpainting, similar to Macdonald et al. [2019]. From a black image, the shape of a cat is selected, and the rest is filled with uniform noise. The shape of a cat is detected by a classifier. *Right:* Data-driven inpainting, similar to Agarwal and Nguyen [2021]. The image of a ship is given and the ship-feature is selected. The data driven distribution inpaints the water back into the image, since in the dataset, ships are always on water. The faulty classifier that relies on the water feature is undetected, as the ship-feature indirectly leads to the correct classification.

a value to every subset of a number of features [Shapley, 2016]. We will shortly discuss the approach proposed in Lundberg and Lee [2017], where features correspond to partial vectors supported on sets.

Let $f : [0,1]^d \to \{-1,1\}$ be a classifier function. Then we can naturally define a characteristic function $\nu_{f,\mathbf{x}} : \mathcal{P}([d]) \to [-1,1]$ as

$$\nu_{f,\mathbf{x}}(S) = \mathbb{E}_{\mathbf{y}\sim\mathcal{D}}[f(\mathbf{y}) \,|\, \mathbf{y}_S = \mathbf{x}_S] = \int f(\mathbf{y}_S, \mathbf{x}_{S^c}) d\mathbb{P}_{\mathbf{y}\sim\mathcal{D}}(\mathbf{y}_{S^c} \mid \mathbf{y}_S = \mathbf{x}_S).$$

The Shapley value for the input component x_i is then defined as

$$\text{Shapley Value}(x_i) = \frac{1}{d} \sum_{S\subseteq[d]\setminus\{i\}} \binom{d-1}{|S|}^{-1} (\nu_{f,\mathbf{x}}(S \cup \{i\}) - \nu_{f,\mathbf{x}}(S)).$$

In Macdonald et al. [2022], the characteristic function is instead used to define a feature selection method as

$$\mathbf{x}_{S^*} \quad \text{where} \quad S^* = \underset{|S|\leq k}{\text{argmin}} \ \text{dist}(\nu([d]), \nu(S)),$$

where $k \in [d]$ is a cap on the set size and dist is an appropriate distance measure.

As in our setup, the problem is that these approaches depend on how well the conditional probability $\mathbb{P}_{\mathbf{y}\sim\mathcal{D}}(\mathbf{y}_{S^c} \mid \mathbf{y}_S = \mathbf{x}_S)$ is modelled. Modelling the data distribution incorrectly makes it possible to manipulate many existing XAI-methods. This is done by changing the classifier in such a way that it gives the same value on-manifold, but arbitrary values off-manifold. To get feature-based explanations independent of the off-manifold behaviour, one needs to model the data manifold very precisely [Aas et al., 2021, Dombrowski et al., 2019]. The authors of Anders et al. [2020], Heo et al. [2019] and Dombrowski et al. [2019] demonstrate this effect for existing techniques, such as sensitivity analysis, LRP, Grad-Cam, IntegratedGradients and Guided Backprop. They are able to manipulate relevance scores at will and demonstrate how this can be used to obfuscate discrimination inside a model. LIME and SHAP can be manipulated as well [Slack et al., 2020] by using a classifier that behaves differently outside off-distribution if the wrong distribution for the explanation. For RDE [Macdonald et al., 2019] it is assumed that features are independent and normally distributed, and it was demonstrated that the off-manifold optimisation can create new features that weren't in the original image [Wäldchen et al., 2022].

We now discuss two approaches proposed to model the data distribution and why each leads to a different problem by under- or over-representing correlation in the data respectively.

Independent distribution: Which means that the conditional probability is modelled as

$$\mathbb{P}_{\mathbf{y}\sim\mathcal{D}}(\mathbf{y}_{S^c} \mid \mathbf{y}_S = \mathbf{x}_S) = \prod_{i\in S^c} p(y_i),$$

where $p(y_i)$ are suitable probability densities on the individual input components. This approach has been used in Fong and Vedaldi [2017] and Macdonald et al. [2022], where optimisers are employed to find small features that maximise the classifier score. In fact, Macdonald et al. [2022] has to make a new approximation of the data distribution in every layer and it has been shown that for neural networks one cannot do much better than applying either this or sampling Macdonald and Wäldchen [2022]. It was highlighted in Wäldchen et al. [2022] how this approach, when modelling the data distribution incorrectly, will create artificial new features that were not present in the original image. Employing an optimisation method with this distribution can result in masks that generate new features that were not present in the original image. We illustrate this problem in Figure 7. Cutting a specific shape out of a monochrome background will with high likelihood result in an image where this shape is visible. If the distribution was true, a monochrome shape would likely lead to an inpainting that is monochrome in the same colour, destroying the artificial feature. But an independent distribution under-represents these reasonable correlations.

Taking a data-determined distribution via generative model: Which means that the conditional probability is modelled as

$$\mathbb{P}_{\mathbf{y}\sim\mathcal{D}}(\mathbf{y}_{S^c} \mid \mathbf{y}_S = \mathbf{x}_S) = G(\mathbf{y}_{S^c}; \mathbf{x}_S),$$

where G is a suitable generative model. Generative models as a means to approximate the data distribution in the context of explainability have been proposed in a series of publications [Agarwal and Nguyen, 2021, Chang et al., 2018, Liu et al., 2019, Mertes et al., 2020]. This setup introduces a problem. If the network and the generator were trained on the same dataset, the biases learned by the classifier will appear might be learned by the generator as well (see Figure 7 for an illustration)! The important cases will be exactly the kind of cases that we will not be able to detect. If the generator has learned that horses and image source tags are highly correlated, it will inpaint an image source tag when a horse is present. This allows the network to classify correctly, even when the network only looks for the tag and has no idea about horses. The faulty distribution over-represents correlations that are not present in the real-world data distribution.

A.3 Design of the Three-Way Game

The basic setup for a prover-verifier game for classification was proposed by Chang et al. with a verifier, a cooperative prover and an adversarial prover for one specific class. The verifier either accepts the evidence for the class or rejects it. Both provers try to convince the verifier, the cooperative prover operates on data from the class, the adversary on data from outside the class. The authors suggest that the way to scale to multiple classes is to have three agents for every class.

In our work, we combine the agents over all classes, to have a single verifier (Arthur), cooperator (Merlin), and adversary (Morgana). The verifier rejecting all the classes in their paper corresponds to our "Don't know!" option. In our design in Section 3, we make the implicit assumption that the class of the data point is unique. Combining the verifiers gives us a numerical advantage for two reasons. First, since a lot of lower-level concepts (e.g. edges and corners for image data) are shared over classes, the lower levels of the neural network benefit by being trained on more and more diverse data. Second, we can leverage the knowledge that the class is unique by outputting a distribution over classes (and "Don't know!"). Both lines of reasoning are standard for deep learning Bridle [1989].

Anil et al. further combine Merlin and Morgana into a single prover that probabilistically produces a certificate for a random class. This has the advantage that it allows for further weight-sharing among the provers. However, the probabilistic nature of the certificate is also a disadvantage. The probability of generating the certificate for the correct class is the inverse of the total number of classes. When applied after training, one only occasionally gets a valid classification. In our case, we can always use Merlin together with Arthur to obtain the correct class together with an interpretable feature

A.4 Alternative setups

Here we discuss alternatives to the Merlin-Arthur setup. Both these alternatives present an interactive classification setup as well. However, we show that they cannot prove bounds with the same generality as we have proven.

A.4.1 Debate Model

The debate setting introduced in Irving et al. [2018] is an intriguing alternative to our proposed setup. However, we are now going to present an example data space on which, in debate mode, Arthur and Merlin can cooperate perfectly without using informative features. For this, we use the fact that in the debate setting, Arthur receives features from both Merlin and Morgana for each classification. Our example illustrates that the debate setting would need stronger requirements on either the data space or Arthur to produce results similar to ours.

Consider the following example of a data space $\mathfrak{D}^{\text{ex}}$, illustrated in Figure 8.

Example A.1. *Given $n \in \mathbb{N}, n \geq 4$, we define the data space $\mathfrak{D}^{ex} = (D, \mathcal{D}, c)$ with*

- $D = D_{-1} \cup D_1$ *where* $D_s = \bigcup_{k=1}^{n} [2k+s, 2k]$
- *for* $T \in \mathcal{P}(D) : \mathcal{D}(T) = \frac{|T|}{N}$,
- $c(\mathbf{x}) = \begin{cases} -1 & \mathbf{x} \in D_{-1} \\ 1 & \mathbf{x} \in D_1. \end{cases}$

None of the features in D_p are informative of the class and the mutual information $I(c(x); \mathbf{z} \in \mathbf{x})$ for any $\mathbf{z} \in D_p$ is zero. Nevertheless, in a debate setting, Arthur can use the following strategy after receiving a total of two features from Merlin and Morgana

$$A(\{\mathbf{z}_1, \mathbf{z}_2\}) = \begin{cases} c(\mathbf{x}^*) & \text{where } \mathbf{x}^* = \operatorname{argmax}_{\mathbf{x} \in D} \|\mathbf{x}\|_1 \text{ s.t. } \mathbf{z}_1 \subseteq \mathbf{x}, \mathbf{z}_2 \subseteq \mathbf{x}, \\ 0 & \text{if } \nexists \mathbf{x} \in D : \mathbf{z}_1 \subseteq \mathbf{x}, \mathbf{z}_2 \subseteq \mathbf{x}. \end{cases}$$

This means he returns the class of the data point with the largest 1-norm that fits the presented features. But now Merlin can use the strategy

$$M(\mathbf{x}) = \underset{\mathbf{z}}{\operatorname{argmin}} \|\mathbf{z}\|_1 \text{ s.t. } \mathbf{z} \subseteq \mathbf{x},$$

which returns the feature with the smaller 1-norm. It is easy to verify that no matter what Morgana puts forward, a feature with smaller or larger entry, nothing can convince Arthur of the wrong class. If she gives the same feature as Merlin, the data point will be correctly determined by Arthur. If she gives the other feature, the true data point is the unique one that has both features. Arthur's strategy works as long as *someone* gives him the smaller feature.

In a setting where Arthur has to evaluate every feature individually, the best strategy that Arthur and Merlin can use achieves $\epsilon_c = \epsilon_s = \frac{1}{3}$, by making use of the asymmetric feature concentration. The AFC for $\mathfrak{D}^{\text{ex}}$ is $\kappa = 2$, as can be easily verified by taking $F = \{[*, 2], [3, *], [*, 6], [7, *]\}$ in the definition of the AFC, see Definition 2.8, and observing that they cover 4 data points in class $l = -1$ and only two in class $l = 1$. But since the AFC-constant appears in the bound, the lower bound for $\operatorname{Pr}_{\mathcal{D}}(M)$ is $\frac{1}{6}$, well below the actual average precision of $\frac{1}{2}$.

This example demonstrates that Arthur and Merlin can successfully cooperate even with uninformative features, as long as Arthur does not have to classify on features by Morgana alone. This implies that to produce similar bounds as in our setup, the debate mode needs stronger restrictions on either the allowed strategies of Arthur or the structure of the data space, such that this example is excluded.

