Interpreting Concept Models For Effective Human-Machine Collaboration

Jack Lawrence Dean Furby

A thesis submitted for the degree of Doctor of Philosophy

Cardiff University

January 2025

Abstract

Deep Neural Networks (DNNs) are often considered black boxes due to their opaque decision-making processes. Concept Bottleneck Models (CBMs) aim to overcome this by predicting human-defined concepts as an intermediate step before predicting task labels and thus enhancing the interpretability of DNNs. In a human-machine setting, greater interpretability enables humans to improve their understanding and build trust in a DNN. However, for interpretability to be meaningful, concept predictions must be grounded in semantically meaningful input features. For example, pixels representing a bone break should contribute to the corresponding concept. Existing literature suggests that CBMs often rely on irrelevant features or encode spurious correlations, leading us to question their interpretations.

This thesis investigates how CBMs represent concepts and how dataset design and model training influence their interpretability. We evaluate the impact of different concept annotation configurations, emphasising the importance of dataset configuration. Using synthetic and real-world datasets, we demonstrate that CBMs can align concepts with semantically meaningful input features when trained appropriately.

We analyse challenges w.r.t. concept correlation and input feature sensitivity, where correlated concepts in training data can lead to concept representations encoding extraneous information and increase concept sensitivity to unrelated input features. To address the challenge of dataset design, we propose best practices for training CBMs that ensure concepts are grounded in semantically meaningful features, minimise leakage and maintain predictable concept accuracy under input feature manipulations.

We conducted the first human studies using CBMs to evaluate human interaction in collaborative task settings. Our findings show that CBMs improve interpretability compared to standard DNNs, leading to increased human-machine alignment. However, this increased alignment did not translate to a significant increase in task accuracy. Understanding the model's decision-making process required multiple interactions, and misalignment between the model's and human decision-making processes could undermine interpretability and model effectiveness in a collaborative setting.

Contents

List of Figures

List of Tables

List of Acronyms

Below is a list of acronyms and their meanings that are used in this thesis. Additional acronyms are listed in their respective chapters.

AI Artificial Intelligence

AUC Area Under the receiver operating characteristic Curve

CBM Concept Bottleneck Model

CLIP Contrastive Language-Image Pretraining

CM Concept Model

CNN Convolutional Neural Network

CUB Caltech-UCSD Birds-200-2011

DL Deep Learning

DNN Deep Neural Network

IG Integrated Gradients

IoU Intersection over Union

LLM Large Language Model

LIME Local Interpretable Model-agnostic Explanations

LR Learning Rate

LRP Layer-wise Relevance Propagation

ML Machine Learning

SCS System Causability Scale

SGD Stochastic Gradient Descent

VGG Visual Geometry Group

XAI eXplainable Artificial Intelligence

List of Publications

The work introduced in this thesis is based on the following publications. All of these are collaborative publications that I lead and provided the material for. At the start of each chapter I detail the related publications.

Published Papers

- Jack Furby, Daniel Cunnington, Dave Braines, and Alun Preece. Towards a deeper understanding of concept bottleneck models through end-to-end explanation. *R2HCAI: The AAAI 2023 Workshop on Representation Learning for Responsible Human-Centric AI*, 2023 (Furby et al., 2023)
  This paper focuses on input feature attribution from concept predictions, and concept attribution from task predictions for CBMs. This included performed qualitative and quantitive metrics, ultimately hypothesising the issues with training CBMs to learn to use semantically meaningful input features was due to the configuration of concepts in the training data. Additionally, the paper proposed that the proportion each concept contributes to task label predictions can be used to explain a model's decision-making process. This thesis covers input feature attribution, concept attribution and proportion of concept contribution in Chapter 3.

- Jack Furby, Daniel Cunnington, Dave Braines, and Alun Preece. Can we constrain concept bottleneck models to learn semantically meaningful input features?, 2024. URL `https://arxiv.org/abs/2402.00912` (Furby et al., 2024)
  This paper evaluates the effect of dataset labelling on the representations

CBMs learn. The paper introduces the synthetic dataset Playing Cards, which proves CBMs can learn to predict concepts using semantically meaningful input features. In addition, this paper analyses CBMs using information leakage metrics to evaluate the information encoded into concept representations. This thesis covers the datasets used in this paper and metrics to evaluate input feature attribution in Chapter 3. In Chapter 4 this thesis covers learned concept representations and information leakage.