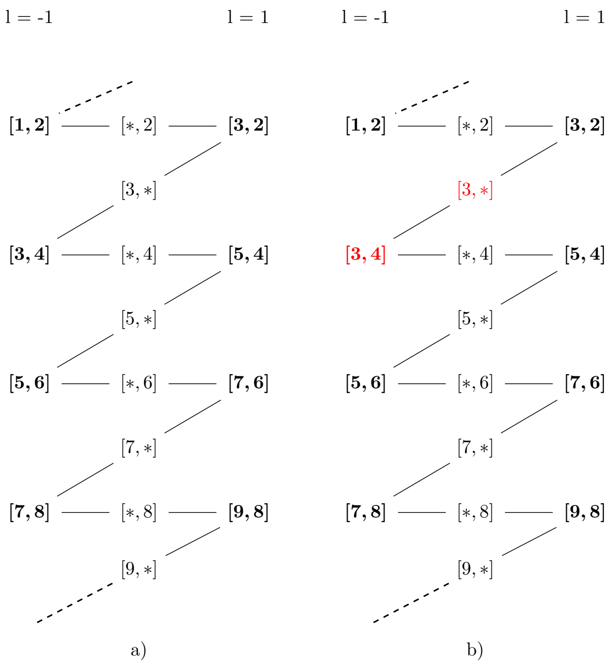
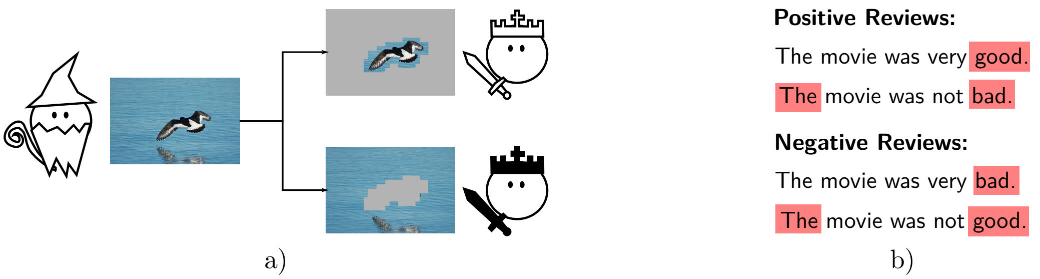


Figure 8: Schematic of $\mathfrak{D}^{\text{ex}}$ as defined in Example A.1. a) The data space forms a bipartite graph, where every data point shares exactly one feature each with two data points from the opposite class. b) Classification on data point $[3, 4]$. Merlin chooses the feature with the smallest 1-norm from this data point, so $[3, *]$. Arthur chooses the class of the data point with the highest 1-norm compatible with the presented features, so correctly $[3, 4]$. Morgana can choose $\varnothing$, $[*, 2]$ or $[3, *]$, but in all cases Arthur can correctly identify the original data point and return class $l = -1$.

A.4.2 Adversarial Classifier

An alternative interactive setup has been proposed in Yu et al. [2019], Dabkowski and Gal [2017], see Figure 9 a) for an illustration. In this setup, a single prover selects a feature from the data point and sends it to a cooperative classifier that decides the class. The rest of the data point is sent to an adversarial classifier that also tries to classify correctly. The aim of the prover is to maximise the probability that the cooperator classifies correctly, and that the adversary cannot perform much better than chance. This setup prevents cheating (in the sense illustrated in Figure 2), because selecting uninformative features would leave the informative features for the adversary. The optimal selections thus captures all the features that are sufficient to decide the class, whereas our Merlin-Arthur setup captures just the features that are necessary to decide the class. We should expect the latter set to be contained in the former.



Positive Reviews:

The movie was very good.

The movie was not bad.

Negative Reviews:

The movie was very bad.

The movie was not good.

Figure 9: a) Illustration of the adversarial classification setup. Here, the prover selects a feature from the data point and sends it the cooperative verifier that decides the class. The data point without the feature is send to an adversary that also aims to decide the class. b) An example dataset that illustrates why this setup might fail to select for informative features. All selected features appear once in each class, thus have zero mutual information. Nevertheless, it is possible to classify perfectly with the selection, but not better than chance with the leftover.

Asymmetric Feature Correlation We can analogously define completeness and soundness values corresponding to the classification accuracy of the cooperator and adversary respectively. But our argument laid out in Figure 4 still holds. Merlin can use the uninformative "fish" and "fruit" features to communicate the class to the cooperator. This works for all images except the two on the left with the many features. These are also the only images where the adversary can then still decide the class. Thus, completeness and soundness can be made arbitrarily good even with uninformative features as long as the AFC can be made arbitrarily large. This implies that the AFC constant of the dataset plays a role in this setup as well.

But even when accounting for the AFC, more care is needed to state bounds. This is illustrated by a counterexample in Figure 9 b). Here, four reviews of a movie are classified as positive or negative. Merlin can use the highlighted sections as features that are sent to the cooperative classifier. Three things are true for this strategy:

1. The mutual information between the features and the class is zero. Every feature appears once in each class.
2. It is possible to classify perfectly with the features since Merlin's selection is unique for each class.
3. It is impossible to classify the leftover text better than chance. Each leftover appears once in each class.

The trick here is of course that Merlin uses the word "The" to indicate negation to his cooperator. This trick works as long as there is a XOR-like relationship in the data that flips the meaning of another feature, in this case the "good" and "bad". These XOR-like features are especially likely in text, but can exist in images as well. Even though the mutual information of the features is zero, they still have a significant overlap with features that actually have high mutual information. So this issue might be resolved by a careful reformulation of the objective of the interactive setup. We also do not claim that this will very likely happen in real-world applications, but it must be dealt with to derive theoretical bounds.

B Theoretical Details

We now give further explanations to our theoretical investigation in the main part as well as provide definitions and proofs for the previously stated theorems and lemmas.

B.1 Conditional entropy and Average Precision

We restate the definition of the average precision and average class conditional entropy to show how one can be bound by the other. The average precision of a feature selector M is defined as

$$\mathrm{Pr}_{\mathcal{D}}(M) := \mathbb{E}_{\mathbf{y}\sim\mathcal{D}}[\mathbb{P}_{\mathbf{x}\sim\mathcal{D}}[c(\mathbf{x}) = c(\mathbf{y}) \,|\, M(\mathbf{y}) \subseteq \mathbf{x}]].$$

The average class conditional entropy with respect to a feature selector is defined as

$$H_{\mathbf{y},\mathbf{x}\sim\mathcal{D}}(c(\mathbf{x}) \,|\, M(\mathbf{y}) \subseteq \mathbf{x}) := \mathbb{E}_{\mathbf{y}\sim\mathcal{D}}[H_{\mathbf{x}\sim\mathcal{D}}(c(\mathbf{x}) \,|\, M(\mathbf{y}) \subseteq \mathbf{x})].$$

We can expand the latter and reorder that expression in the following way:

$$\begin{aligned} H_{\mathbf{y},\mathbf{x}\sim\mathcal{D}}(c(\mathbf{x}) \,|\, M(\mathbf{y}) \subseteq \mathbf{x}) &= -\mathbb{E}_{\mathbf{y}\sim\mathcal{D}}\left[\sum_{l\in\{-1,1\}} P(c(\mathbf{x}) = l \,|\, M(\mathbf{y}) \subseteq \mathbf{x}) \log(P_{\mathbf{x}\sim\mathcal{D}}(c(\mathbf{x}) = l \,|\, M(\mathbf{y}) \subseteq \mathbf{x}))\right] \\ &= -\mathbb{E}_{\mathbf{y}\sim\mathcal{D}}\left[\sum_{l\in\{c(\mathbf{y}),-c(\mathbf{y})\}} P(c(\mathbf{x}) = l \,|\, M(\mathbf{y}) \subseteq \mathbf{x}) \log(P_{\mathbf{x}\sim\mathcal{D}}(c(\mathbf{x}) = l \,|\, M(\mathbf{y}) \subseteq \mathbf{x}))\right] \\ &= \mathbb{E}_{\mathbf{y}\sim\mathcal{D}}[H_b(P_{\mathbf{x}\sim\mathcal{D}}(c(\mathbf{x}) = c(\mathbf{y}) \,|\, M(\mathbf{y}) \subseteq \mathbf{x})], \end{aligned}$$

where $H_b(p) = -p\log(p) - (1-p)\log(1-p)$ is the binary entropy function. Since H_b is a concave function, we can use Jensen's inequality and arrive at the bound

$$H_{\mathbf{y},\mathbf{x}\sim\mathcal{D}}(c(\mathbf{x}) \,|\, M(\mathbf{y}) \subseteq \mathbf{x}) \leq H_b(\mathrm{Pr}_{\mathcal{D}}(M)).$$

We now give a short proof for Lemma 2.6.

Lemma 2.6. *Given $\mathfrak{D} = (D, \mathcal{D}, c)$, $M \in \mathcal{M}(D)$ and $\delta \in [0,1]$. Let $\mathbf{x}, \mathbf{y} \sim \mathcal{D}$, then with probability $1 - \delta^{-1}(1 - \mathrm{Pr}_{\mathcal{D}}(M))$, $M(\mathbf{y})$ is a feature s.t.*

$$\mathbb{P}_{\mathbf{x}\sim\mathcal{D}}[c(\mathbf{x}) = c(\mathbf{y}) \,|\, M(\mathbf{y}) \subseteq \mathbf{x}] \geq 1 - \delta.$$

Proof. The proof follows directly from Markov's inequality, which states that for a non-negative random variable Z and $\delta > 0$

$$\mathbb{P}[Z \geq \delta] \leq \frac{\mathbb{E}[Z]}{\delta}.$$

Choosing $Z = 1 - \mathbb{P}_{\mathbf{x}\sim\mathcal{D}}[c(\mathbf{x}) = c(\mathbf{y}) \,|\, M(\mathbf{y}) \subseteq \mathbf{x}]$ with $\mathbf{y} \sim \mathcal{D}$ leads to the result. □

B.2 Min-Max Theorem

We now present the proof for Theorem 2.7 which we restate here.

Theorem 2.7. *[Min-Max] Let $M \in \mathcal{M}(D)$ be a feature selector and let*

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

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

$$\mathrm{Pr}_{\mathcal{D}'}(M) = 1, \quad \textit{thus} \quad H_{\mathbf{x},\mathbf{y}\sim\mathcal{D}'}(c(\mathbf{y}) \mid M(\mathbf{y}) \subseteq \mathbf{x}) = 0.$$

Proof. From the definition of ϵ_M it follows that there exists a not necessarily unique $A^\sharp \in \mathcal{A}$ such that

$$\max_{\widehat{M}\in\mathcal{M}} \mathbb{P}_{\mathbf{y}\sim\mathcal{D}}\Big[\mathbf{y} \in E_{M,\widehat{M},A^\sharp}\Big] = \epsilon_M. \tag{5}$$

Given $A^\sharp$, an optimal strategy by Morgana is certainly

$$\widehat{M}^\sharp(\mathbf{y}) = \begin{cases} \mathbf{z} \text{ s.t. } A(\mathbf{z}) = -c(\mathbf{y}) & \text{if possible,} \\ \varnothing & \text{otherwise,} \end{cases}$$

and every optimal strategy differs only on a set of measure zero. Thus, we can define

$$D' = D \setminus E_{M,\widehat{M}^\sharp,A^\sharp},$$

and have $\mathbb{P}_{\mathbf{y}\sim\mathcal{D}}[\mathbf{y} \in D'] \geq 1 - \epsilon_M$. We know that $A(M(\mathbf{y})) \neq 0$ when $\mathbf{y} \in D'$ and thus can finally assert that

$$\forall \mathbf{y}, \mathbf{x} \in D' : M(\mathbf{y}) \subseteq \mathbf{x} \Rightarrow c(\mathbf{x}) = c(\mathbf{y}).$$

Otherwise there would be at least one $\mathbf{x} \in D'$ that would also be in $E_{M,\widehat{M}^\sharp,A^\sharp}$. Thus, we conclude $\mathrm{Pr}_{\mathcal{D}'}(M) = 1$, and from

$$0 \leq H_{\mathbf{y},\mathbf{x}\sim\mathcal{D}'}(c(\mathbf{x}) \mid \mathbf{x} \in M(\mathbf{y})) \leq H_b(\mathrm{Pr}_{\mathcal{D}'}(M)) = 0,$$

it follows that $H_{\mathbf{y},\mathbf{x}\sim\mathcal{D}'}(c(\mathbf{x}) \mid M(\mathbf{y}) \subseteq \mathbf{x}) = 0$. □

This theorem states that if Merlin uses a strategy that allows Arthur to classify almost always correctly, thus small ϵ_M, then there exists a dataset that covers almost the entire original dataset and on which the class entropy conditioned on the features selected by Merlin is zero.

This statement with a new set D' appears convoluted at first, and we would prefer a simple bound, such as

$$\mathrm{Pr}_{\mathcal{D}}(M) \geq 1 - \epsilon_M,$$

where we take the average precision over the whole dataset. This is, however, not straightforwardly possible due to a principle we call *asymmetric feature correlation* and which we introduce in the next section.

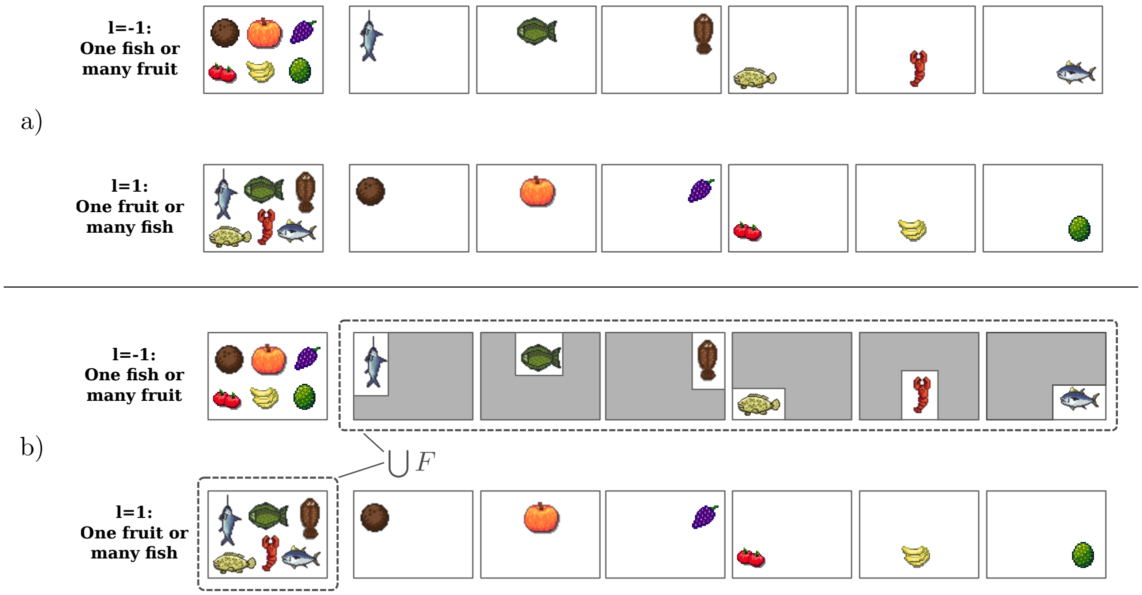


Figure 10: Illustration of a data space with strong asymmetric feature correlation. *a)* A dataset with fish and fruit features. The features are asymmetrically correlated, because all fruit features are maximally correlated in class -1 (they are all in the same image) and maximally uncorrelated in 1 (no two fruits share the same image). The reverse is true for the fruits. See Figure 11 for a strategy for Merlin that ensures strong completeness and soundness with uninformative features.
b) Asymmetric feature correlation for a specific feature selection. For the class -1, we select the set F of all "fish" features. Each individual fish feature in F covers a fraction of $\frac{1}{6}$ of $(F^*) \cap D_{-1}$ and all images (one) in $(F^*) \cap D_1$. The expected value in Equation (6) thus evaluates to $k = 6$. This is also the maximum AFC for the entire dataset as no different feature selection gives a higher value.

B.3 Asymmetric Feature Correlation

Asymmetric feature correlation (AFC) is a concept that will allow us to state our main result. It measures if there is a set of features that are concentrated in a few data points in one class, but spread out over almost all data points in the other class. This represents a possible quirk of datasets that can result in a scenario where Arthur and Merlin cooperate successfully with high probability, Morgana is unable to fool Arthur except with low probability– and yet the features exchanged by Arthur and Merlin are uninformative for the class. An illustration of such an unusual dataset is given in Figure 10.

For an illustration of the asymmetric feature correlation, consider two-class data space $\mathfrak{D} = \{D, \mathcal{D}, c\}$, e.g. the "fish and fruit" data depicted in Figure 10. Let us choose $F \subset D_p$ to be all the "fish" features. We see that these features are strongly anti-concentrated in class $l = -1$ (none of them share an image) and strongly concentrated in class $l = 1$ (all of them are in the same image).

For now, let us assume F is finite and let $\mathcal{F} = \mathcal{U}(F)$, the uniform measure over F. Let us recall that $F^* := \{\mathbf{x} \in D \mid \exists\, \mathbf{z} \in F : \mathbf{z} \subseteq \mathbf{x}\}$. We have strong AFC if the class-wise ratio of what each feature covers individually is much larger than what the features cover as a whole:

$$\mathbb{E}_{\mathbf{z}\sim\mathcal{F}}\left[\frac{\mathbb{P}_{\mathbf{x}\sim\mathcal{D}_{-l}}[\mathbf{z}\subseteq\mathbf{x}]}{\mathbb{P}_{\mathbf{x}\sim\mathcal{D}_{l}}[\mathbf{z}\subseteq\mathbf{x}]}\right] \gg \frac{\mathbb{P}_{\mathbf{x}\sim\mathcal{D}_{-l}}[\mathbf{x}\in F^*]}{\mathbb{P}_{\mathbf{x}\sim\mathcal{D}_{l}}[\mathbf{x}\in F^*]}.$$

In our example, every individual "fish" feature covers one image in each class, so the left side is equal to 1. As a feature set, they cover six images in class -1 and one in class 1, so the right side is $\frac{1}{6}$. Using

$$\frac{\mathbb{P}[\mathbf{z}\subseteq\mathbf{x}]}{\mathbb{P}[\mathbf{x}\in F^*]} = \mathbb{P}[\mathbf{z}\subseteq\mathbf{x}\,|\,\mathbf{x}\in F^*],$$

we can restate this expression as

$$\mathbb{E}_{\mathbf{z}\sim\mathcal{F}}[\kappa_l(\mathbf{z}, F)] \gg 1, \tag{6}$$

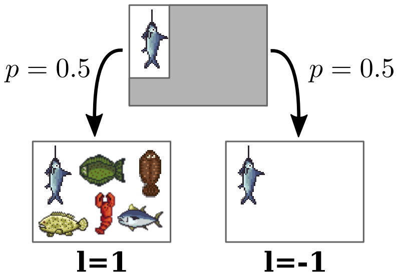


Figure 11: In the dataset presented in Figure 10, Merlin can use the strategy to always select the fish features for class $l = -1$ and the fruit features for class $l = 1$ if they exist and choose something arbitrary otherwise. Arthur can then guess $l = 1$ if he gets a fish and $l = -1$ for a fruit. This strategy fails only for the images containing all fruits or fish, and can only be exploited by Morgana for those same two images out of 14. The completeness and soundness constants in this case are both $\frac{1}{7}$. But as illustrated here, each "fish" feature is completely uninformative of the class. Conditioned on the selected fish, it could either be the image from class $l = -1$ or from $l = 1$.

where

$$\kappa_l(\mathbf{z}, F) = \frac{\mathbb{P}_{\mathbf{x}\sim\mathcal{D}_{-l}}[\mathbf{z} \subseteq \mathbf{x} \,|\, \mathbf{x} \in F^*]}{\mathbb{P}_{\mathbf{x}\sim\mathcal{D}_l}[\mathbf{z} \subseteq \mathbf{x} \,|\, \mathbf{x} \in F^*]}.$$

For our "fish" features we have $\kappa_{-1}(\mathbf{z}, F) = 6$ for every feature $\mathbf{z} \in F$. Considering an infinite set F, we need a way to get a reasonable measure $\mathcal{F}$, where we don't want to "overemphasise" features that appear only in very few data points. We thus define a feature selector $f_F \in \mathcal{M}(F^*)$ as

$$f_F(\mathbf{x}) = \underset{\substack{\mathbf{z}\in F \\ \text{s.t. } \mathbf{z}\subseteq\mathbf{x}}}{\operatorname{argmax}} \, \kappa(\mathbf{z}, F), \tag{7}$$

and we can define the push-forward measure $\mathcal{F} = f_{F*}\mathcal{D}_l|_{F^*}$. For our fish and fruit example, where F is the set of all fish features, f_{F*} would select a fish feature for every image in class -1 that is in F^*. Putting everything together, we get the following definition.

Definition B.1 (Asymmetric feature correlation)**.** *Let $(D, \mathcal{D}, c)$ be a two-class data space, then the asymmetric feature correlation κ is defined as*

$$\kappa = \max_{l\in\{-1,1\}} \max_{F\subset D_p} \mathbb{E}_{\mathbf{y}\sim\mathcal{D}_l|_{F^*}} \left[\max_{\substack{\mathbf{z}\in F \\ s.t. \ \mathbf{z}\subseteq\mathbf{y}}} \frac{\mathbb{P}_{\mathbf{x}\sim\mathcal{D}_{-l}}[\mathbf{z} \subseteq \mathbf{x} \,|\, \mathbf{x} \in F^*]}{\mathbb{P}_{\mathbf{x}\sim\mathcal{D}_l}[\mathbf{z} \subseteq \mathbf{x} \,|\, \mathbf{x} \in F^*]} \right].$$

As we have seen in the "fish and fruit" example, the AFC can be made arbitrarily large, as long as one can fit many individual features into a single image. We can prove that the maximum amount of features per data point indeed also gives an upper bound on the AFC. We now come back to Lemma 2.9 and prove it.