- Jack Furby, Daniel Cunnington, Dave Braines, and Alun Preece. The impact of concept explanations and interventions on human-machine collaboration. In *Explainable Artificial Intelligence*. Springer Nature, 2025 (Furby et al., 2025)
  In this paper we conduct the first human studies evaluating CBMs in a human-machine collaborative setting. This paper analyses the original promises and capabilities of CBMs when used by humans. This thesis covers the content of this paper in Chapter 5.

Related Articles

The following publications listed here are relevant to this thesis, but are not considered primary. My contribution to each publication is identified in the summaries below.

- Katie Barrett-Powell, Jack Furby, Liam Hiley, Marc Roig Vilamala, Harrison Taylor, Federico Cerutti, Alun Preece, Tianwei Xing, Luis Garcia, Mani Srivastava, and Dave Braines. An experimentation platform for explainable coalition situational understanding, 2020. URL `https://arxiv.org/abs/2010.14388` (Barrett-Powell et al., 2020)
  My primary contribution to this paper was integrating a DNN with an eXplainable Artificial Intelligence (XAI) technique known as video-audio

discriminative relevance (Taylor et al., 2020) into an interface designed for situational awareness. This XAI approach enabled the separation of individual input contributions to the model's predictions. This XAI technique influenced the selection of XAI techniques used in Chapter 3 and directly led to a method that computed the contribution of concept predictions to the task prediction made by a CBM.

Acknowledgements

First and foremost, I would like to thank my supervisor, Professor Alun Preece, for his insight, guidance and patience throughout my PhD. Without his encouragement and constructive feedback, I would not have been able to complete this thesis. I am also deeply grateful to my industrial supervisor, David Braines, and my collaborator, Daniel Cunnington, both of whose support and regular discussions have played a crucial role in shaping the research in this thesis.

I'd also like to thank my colleagues in the school of Computer Science and the school of Mathematics at Cardiff University. They have been welcoming and provided a social environment throughout my PhD. Additionally, the opportunities to present and discuss my research by the Cardiff SIAM-IMA Student Chapter have contributed to my academic growth. To name a few colleagues, I am grateful to Max Curtis and Paul Goddard for their guidance in running human studies and testing my study platform.

Beyond academia, I am deeply thankful to my family and friends for their constant encouragement and support. A special mention goes to Zack, Douglas, Alice, Max, and Siân for their support which has kept me motivated. In particular, Alice has provided support during some of the toughest moments.

This research is funded by the UK Engineering and Physical Sciences Research Council (EPSRC) and IBM UK via an Industrial CASE (ICASE) award. Doctoral training was supported by the UKRI AIMLAC CDT, funded by grant EP/S023992/1.

Chapter 1

Introduction

Deep Learning (DL) has transformed and will continue to be integrated into many domains such as computer vision and natural language processing, in addition to many industries including healthcare and finance. DL involved the training of Deep Neural Networks (DNNs) that aim to mimic the networks of a human brain with the artificial perceptron introduced in 1958 (Rosenblatt, 1958) and Convolutional Neural Networks (CNNs) (a type of DNN) introduced in 1989 (LeCun et al., 1989). It was not until 2012 with AlexNet (Krizhevsky et al., 2012) when DL and DNNs were shown to be highly accurate in computer vision and hardware was feasible to train DNNs. This can be attributed to their current popularity.

DNNs are often seen as black-box systems as the relationship between neurons are non-linear (Benitez et al., 1997), meaning that the path they use to arrive at an output, such as identifying an object in an image, is opaque to human understanding. This opacity can lead to issues in critical applications, such as healthcare, where ensuring that models learn meaningful features is crucial for accuracy and trust. Trust will be hard to achieve if we cannot explain, and thus understand, what the models are doing (Miller, 2019). Furthermore, regulatory requirements, such as those under the General Data Protection Regulation, legally require interpretability and transparency in Artificial Intelligence (AI) (European Parliament and Council of the European Union, 2016).

Because DNNs are good at pattern matching, they can be used to automate repetitive tasks while humans will complete creative and problem-solving tasks. In a human-machine collaborative setting, this could see improvements in productivity and more accurate decision-making (Kamar, 2016). However, as discussed, enabling a human to understand a black-box model's decision-making process is a challenging task, but without this understand effectiveness of human-machine teams will be affected (Paleja et al., 2024). Assuming we have an DNN assisting a human by giving a second opinion, we need the DNN to be equipped such that the human can trust the DNN, as failure for trust to form will leave the DNN being ignored. Equally, we do not want overtrust, as this may mean the human does not override the DNN when it makes mistakes (Ososky et al., 2013).