Lemma 2.9. *Let $\mathfrak{D}$ be a two-class data space with AFC of κ. Let $K = \max_{\mathbf{x}\in D} |\{\mathbf{z} \in D_p \,|\, \mathbf{z} \subseteq \mathbf{x}\}|$ be the maximum number of features per data point. Then $\kappa \leq K$.*

Proof. Let $l \in \{-1, 1\}$ and let $F \subset D_p$. We define $f_F \in \mathcal{M}(F^*)$ as in Equation (7) as well as $\mathcal{F} = f_{F*}\mathcal{D}_l|_{F^*}$. We can assert that

$$\mathbb{P}_{\mathbf{x}\sim\mathcal{D}_l}[\mathbf{z} \subseteq \mathbf{x} \,|\, \mathbf{x} \in F^*] \geq \mathbb{P}_{\mathbf{x}\sim\mathcal{D}_l}[f(\mathbf{x}) = \mathbf{z} \,|\, \mathbf{x} \in F^*] = \mathcal{F}(\mathbf{z}). \tag{8}$$

We then switch the order of taking the expectation value in the definition of the AFC:

$$\begin{aligned}
\kappa_l(F) &= \mathbb{E}_{\mathbf{z}\sim\mathcal{F}} \left[\frac{\mathbb{P}_{\mathbf{x}\sim\mathcal{D}_{-l}}[\mathbf{x} \in F \,|\, \mathbf{x} \in F^*]}{\mathbb{P}_{\mathbf{y}\sim\mathcal{D}_l}[\mathbf{z} \subseteq \mathbf{y} \,|\, \mathbf{y} \in F^*]} \right] \\
&= \mathbb{E}_{\mathbf{z}\sim\mathcal{F}} \left[\frac{\mathbb{E}_{\mathbf{x}\sim\mathcal{D}_{-l}}[\mathbf{1}(\mathbf{z} \subseteq \mathbf{x}) \,|\, \mathbf{x} \in F^*]}{\mathbb{P}_{\mathbf{y}\sim\mathcal{D}_l}[\mathbf{z} \subseteq \mathbf{y} \,|\, \mathbf{y} \in F^*]} \right] \\
&= \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_{-l}} \left[\mathbb{E}_{\mathbf{z}\sim\mathcal{F}} \left[\frac{\mathbf{1}(\mathbf{z} \subseteq \mathbf{x})}{\mathbb{P}_{\mathbf{y}\sim\mathcal{D}_l}[\mathbf{z} \subseteq \mathbf{y} \,|\, \mathbf{y} \in F^*]} \right] \middle| \, \mathbf{x} \in F^* \right].
\end{aligned}$$

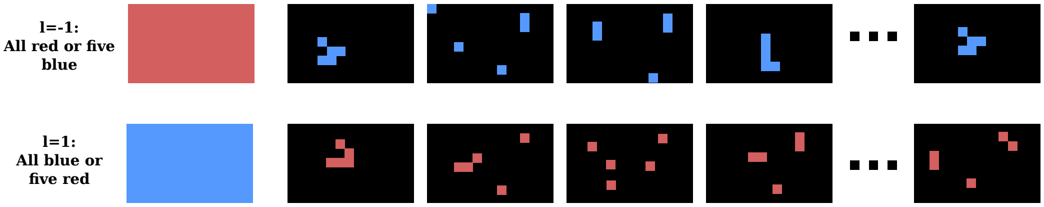


Figure 12: An example of a dataset with very high asymmetric feature correlation. The completely red image shares a feature with each of the m-red-pixel images (here $m = 5$), of which there are $\binom{d}{m}$ many. In the worst case $m = \frac{d}{2}$, resulting in $k = \binom{d}{d/2}$ thus exponential growth in d.

Since there are only finitely many features in a data point we can express the expectation value over a countable sum weighted by the probability of each feature:

$$\begin{aligned}\kappa_l(F) &= \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_{-l}}\left[\sum_{\mathbf{z}\in F\colon \mathbf{x}\in F}\left[\frac{\mathcal{F}(\mathbf{z})}{\mathbb{P}_{\mathbf{y}\sim\mathcal{D}_l}[\mathbf{z}\subseteq\mathbf{y}\,|\,\mathbf{y}\in F^*]}\right]\middle|\,\mathbf{x}\in F^*\right]\\ &\leq \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_{-l}}\left[\sum_{\mathbf{z}\in F\colon \mathbf{x}\in F} 1\,\middle|\,\mathbf{x}\in F^*\right]\\ &\leq \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_{-l}}[K\,|\,\mathbf{x}\in F^*]\\ &= K,\end{aligned}$$

where in the first step we used Equation (8) and the definition of K in the second. Then we see that

$$\kappa = \max_{l\in\{-1,1\}}\max_{F\subset D_p}\kappa_l(F)\leq K.$$

□

The number of features per data point is dependent on which kinds of features we consider. Without limitations, this number can be 2^d, i.e., exponentially high. See Figure 12 for an example of exponentially large AFC parameters. If we consider only features of a fixed size and shape, such as in image data, the number of features per data point drops to $\approx d$.

We now prove an intermediate lemma that will later allow us to prove Theorem 2.11.

Lemma B.2. *Let $(D, \mathcal{D}, c)$ be a two-class data space with asymmetric feature correlation of κ and class imbalance B. Let $A : [0,1]^d \to \{-1,0,1\}$ be a feature classifier and $M \in \mathcal{M}(D)$ a feature selector for D. From*

1. *Completeness:*

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

2. *Soundness:*

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

follows

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

This lemma gives a bound on the probability that data points with features selected by Merlin will also belong to the same class. This probability is large as long as we have a bound on the AFC of the dataset.

Proof. The proof of our lemma is fairly intuitive, although the notation can appear cumbersome. Here we give a quick overview over the proof steps.

1. In the definition of the AFC, we maximise over all possible features sets. We will choose as a special case (for each class $l \in \{-1, 1\}$) the features that Merlin selects for data points that Arthur classifies correctly.
2. These feature sets cover the origin class at least up to $1 - \epsilon_c$, and the other class at most up to ϵ_s, which is required by the completeness and soundness criteria, respectively. This gives us a high precision for the whole feature set.
3. The AFC upper bounds the quotient of the precision of the whole feature set and expected precision of the individual features, which finally allows us to state our result.

Let us define a partition of D according to the true class and the class assigned by Arthur. From now on, let $l \in \{-1, 1\}$ and $m \in \{-1, 0, 1\}$. We introduce the datasets

$$D_{l,m} = \{\mathbf{x} \in D_l \,|\, A(M(\mathbf{x})) = m\},$$

which means that $D_{l,l}$ are the data points that Arthur classifies correctly, and furthermore

$$F_l = M(D_{l,l}).$$

To use the AFC bound we need a feature selector $f : D_l|_{F^*} \to F$. Merlin itself maps to features outside F when applied to data points in $D_l|_{F_l^*} \setminus D_{l,l}$. Let us thus define $\sigma : D_F \setminus D_{l,l} \to F$ which selects an arbitrary feature from F (in case one is concerned whether such a representative can always be chosen, consider a discretised version of the data space which allows for an ordering). Then we can define

$$f(\mathbf{x}) = \begin{cases} M(\mathbf{x}) & \mathbf{x} \in D_{l,l}, \\ \sigma(\mathbf{x}) & \mathbf{x} \in D_l|_{F_l^*} \setminus D_{l,l}, \end{cases} \quad \text{and} \quad \mathcal{F}_l = f_* D_l|_{F_l^*}.$$

This feature selector will allow us to use the AFC bound. We now abbreviate

$$p(\mathbf{x}, f) = \mathbb{P}_{\mathbf{y}\sim\mathcal{D}}[c(\mathbf{y}) \neq c(\mathbf{x}) \,|\, \mathbf{y} \in f(\mathbf{x})] \quad \text{and} \quad P_l = \mathbb{P}_{\mathbf{x}\sim\mathcal{D}}[\mathbf{x} \in D_l].$$

Then

$$1 - \mathrm{Pr}_{\mathcal{D}}(M) = \mathbb{E}_{\mathbf{x}\sim\mathcal{D}}[p(\mathbf{x}, M)] = \sum_{l \in \{-1,1\}} \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_l}[p(\mathbf{x}, M)] P_l. \tag{9}$$

Using the completeness criterion and the fact that $p(\mathbf{x}, M) \leq 1$ we can bound

$$\begin{aligned}
\mathbb{E}_{\mathbf{x}\sim\mathcal{D}_l}[p(\mathbf{x}, M)] &= \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_l}[p(\mathbf{x}, M)\mathbf{1}(\mathbf{x} \in D_{l,l})] + \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_l}[p(\mathbf{x}, M)\mathbf{1}(\mathbf{x} \notin D_{l,l})] \\
&\leq \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_l}[p(\mathbf{x}, M)\mathbf{1}(\mathbf{x} \in D_{l,l})] + \epsilon_c \\
&\leq \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_l}[p(\mathbf{x}, M)\mathbf{1}(\mathbf{x} \in D_{l,l})] + \epsilon_c + \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_l}\left[p(\mathbf{x}, \sigma)\mathbf{1}\big(\mathbf{x} \in D_l|_{F_l^*} \setminus D_{l,l}\big)\right] \\
&\leq \frac{\mathbb{E}_{\mathbf{x}\sim\mathcal{D}_l}\left[p(\mathbf{x}, M)\mathbf{1}(\mathbf{x} \in D_{l,l}) + p(\mathbf{x}, \sigma)\mathbf{1}\big(\mathbf{x} \in D_l|_{F_l^*} \setminus D_{l,l}\big)\right]}{\mathbb{P}_{\mathbf{x}\sim\mathcal{D}_l}\left[\mathbf{x} \in D_l|_{F_l^*}\right]} + \epsilon_c \\
&= \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_l|_{F_l^*}}[p(\mathbf{x}, f)] + \epsilon_c \\
&= \mathbb{E}_{\mathbf{z}\sim\mathcal{F}_l}[\mathbb{P}_{\mathbf{y}\sim\mathcal{D}}[c(\mathbf{y}) = -l \,|\, \mathbf{z} \subseteq \mathbf{y}]] + \epsilon_c.
\end{aligned}$$

We can expand the expression in the expectation using Bayes' Theorem:

$$\begin{aligned}
\mathbb{P}_{\mathbf{y}\sim\mathcal{D}}[c(\mathbf{y}) = -l \,|\, \mathbf{y} \in \mathbf{z}] &= \frac{\mathbb{P}_{\mathbf{y}\sim\mathcal{D}_{-l}}[\mathbf{z} \subseteq \mathbf{y}]\mathbb{P}_{-l}}{\mathbb{P}_{\mathbf{y}\sim\mathcal{D}_{-l}}[\mathbf{z} \subseteq \mathbf{y}]P_{-l} + \mathbb{P}_{\mathbf{y}\sim\mathcal{D}_l}[\mathbf{z} \subseteq \mathbf{y}]P_l} \\
&= h\left(\frac{\mathbb{P}_{\mathbf{y}\sim\mathcal{D}_{-l}}[\mathbf{z} \subseteq \mathbf{y}]\mathbb{P}_{-l}}{\mathbb{P}_{\mathbf{y}\sim\mathcal{D}_l}[\mathbf{z} \subseteq \mathbf{y}]P_l}\right),
\end{aligned}$$

where $h(t) = (1 + t^{-1})^{-1}$. Since $h(t)$ is a concave function for $t \geq 0$, we know that for any random variable R have $\mathbb{E}[h(R)] \leq h(\mathbb{E}[R])$, so

$$\mathbb{E}_{\mathbf{x}\sim\mathcal{D}_l}[p(\mathbf{x}, M)] \leq h\left(\mathbb{E}_{\mathbf{z}\sim\mathcal{F}_l}\left[\frac{\mathbb{P}_{\mathbf{y}\sim\mathcal{D}_{-l}}[\mathbf{z} \subseteq \mathbf{y}]}{\mathbb{P}_{\mathbf{y}\sim\mathcal{D}_l}[\mathbf{z} \subseteq \mathbf{y}]}\right]\frac{P_{-l}}{P_l}\right) + \epsilon_c. \tag{10}$$

From the definition of the AFC κ we know that

$$\mathbb{E}_{\mathbf{z}\sim\mathcal{F}_l}\left[\frac{\mathbb{P}_{\mathbf{y}\sim\mathcal{D}_{-l}}[\mathbf{z}\subseteq\mathbf{y}]}{\mathbb{P}_{\mathbf{y}\sim\mathcal{D}_l}[\mathbf{z}\subseteq\mathbf{y}]}\right] \leq \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_l|_{F_l^*}}\left[\max_{\substack{\mathbf{z}\in F\\ \text{s.t. } \mathbf{z}\subseteq\mathbf{x}}}\frac{\mathbb{P}_{\mathbf{y}\sim\mathcal{D}_{-l}}[\mathbf{z}\subseteq\mathbf{y}]}{\mathbb{P}_{\mathbf{y}\sim\mathcal{D}_l}[\mathbf{z}\subseteq\mathbf{y}]}\right] \leq \kappa\frac{\mathbb{P}_{\mathbf{y}\sim\mathcal{D}_{-l}}[\mathbf{y}\in F^*]}{\mathbb{P}_{\mathbf{y}\sim\mathcal{D}_l}[\mathbf{y}\in F^*]}. \tag{11}$$

Now we make use of the fact that we can lower bound $\mathbb{P}_{\mathbf{y}\sim\mathcal{D}_l}[\mathbf{y}\in F]$ by the completeness property

$$\mathbb{P}_{\mathbf{y}\sim\mathcal{D}_l}[\mathbf{y}\in F^*] \geq 1-\epsilon_c,$$

and upper bound $\mathbb{P}_{\mathbf{y}\sim\mathcal{D}_{-l}}[\mathbf{y}\in F^*]$ with the soundness property

$$\mathbb{P}_{\mathbf{y}\sim\mathcal{D}_{-l}}[\mathbf{y}\in F^*] \leq \epsilon_s.$$

This is because $\mathbf{y}\in F^*$ implies that there are features Morgana can use to convince Arthur of class l whereas $\mathbf{y}\sim\mathcal{D}_{-l}$. Together with Equation (10) and Equation (11) we arrive at

$$\mathbb{E}_{\mathbf{x}\sim\mathcal{D}_l}[p(\mathbf{x},M)] \leq h\left(\kappa\frac{\epsilon_s}{1-\epsilon_c}\frac{P_{-l}}{P_l}\right)+\epsilon_c = \frac{\kappa\epsilon_s\frac{P_{-l}}{P_l}}{1-\epsilon_c+\kappa\epsilon_s\frac{P_{-l}}{P_l}}+\epsilon_c.$$

Using $\frac{P_l}{P_{-l}}\leq B$ thus $\frac{P_{-l}}{P_l}\geq B^{-1}$, we can express

$$\mathbb{E}_{\mathbf{x}\sim\mathcal{D}_l}[p(\mathbf{x},M)] \leq \frac{\kappa\epsilon_s\frac{P_{-l}}{P_l}}{1-\epsilon_c+\kappa\epsilon_s B^{-1}}+\epsilon_c.$$

Inserted back into equation Equation (9) leads us to

$$1-\text{Pr}_{\mathcal{D}}(M) \leq \frac{\kappa\epsilon_s}{1-\epsilon_c+\kappa\epsilon_s B^{-1}}+\epsilon_c.$$

□

Bounding the AFC is a hard task that is not possible for most real-world dataset. This is why we evaluate Assumption 2.12 on the UCI and MNIST dataset. Here, we want to give further intuition why we think that we are unlikely to encounter effects of high AFC when training Merlin-Arthur classifiers, even if the dataset theoretically has a high AFC:

1. A learning barrier: Merlin and Arthur's strategy cannot be arbitrary, because it has to be learned on the training dataset and generalise to the test dataset. This precludes any set of features (for example an intricate set of pixels) that is over-optimised on the training data.
2. A computational barrier: We conjecture that for datasets where the AFC is computationally hard to estimate, it will also be hard for Merlin and Arthur to exploit. There are evidence provided by Wäldchen, which looks at the hardness of deceptive certificate selection.

B.4 Relative Success Rate and Realistic Algorithms

As discussed in Section 2.4 realistic algorithms will not be optimal players as in Theorem 2.7. It will turn out that we can relax the requirement for Morgana to play optimally in two important ways: (i) She is only required to find features that can also be found by Merlin (2) She only needs success on a similar rate as Merlin. Thus, what's crucial is the relative strength between the algorithms utilized for Merlin and Morgana. We can define *relative success rate* in the following way:

Definition B.3 (Relative Success Rate)**.** *Let $\mathfrak{D}=(D,\mathcal{D},c)$ be a two-class data space. Let $A\in\mathcal{A}$ and $M,\widehat{M}\in\mathcal{M}(D)$ Then the relative success rate α of $\widehat{M}$ with respect to A,M and $\mathfrak{D}$ is defined as*

$$\alpha := \min_{l\in\{-1,1\}}\frac{\mathbb{P}_{\mathbf{x}\sim\mathcal{D}_{-l}}\left[A(\widehat{M}(\mathbf{x}))=l\,|\,\mathbf{x}\in F_l^*\right]}{\mathbb{P}_{\mathbf{x}\sim\mathcal{D}_l}[A(M(\mathbf{x}))=l\,|\,\mathbf{x}\in F_l^*]}, \quad \textit{where} \quad F_l^* := \{\mathbf{x}\in D\,|\,\exists\mathbf{z}\subseteq\mathbf{x}:\ \mathbf{z}\in M(D_l), A(\mathbf{z})=l\}$$

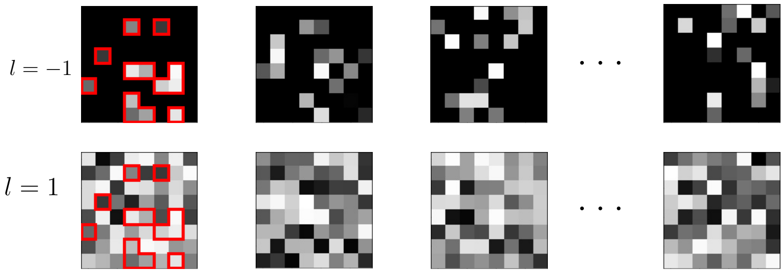


Figure 13: Illustration of a dataset where the complexity of finding the relevant features depends strongly on the the irrelevant features. Class -1 consists of k-sparse images whose pixel values sum to some number s. For each of these images, there is a non-sparse image in class 1 that shares all non-zero values (marked in red for the first image). Merlin can use the strategy to show all k non-zero pixels for an image from class -1 and $k+1$ arbitrary non-zero pixels for class 1. Arthur checks if the sum is equal to s or if the number of pixels equal to $k+1$, otherwise he says "I don't know!". He will then classify with 100% accuracy. Nevertheless, the features Merlin uses for class -1 are completely uncorrelated with the class label. To exploit this, however, Morgana would have to solve the NP-hard [Kleinberg and Tardos, 2006] subset sum problem to find the pixels for images in class 1 that sum to s. The question is not in which class we can find the features, but in which class we can find the features *efficiently.*

The set F_l contains all features that Merlin selects in class l that successfully convince Arthur of class l, and F_l^* is the set of all data points containing such a feature. We condition on the fact that a feature that Merlin uses is in the data point and then compare the rates of Morgana and Merlin convincing Arthur. Morgana can of course also select different features but is not required to find features that also Merlin could not find.

Given that one of Merlins features is in the data point, the question is thus how much the context given by the other features affect the hardness of finding of the former. We can easily construct scenarios in which the context matters very strongly, see Figure 13 for an example. We expect that for most real-world data this dependence is only weak and can be upper bounded.

Example B.4. *Let $\mathfrak{D}$ be a data space with maximum number of features per data point K. Let Morgana operate the algorithm described in Algorithm 2, in which she randomly searches for a convincing feature. Then we have*

$$\alpha \geq \frac{K}{n_{try}},$$

which corresponds to an upper bound on the probability that Morgana will succeed on an individual data point when there is only one convincing feature.

Algorithm 2 Merlin-Arthur Training

```
Input: $\mathbf{x} \in D$, $n_{\text{try}}$
Output: $\mathbf{z} \in D_p$
for $i \in [n_{\text{try}}]$ do
    Pick random feature $\mathbf{z}$ s.t. $\mathbf{z} \in \mathbf{x}$
    if $A(\mathbf{z}) = -c(\mathbf{x})$ then
        return $\mathbf{z}$
    end if
end for
return $\varnothing$
```

We generally want Morgana's algorithm to be at least as powerful as Merlin's so in case of an optimiser one can consider giving more iterations or more initial starting values.

We now want to prove Theorem 2.11 which we restate here.

Theorem 2.11. *Let $\mathfrak{D} = (D, \mathcal{D}, c)$ be a two-class data space with AFC of κ and class imbalance B. Let $A \in \mathcal{A}$, and $M, \widehat{M} \in \mathcal{M}(D)$ such that $\widehat{M}$ has a relative success rate of α with respect to A, M and $\mathfrak{D}$. Define*

1. *Completeness:*

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

2. *Soundness*

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

Then it follows that

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

Proof. We follow the same proof steps and definitions as in the proof of Lemma B.2 up to Equation (11). Then we consider the following

$$\alpha\frac{\mathbb{P}_{\mathbf{y}\sim\mathcal{D}_{-l}}[\mathbf{y}\in F]}{\mathbb{P}_{\mathbf{y}\sim\mathcal{D}_{l}}[\mathbf{y}\in F]} \leq \frac{\mathbb{P}_{\mathbf{y}\sim\mathcal{D}_{-l}}[\mathbf{y}\in F]}{\mathbb{P}_{\mathbf{y}\sim\mathcal{D}_{l}}[\mathbf{y}\in F]}\frac{\mathbb{P}_{\mathbf{x}\sim\mathcal{D}_{-l}}\left[A(\widehat{M}(\mathbf{x}))=l\,|\,\mathbf{x}\in F_l^*\right]}{\mathbb{P}_{\mathbf{x}\sim\mathcal{D}_{l}}[A(M(\mathbf{x}))=l\,|\,\mathbf{x}\in F_l^*]} \tag{12}$$

$$= \frac{\mathbb{P}_{\mathbf{x}\sim\mathcal{D}_{-l}}\left[A(\widehat{M}(\mathbf{x}))=l,\,\mathbf{x}\in F_l^*\right]}{\mathbb{P}_{\mathbf{x}\sim\mathcal{D}_{l}}[A(M(\mathbf{x}))=l,\,\mathbf{x}\in F_l^*]} \tag{13}$$

$$\leq \frac{\mathbb{P}_{\mathbf{x}\sim\mathcal{D}_{-l}}\left[A(\widehat{M}(\mathbf{x}))=l\right]}{\mathbb{P}_{\mathbf{x}\sim\mathcal{D}_{l}}[A(M(\mathbf{x}))=l]} \tag{14}$$

where in the last step we used that $\mathbf{x}\in D_l \wedge A(M(\mathbf{x}))=l \Rightarrow \mathbf{x}\in F_l^*$ by definition of F_l^*. We know by the soundness and completeness criteria that

$$\mathbb{P}_{\mathbf{x}\sim\mathcal{D}_{-l}}[A(M(\mathbf{x},l))=l]\leq\epsilon_s \quad\text{and}\quad \mathbb{P}_{\mathbf{x}\sim\mathcal{D}_{l}}[M(\mathbf{x},l)\in F]\geq 1-\epsilon_c$$

Putting everything together we arrive at

$$\frac{\mathbb{P}_{\mathbf{y}\sim\mathcal{D}_{-l}}[\mathbf{y}\in F]}{\mathbb{P}_{\mathbf{y}\sim\mathcal{D}_{l}}[\mathbf{y}\in F]}\leq\frac{\alpha^{-1}\epsilon_s}{1-\epsilon_c},$$

which allows us to continue the proof analogously to Lemma B.2. □

B.5 Finite and Biased datasets

Real datasets come with further challenges when evaluating the completeness and soundness.