Addressing the challenge of interpretability and building trust with DNN-based models, the field of eXplainable Artificial Intelligence (XAI) has emerged with the focus on developing methods and techniques that make DNNs more interpretable and transparent (Adadi and Berrada, 2018). Several approaches have been proposed to enhance the explainability of DNNs. One approach is gradient-based attribution methods. These methods involve calculating the gradient of the output with respect to the input features, thereby identifying which features most influence the model's predictions (Bach et al., 2015). This can be displayed in the form of a saliency map which visually shows the regions of an input a DNN used for an output prediction. An example of a saliency map is shown in Figure 1.1. The goal of XAI is to make AI more transparent. Explanations should also be designed for human consumption by aligning with the explainee's beliefs, and must not be overwhelming (Miller, 2019).

Saliency maps are not the only type of explanation; another approach involves modifying the model architecture itself to improve interpretability. *Concept Bottleneck Models (CBMs)*(Koh et al., 2020) are one such example. CBMs belong to a broader class of *Concept Models (CMs)*, which aims to improve interpretability by structuring predictions around human-understandable components, called concepts. These concepts often correspond to intermediate attributes of the task, effectively splitting the prediction process into sub-tasks. The motivation for this approach is to make model task predictions understandable to humans.

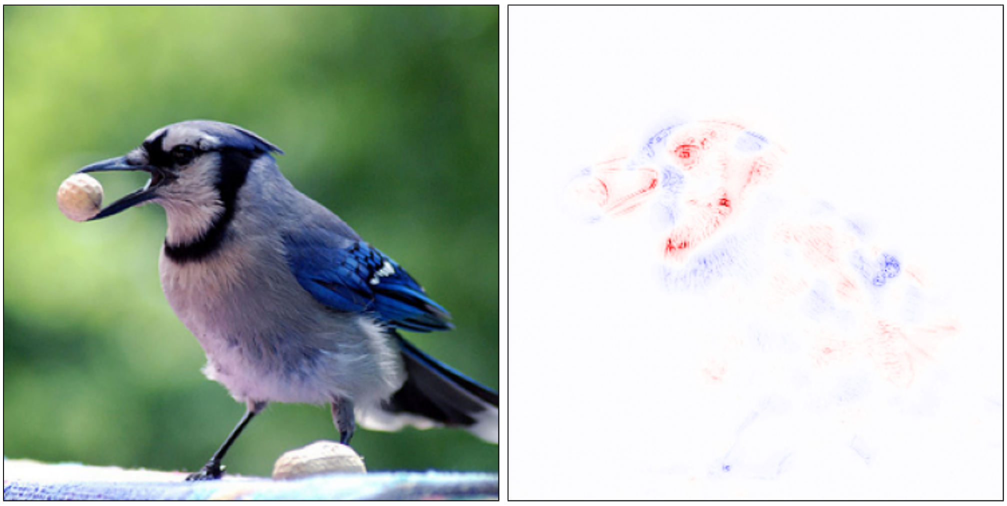


Figure 1.1: Saliency maps visually represents the contribution of input features for a task prediction by a DNN. The colour red is commonly used to represent positive contribution, and the colour blue for negative contribution. In this example input features are pixels from the image on the left, and the saliency map on the right has highlighted the pixels that contributed to the models output.

Recently, in XAI it has been estimated only around 20% of papers consider humans (Nauta et al., 2023). What is needed is additional research in the area of human-machine collaboration that evaluates XAI methods and techniques with a focus on verifying they are beneficial to human-machine collaborative settings. Doing so would allow us to evaluate methods against human behaviour instead of just automated evaluation techniques.

In this thesis, we focus on image classification tasks using CBMs. We examine these models using XAI, both as a method to analyse how these models learn to represent classes and concepts from their training data and as an additional component that enhances their interpretability. Our approach includes automated evaluation metrics and human studies on real-world tasks, providing a comprehensive understanding of CBM's capabilities and their impact on human-machine collaboration.

1.1 Motivation

The primary motivation of this thesis is to explore how DNN-based models and humans can effectively collaborate on shared tasks. In applications where humans and DNNs work together, humans must be able to trust and understand the decision-making process of DNNs. This is where XAI techniques and CBMs, designed to be inherently interpretable, offer an advantage.