Let us introduce two data distributions $\mathcal{T}$ and $\mathcal{D}$ on the same dataset D, where $\mathcal{T}$ is considered the "true" distribution and $\mathcal{D}$ a different, potentially biased, distribution. We define this bias with respect to a specific feature $\mathbf{z}\in D_p$ as

$$d^l_{\mathbf{z}}(\mathcal{D},\mathcal{T}) := |\mathbb{P}_{\mathbf{x}\in\mathcal{D}}[c(\mathbf{x})=l\,|\,\mathbf{z}\in\mathbf{x}]-\mathbb{P}_{\mathbf{x}\in\mathcal{T}}[c(\mathbf{x})=l\,|\,\mathbf{z}\in\mathbf{x}]|.$$

Note that $d^1_{\mathbf{z}}(\mathcal{D},\mathcal{T}) = d^{-1}_{\mathbf{z}}(\mathcal{D},\mathcal{T}) =: d_{\mathbf{z}}(\mathcal{D},\mathcal{T})$ and $0\leq d_{\mathbf{z}}(\mathcal{D},\mathcal{T})\leq 1$. This distance measures if data points containing $\mathbf{z}$ are distributed differently to the two classes for the two distributions.

For example, consider as $\mathbf{z}$ the water in the boat images of the PASCAL VOC dataset Lapuschkin et al. [2019]. The feature is a strong predictor for the "boat" class in the test data distribution $\mathcal{D}$ but should not be indicative for the real world distribution which includes images of water without boats and vice versa. We now want to prove that a feature selected by M is either an informative feature or is misrepresented in the test dataset.

Lemma B.5. *Let $\mathfrak{D}, k, B, A, M, \alpha, \epsilon_c$ and ϵ_s be defined as in Theorem 2.11. Let $\mathcal{T}$ be the true data distribution on D. Then for $\delta\in[0,1]$ we have*

$$\mathbb{P}_{\mathbf{y}\sim\mathcal{T}}[c(\mathbf{y})=c(\mathbf{x})\,|\,M(\mathbf{x})\subseteq\mathbf{y}]\geq 1-\delta-d_{M(\mathbf{x})}(\mathcal{D},\mathcal{T}),$$

with probability $1-\frac{1}{\delta}\left(\frac{k\alpha\epsilon_s}{1+k\alpha\epsilon_s B^{-1}-\epsilon_c}+\epsilon_c\right)$ for $\mathbf{x}\sim\mathcal{D}$.

This follows directly from Lemma 2.6, Theorem 2.11, the definition of $d_{\mathbf{z}}(\mathcal{D},\mathcal{T})$ and the triangle inequality. This means that if an unintuitive feature was selected in the test dataset, we can pinpoint to where the dataset might be biased.

We provided Lemma B.5 in the context of biased datasets. The next iteration considers the fact that we only sample a finite amount of data from the possibly biased test data distribution. This will only give us an approximate idea of the soundness and completeness constants.

Lemma B.6. *Let $D, \sigma, \mathcal{D}, cA, M$ and $\mathcal{T}$ be defined as in Lemma B.5. Let $D^{test} = (\mathbf{x}_i)_{i=1}^N\sim\mathcal{D}^N$ be N random samples from $\mathcal{D}$. Let*

$$\epsilon_c^{test} = \max_{l\in\{-1,1\}}\frac{1}{N}\sum_{\mathbf{x}\in D_l^{test}}\mathbf{1}(A(M(\mathbf{x},c(\mathbf{x})))\neq c(\mathbf{x})),$$

and

$$\epsilon_s^{test} = \max_{l\in\{-1,1\}} \frac{1}{N} \sum_{\mathbf{x}\in D_l^{test}} \mathbf{1}(A(M(\mathbf{x}, -c(\mathbf{x}))) = -c(\mathbf{x})).$$

Then it holds with probability $1-\eta$ *where* $\eta \in [0,1]$ *that on the true data distribution* $\mathcal{T}$ A *and* M *obey completeness and soundness criteria with*

$$\epsilon_c \leq \epsilon_c^{test} + \epsilon_{dist} + \epsilon_{sample} \quad \textit{and}$$
$$\epsilon_s \leq \epsilon_s^{test} + \epsilon_{dist} + \epsilon_{sample}$$

respectively, where $\epsilon_{dist} = \max_{l\in\{-1,1\}} \|\mathcal{D}_l - \mathcal{T}_l\|_{TV}$ *and* $\epsilon_{sample} = \sqrt{\frac{1}{2N}\log\left(\frac{4}{\eta}\right)}$.

The proof follows trivially from Hoeffding's inequality and the definition of the total variation norm.

Proof. We define $E_{c,l} = \{\mathbf{x} \in D_l \,|\, A(M(\mathbf{x}, c(\mathbf{x}))) \neq c(\mathbf{x})\}$ for $l \in \{-1,1\}$ and let $E_{c,l}^{\mathcal{D}}$ be the Bernoulli random variable for the event that $X \in E_{c,l}$ where $X \sim \mathcal{D}$. Then

$$\mathbb{P}_{\mathbf{x}\sim\mathcal{D}_l}[A(M(\mathbf{x}, c(\mathbf{x}))) \neq c(\mathbf{x})] = \mathbb{E}\left[E_{c,l}^{\mathcal{D}}\right].$$

Using Hoeffding's inequality, we can bound for any $t > 0$

$$\mathbb{P}\left[\left|\left(\frac{1}{N}\sum_{\mathbf{x}\in D_l^{\text{test}}} \mathbf{1}(\mathbf{x} \in E_{c,l})\right) - \mathbb{E}\left[E_{c,l}^{\mathcal{D}}\right]\right| > t\right] \leq e^{-2nt^2}.$$

We choose t such that $e^{-2t^2} = \frac{\eta}{4}$. We use a union bound for the maximum over $l \in \{-1,1\}$ which results in a probability of $2\frac{\eta}{4} = \frac{\eta}{2}$ we have

$$\max_{l\in\{-1,1\}} \mathbb{E}\left[E_{c,l}^{\mathcal{D}}\right] > \epsilon_c^{\text{test}} + \sqrt{\frac{1}{2N}\log\left(\frac{4}{\eta}\right)},$$

and thus with $1 - \frac{\eta}{2}$ we have $\max_{l\in\{-1,1\}} \mathbb{E}\left[E_{c,l}^{\mathcal{D}}\right] \leq \epsilon_c^{\text{test}} + \epsilon^{\text{sample}}$. Using the definition of the total variation norm

$$\|\mathcal{T} - \mathcal{D}\|_{TV} = \sup_{J\subset D} |\mathcal{T}(J) - \mathcal{D}(J)|,$$

with $J = E_{c,l}$ we can derive $\mathbb{E}\left[E_{c,l}^{\mathcal{T}}\right] \leq \mathbb{E}\left[E_{c,l}^{\mathcal{T}}\right] + \|\mathcal{T} - \mathcal{D}\|_{TV}$ and thus

$$\epsilon_c \leq \epsilon_c^{\text{test}} + \epsilon^{\text{sample}} + \epsilon^{\text{dist}}.$$

We can treat ϵ_s analogously and take a union bound over both the completeness and soundness bounds holding which results in the probability of $1 - \eta$. □

C Numerical Experiments

For the numerical experiments, we implement Arthur, Merlin and Morgana in Python 3.8 using PyTorch (BSD-licensed). We perform our experiments on the UCI Census Income dataset and the MNIST dataset, which is licensed under the Creative *Commons Attribution-Share Alike 3.0* licence. All experiments were executed efficiently, with each run completing in less than 15 minutes.

C.1 UC Irvine Census Income Dataset

In the following, we provide the technical details for the experiments performed on the UCI Census Income dataset, including preprocessing steps and training configurations. We also describe additional experiments that complements our analysis in Section 3.

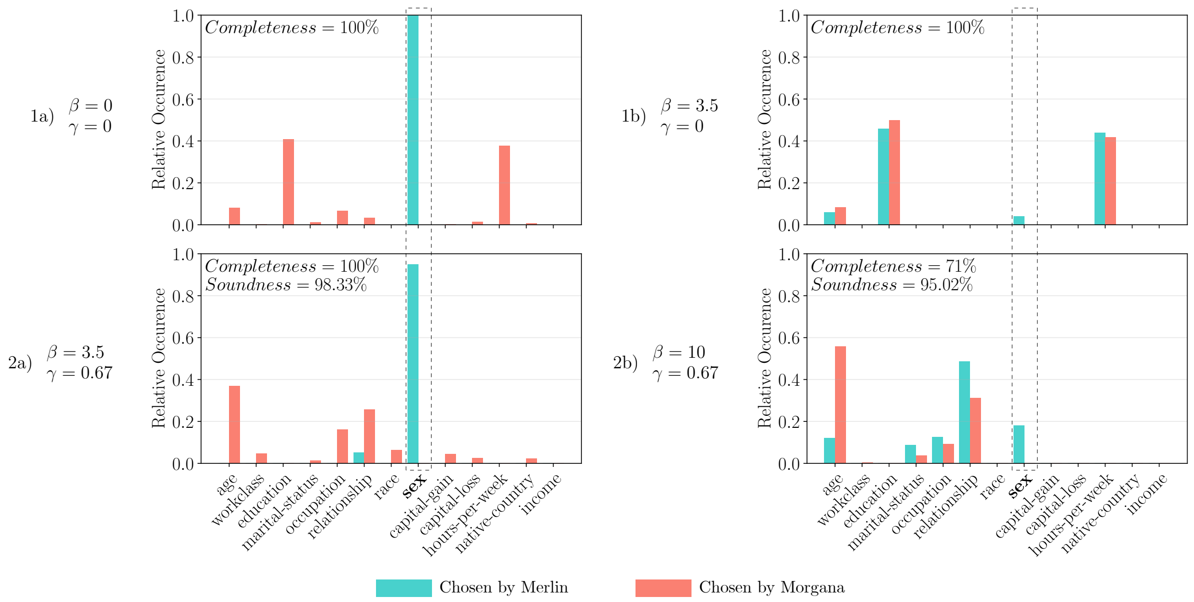


Figure 14: Results from the experiments in Section 3.1, with the features "marital-status" and "relationship" are included. 1) The selected features are the same as in the setup, excluding the two features. 2a) Again, the requirement of soundness results in Merlin selecting "sex" regardless of the punishment. 2b) Despite the higher punishment ($\beta = 10$), Arthur and Merlin achieve completeness of 71%, unlike the results presented in Section 3.1, due to the selection of "relationship", a feature strongly correlated with "sex".

C.1.1 Training Setup

Please note that, when predicting income, a class map $c : D \to \{-1, 1\}$ cannot be defined for all data points, i.e., there are different incomes for exactly the same input. However, only small fraction of data points (0.6%) have this issue.

Data Preprocessing The UCI Census Income dataset consists of both continuous and categorical features, 14 features in total. The target class is chosen to be the feature "sex", which contains the categories "male" and "female", to indicate a possible case of discrimination. The feature "fnlwgt" is removed from the set of features, since it does not contain any meaningful information. In addition, the features "marital-status" and "relationship" are also removed as they strongly indicate the target class. After removal, 11 features remain for each data point. The continuous features are scaled according to the min-max scaling method. To simplify feature masking, all features are padded to a vector of the same fixed dimension. Continuous features are then repeated along the padding dimension, while categorical features are one-hot encoded to the length of the fixed dimension. Note, that the fixed dimension is determined by the categorical feature with the most categories.

Finally, the train and test datasets are balanced with respect to the target class, resulting in 19564 train samples and 9826 test samples.

Model Description Arthur is modelled as a NN with a single hidden linear layer of size 50 followed by a ReLU activation function. The output is converted to a probability distribution via the softmax function with three output dimensions, where the third dimension corresponds to the "Don't know!" option. The resulting NN contains approximately 23k parameters. Merlin and Morgana, on the other hand, are modelled as Frank-Wolfe optimisers, which are discussed in more detail in the overview of the experiments conducted on the MNIST dataset.

Model Selection The training and testing of the models is divided into two phases. First, we pre-train Arthur on the preprocessed UCI Census Income dataset without Merlin and Morgana. Second, we use the pre-trained model to perform further training including Merlin and Morgana. For the pre-training of Arthur, our experiments were conducted such that the models' parameters were saved separately after each epoch. Consequently, for each

Model	Learning Rate	Batch Size	Frank-Wolfe Learning Rate	Epochs
Pre-training Arthur	10^{-4}	512	-	100
Merlin-Arthur Classifier	0.5	512	0.1	20

Table 1: Configurations for pre-training Arthur and training the Merlin-Arthur classifiers on the UCI Census Income dataset.

experiment, we had access to a set of candidate models to choose from for further analysis. To ensure high completeness, we select the pre-trained candidate with the highest accuracy with respect to the test dataset. The selected model then represents the pre-trained classifier for all subsequent experiments involving Merlin and Morgana. Similar to the pre-training process, we also stored the model parameters after each epoch of training with Merlin and Morgana. The results presented in Figure 5 correspond to the models with the highest completeness at a soundness of at least 0.9 among all epochs. The training configurations regarding the pre-training of Arthur and the training of the Merlin-Arthur classifiers can be obtained from Table 1.

C.1.2 Marital-Status and Relationship Included

In Section 3.1, to show the effect most clearly, we kept the focus on a single sensitive feature ("sex") and removed the "relationship" and "marital status" features that strongly correlate with it and are thus informative. We now show the results for the experiments including these features in Figure 14, where Merlin uses these features to communicate "sex" with a completeness of 71%. In the case 2b), where there is an incentive for soundness, but also to hide "sex," Merlin manages to discriminate with 71% accuracy, mostly by relying on the "relationship" feature. As the "relationship" feature is indeed informative about the "sex" of the applicant, this is in line with our theory. In such a case, further investigation is neccessary to decide whether such a feature should be protected.

C.1.3 Other Manipulation Techniques

As mentioned in Section 3.1, we want to further explain the argument why the off-manifold manipulations does not work on the Merlin-Arthur classifier.

LIME Ribeiro et al. [2016] and ShaP Lundberg and Lee [2017] explanations are generated by sampling random inputs around the original data point and compare the classifier output on these points with the original output. The manipulation technique of Slack et al. makes use of the fact that these random samples are off-manifold and change the off-manifold behaviour of the classifier. This allows to get the desired interpretation, while leaving the on-manifold behaviour constant (to continue to discriminate for example).

Changing the off-manifold behaviour of Merlin is without effect for our interpretation, since Merlin will always be used on-manifold. Training Merlin to give off-manifold features to Arthur on which he changes behaviour also does not work. This behaviour is either solely off-manifold, in which case, again, it does not matter. Or it is on-manifold, in which case it is subject to potential exploits by Morgana. In case it cannot be exploited by Morgana, we already proved that the features then need to be informative.

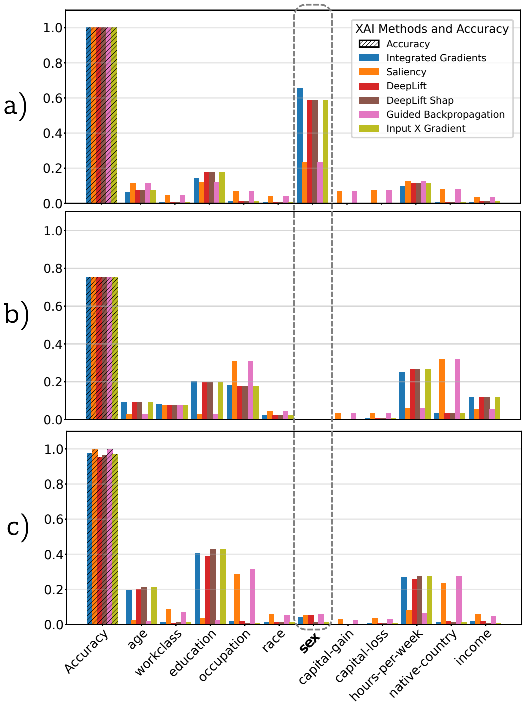


Figure 15: Normalised relevance scores for six XAI methods in the sex discrimination task, generated using the open-source library Captum. a) The unmanipulated network with sex feature as input achieves 100% accuracy in the sex discrimination task, and all methods predominantly highlight the sex feature. b) The sex feature was omitted from the input. The network achieves an accuracy of 75% to predict sex, and the interpretability methods predominantly highlight "education" and "hours-per-week". c) The sex feature is included in the input, but for every interpretability method a new network was trained with the additional objective to minimise the attribution to the sex feature. In each case, the network achieves above 98% accuracy, but the sex feature is not strongly highlighted.

C.1.4 Manipulating XAI Methods through the Network Training Objective

In Section 3.1, we demonstrate how an adversarial setup prevents a typical approach to manipulating explanations that has been demonstrated for numerous XAI methods in Dimanov et al., Heo et al. and Anders et al.. Here, we complement this analysis by showing that for the same setup, the manipulation method proposed in to obscure relevant features works for a wide range of popular post-hoc XAI methods. We used Integrated Gradients, Saliency, DeepLift, DeepShap, Guided Backpropagation, and Input X Gradient from the open-source Captum library [Kokhlikyan et al., 2020].

For this, we trained a regular classifier network Φ of the same architecture as Arthur on the sex discrimination task. The loss L is the binary cross entropy loss for predicting "male" versus "female". We looked at three different scenarios, see Figure 15:

a) The classifier has L as its sole objective and has access to the sex feature. Unsurprisingly, it achieves perfect accuracy. All interpretability methods rate "sex" as the most important feature. "Education" and "hours-per-week", which correlate with "sex", are highlighted as well.

b) The objective remains the same, but the sex feature is not part of the input anymore. The classifier achieves a maximal accuracy of 75%, relying on the "education" and "hours-per-week" features.

c) The training objective for Φ was changed to

$$L'(\Phi, \mathbf{x}) := L(\Phi, \mathbf{x}) + \beta_1 \|\mathbf{r}_{\text{sex}}(\Phi, \mathbf{x})\| - \beta_2 \big\|\mathbf{r}_{\setminus\text{sex}}(\Phi, \mathbf{x})\big\|,$$

where $\mathbf{r}_{\text{sex}}$ and $\mathbf{r}_{\setminus\text{sex}}$ are the relevance scores for the "sex" feature and every other feature respectively. We trained a different network for every XAI method and varied β_1 and β_2 in the range of $[2, 20]$, choosing the best performing.

We observe that the inclusion of the "sex" feature raises the accuracy to predict "male" vs. "female" from 75% to almost 98%. At the same time, our "manipulated" network was able to almost perfectly hide the crucial feature for all of these common XAI approaches. This demonstrates that without an adversarial aspect as in the Merlin-Arthur framework, post-hoc interpretability is not robust against manipulation.

C.2 Training Setup for MNIST

Here, we give a detailed description of our training setup for MNIST and show the error bars of the numerical results presented in the main part of the paper over 10 different training runs.

Structure of Arthur. Arthur is modelled using a neural network. Specifically, we consider a convolutional NN with a ReLU activation function. For the case of two classes, we consider a NN with two convolution layers, whereas for the five class case we consider three convolution layers. The output of the convolution is then passed through two linear layers before being used for the output. Table 2 describes the used architecture in detail.

Structure of Merlin and Morgana. Recall that Merlin and Morgana aim to ideally solve

$$M(\mathbf{x}) = \operatorname*{argmin}_{\mathbf{s} \in B_k^d} L_M(A, \mathbf{x}, \mathbf{s}) \quad \text{and} \quad \widehat{M}(\mathbf{x}) = \operatorname*{argmax}_{\mathbf{s} \in B_k^d} L_{\widehat{M}}(A, \mathbf{x}, \mathbf{s}),$$

respectively, where L_M and $L_{\widehat{M}}$ are the loss functions defined in Section 3, and B_k^d is the space of k-sparse binary vectors. Thus, Merlin and Morgana take an image as input and produce a mask of the same dimension with k one-entries and zero-entries otherwise. We additionally added a regularisation term in the form of $\lambda \|\mathbf{s}\|_1$ to both of the objectives, and set $\lambda = 0.25$. We realise the pair Merlin and Morgana in four different ways, which we explain in the following. All of these approaches return a mask $\mathbf{s} \in [0, 1]^d$ that we then binarise by setting the k largest values to one and the rest to zero.

Frank-Wolfe Optimisers In the first case, Merlin and Morgana are modelled by an optimiser using the Frank-Wolfe algorithm [Jaggi, 2013]. We follow the approach outlined by Macdonald et al. [2022] with the Frank-Wolfe package provided by Pokutta et al. [2020]. The optimiser searches over a convex relaxation of B_k^d, i.e.,

$$\mathcal{B}_k^d = \big\{ \mathbf{v} \in [0, 1]^d \;\big|\; \|\mathbf{v}\|_1 \leq k \big\},$$

Layer Name	Parameters
Conv2D	I=3, O=32, K=3
ReLU	
Conv2D	I=32, O=64, K=3
ReLU	
Conv2D	I=64, O=64, K=3
ReLU	
MaxPool2d	K=2
Linear	I=7744, O=1024
ReLU	
Linear	I=1024, O=128
ReLU	
Linear	I=128, O=1

Table 2: Description of the NN architecture used for feature classifier Arthur in the MNIST experiments.

the k-sparse polytope with radius 1 limited to the positive octant. To optimise the objective we use the solver made available at `https://github.com/ZIB-IOL/StochasticFrankWolfe` with 200 iterations.

U-Net Approach For the second case, we model Merlin and Morgana using NNs, specifically a U-Net that has already been used in the XAI domain to create saliency maps, see [Dabkowski and Gal, 2017]. We copy the U-Net design by Ronneberger et al. [2015] since it achieves good results in image segmentation, a task reasonably similar to ours. We predict mask values between zero and one for each image pixel, and rescale the mask should it lie outside of $\mathcal{B}_k^d$. The binarisation of the mask is ignored during the backpropagation that trains the U-Nets and only employed to produce the masks that Arthur is trained on.

Hybrid Approach In the Hybrid approach, Merlin is modelled by a U-Net, whereas Morgana is still modelled by the FW-optimiser. This approach is useful since for a sound Arthur that cannot be fooled, the training of the Morgana U-Net becomes difficult and the U-Net diverges. It then takes a while of training for the U-Net to adapt, should Arthur open himself to possible adversarial masks. The optimiser is applied to each individual image instead and can find possible weaknesses instantly.

Class-Networks One of the alternatives to Merlin and Morgana that we propose are class-specific U-Nets. Instead of Merlin and Morgana each being represented by a network, there is a U-Net associated with each class that is trained to produce a feature mask that convinces Arthur of its own class for any input image, i.e., for $l \in [C]$ try to solve

$$M_l(\mathbf{x}) = \operatorname*{argmin}_{\mathbf{s}\in B_k^d} -\log(A(\mathbf{s}\cdot\mathbf{x})_l) + \lambda\|\mathbf{s}\|_1.$$

Merlin is then implemented as an algorithm to choose the U-Net corresponding to the true class, so

$$M(\mathbf{x}) = M_{c(\mathbf{x})}(\mathbf{x}).$$

Morgana instead uses for each individual image the output of the U-Net that most convinces Arthur of a wrong class (maximises the Morgana-loss), i.e.,

$$M(\mathbf{x}) = M_l(\mathbf{x}) \quad \text{with} \quad l = \operatorname*{argmax}_{l\in C\setminus\{c(\mathbf{x})\}} L_{\widehat{M}}(A, \mathbf{x}, M_l(\mathbf{x})).$$

Ideally, this training setup would be more stable than the normal U-Net approach. When Arthur cannot be fooled, the class-U-Nets still have a reasonable objective in convincing him of the correct class, which hopefully prevents divergence as for the Morgana U-Net. Experimentally, however, the class-networks proved to be much less stable than the simple U-Net approach, see Figure 17.

We give an overview over the parameters used for the four different approaches in Table 3.

Parameter	Value
Batch Size	128
Baseline Value	0.3
Max FW Iterations	200
FW Momentum	0.9
Regularisation λ	0.25
Max NN Passes	5
Arthur LR	1e-4
Merlin LR	1e-4
Morgana LR	1e-6

Table 3: Training parameters for the Merlin-Arthur classifier on the MNIST dataset.

Merlin-Arthur Classifier Training The overall training procedure proceeds as outlined in Algorithm 1. We initially train Arthur directly on the training data. In the optimiser approach, this pre-trained network is used to search for the optimal masks for Merlin and Morgana. In the U-Net approach, these masks are directly produced by the U-Nets. Arthur is then trained on the masked images over the whole dataset. The U-Nets are then trained on the dataset with a fixed Arthur according to their respective objectives. We cycle through this process for five epochs. The learning rate is 1e-4 for the Arthur and Merlin network and 1e-6 for the Morgana and the class-specific networks.

C.2.1 Purely Cooperative Setup and “Cheating” for MNIST

Here, we discuss the Merlin-Arthur classifier when only Merlin is used with no Morgana. Our results demonstrate that the inclusion of Morgana is necessary for Merlin and Arthur to exchange meaningful features and abstain from “cheating”. For a purely cooperative setup, information about the class $c(\mathbf{x})$ that Arthur infers is dominated by the fact that Merlin chose that feature, rather than the feature itself, i.e., $H(c(\mathbf{x})|M(\mathbf{x}) = \mathbf{z}) \ll H(c(\mathbf{y})|\mathbf{z} \subseteq \mathbf{y})$. We can upper bound $H(c(\mathbf{x})|M(\mathbf{x}) = \mathbf{z})$ through the classification error P_e that Arthur and Merlin achieve via Fano’s inequality [Fano, 1961]:

$$H(c(\mathbf{x})|M(\mathbf{x}) = \mathbf{z}) \leq H_b(P_e) + P_e \log(|C| - 1),$$

where $|C|$ is the number of classes. We can then bound the amount of information that is transferred by the choice of the mask rather than the feature itself as

$$\begin{aligned} I(c(\mathbf{x}); M(\mathbf{x}) = \mathbf{z}) - I(c(\mathbf{y}); \mathbf{z} \subseteq \mathbf{y}) &= H(c(\mathbf{y}) \mid \mathbf{z} \subseteq \mathbf{y}) - H(c(\mathbf{x}) \mid M(\mathbf{x}) = \mathbf{z}) \\ &\geq H(c(\mathbf{y}) \mid \mathbf{z} \subseteq \mathbf{y}) - (H_b(P_e) + P_e \log(|C| - 1)). \end{aligned}$$

We define the *cooperative information* as

$$I_{\text{coop}} := \max\left(0, H(c(\mathbf{y}) \mid \mathbf{z} \subseteq \mathbf{y}) - (H_b(P_e) + P_e \log(|C| - 1))\right),$$

which lower bounds how much Arthur and Merlin “cheat”. We train Merlin and Arthur on the MNIST dataset and show the results in Figure 16 that in the purely cooperative case I_{coop} keeps the classification error low despite exchanging uninformative features. Including Morgana on the other hand pushes I_{coop} to zero even for small masks. In this case Merlin produces highly informative features.

Figure 16 depicts “Cheating” measured by I_{coop} over the MNIST test dataset (restricted to 5 classes) for different setups. In a purely cooperative setup, Arthur and Merlin learn to communicate over uninformative features. Thus, the classification error P_e stays low, while the conditional entropy of the class with respect to the feature goes up. When Morgana is present and soundness is high (in our analysis $\epsilon_s < 0.05$), Arthur and Merlin cannot cheat, the classification error increases for small k.

Why is this observation important? Note that the purely cooperative setup could already be seen as interpretable. With Merlin as a network it appears as a version of a self-interpreting network. With Merlin as an optimiser, it is similar to Rate-Distortion Explanations (RDE) [Macdonald et al., 2019]. In fact, RDE have been criticised in [Wäldchen et al., 2022] for producing “cheating” masks that artificially create features that were not present

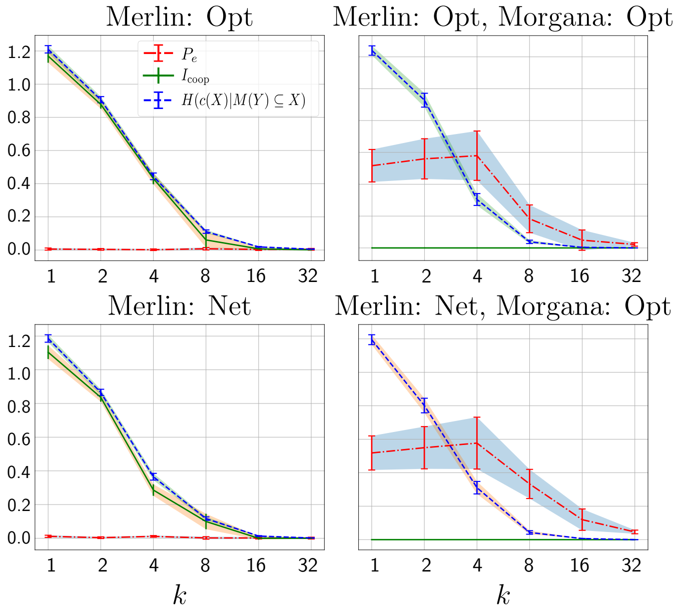


Figure 16: This figure depicts the mean and standard deviation over 10 training runs for the error probability, cooperative information and the class entropy. This was obtained from 5-class classification with classes 1,2,3,4 and 5 with $\gamma = 0.75$ for the purely cooperative setup (*left*) and the adversarial setup (*right*), where Merlin was realised as an optimiser (*top*) or as a neural network (*bottom*).

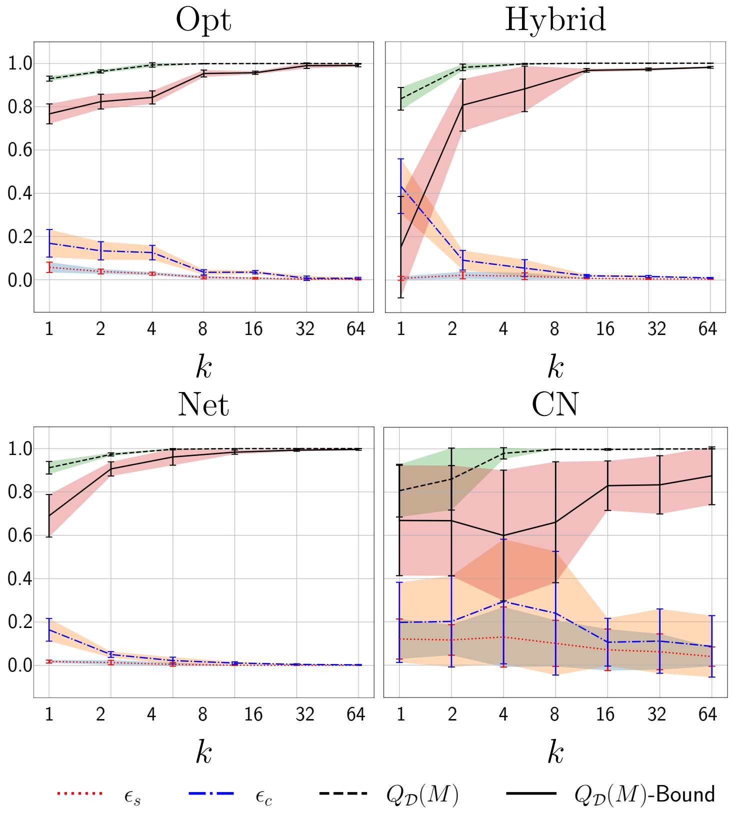


Figure 17: We show the mean and standard deviation over 10 training runs for completeness and soundness, along with the average precision and its bound as obtained from 2-class classification with classes 6 and 9 with $\gamma = 0.75$ for different settings.

in the original image. We connect this to the lack of robustness with respect to Morgana. Designers of self-interpreting setups need to explain how they circumvent the problem of "cheating" masks.

Figure 17 depicts the results from the main paper in more detail. Specifically, it depicts averages and the standard deviation as error bars for each parameter using samples from 10 different training runs. The results presented here are consistent with those presented in the main body of the paper, except for the class networks. The error bars are large for small mask sizes, but shrink as the mask size increases. The class-network approach is considerably less stable than our other implementations, even though it was conceived as a stabler alternative to the simple network approach. One possibility might be that since each U-Net both cooperates with Arthur and wants to fool him, they are more sensitive when Arthur switches between achieving good soundness and completeness. We hope that further research will determine if this realisation can be trained in a stable manner.

C.2.2 Stability of the Three-player Game

Min-max games are indeed challenging to optimise, nevertheless they are common in state-of-the art approaches, e.g., in adversarial robustness or GAN-training [Roth et al., 2017, Wiatrak et al., 2019]. More sophisticated training routines that have already been developed in these areas are a way to stabilise Merlin-Arthur training for more complex datasets.

Figure 17 shows the error bars for soundness and completeness for 10 independent runs with 12 rounds of training, respectively. One can see that the variance for the Opt, Hybrid and Network case is quite small. We find that the more information Merlin is allowed to send to Arthur, the more stable the training. For large features, we come close to the regime studied by Anil et al.. They show that when a strategy is assumed to have perfect completeness and soundness, there is no reason for the agents to change their strategies and the system is in equilibrium. However, in Figure 6, we also investigate the more interesting regime where the mask size is so small that no perfect strategies exist, leading to a trade-off between soundness and completeness. In our experiments, the agents vary along this trade-off during training. We push this trade-off strongly in the direction of soundness ($\gamma > 0.5$ in Arthur's objective), and obtain greater stability.

Another factor is that stability is not a prime concern for our setup. The advantage of our theoretical bound is that, compared to Anil et al., we do not assume that the system is in equilibrium. Even if the agents do not converge to the equilibrium, one can take a well-performing snapshot during training, and our bounds still apply. However, in our experiments this was not necessary, and we always evaluate soundness and completeness after 12 rounds of training.