CBMs (Koh et al., 2020) have been positioned as improving human-machine collaboration as they are inherently interpretable (Koh et al., 2020). This capability is enabled by the model predicting a vector of human-defined concepts which are then directly used to predict a task label (see Figure 1.2). Concept predictions, known as concept explanations, can be inspected to reveal how a model came to a task prediction more easily. As the task label is predicted solely from a set of predicted concepts, the predicted concept values can be updated by a human operator, known as *intervening.* This can either correct concept predictions and improve the models accuracy, or allow the operator to probe the CBM with various combinations of concepts and inspect the updated predicted task label. This enables the human to ask the model "what-if" questions, e.g., "What if the model instead predicted these concepts?". Interventions are a type of counterfactual explanation (Koh et al., 2020) which in regards to XAI can help to answer why a task prediction was made (Miller, 2019).

However, concept predictions may be misleading to humans interpreting the model's outputs if the model does not predict concepts based on the expected set of input features, but the human assumes it does. For instance, consider a model identifying bird species from images using concepts such as "beak shape" or "wing pattern" (illustrated in Figure 1.2). A human might assume that the model identifies these concepts using the same visual features they would rely on, such as detailed patterns or proportions. In this thesis we use the term *semantically meaningful* to define sets of input features with the same meaning of a concept

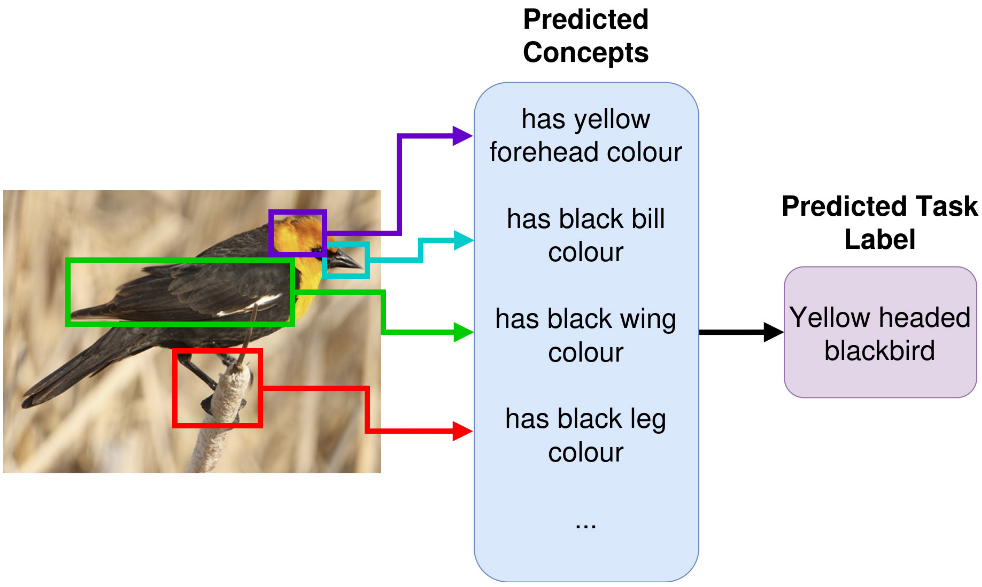


Figure 1.2: CBMs predict concepts based on input features, and task predict labels from the previously predicted concepts. Ideally the set of input features used for concept predictions hold the same meaning as the concept they are predicting.

label (Margeloiu et al., 2021). Alternatively, if the model uses unrelated visual features such as background elements or other bird parts, this misalignment between human and model decision-making could lead to incorrect interpretations of the model's predictions. This issue is not unique to bird identification and may arise in other domains which may have higher stakes, such as medical diagnosis based on X-ray images, where a radiologist might overtrust the model by assuming it uses clinically relevant features to diagnose patients. For humans to make full use of a model they will need to trust the model's output, but a lack of understanding of the causes for a decision may result in a loss of trust (Miller, 2019). To fully realise the interpretability benefits CBMs provide, the ideal case is where concepts are predicted from semantically meaningful input features which, in turn, are aligned with human intuition.

During training, both the concepts and the task labels are supervised with the model split into two parts, a *concept encoder* to map input features to concepts, and a *task predictor* to map concepts to task labels. Splitting the model during training is what enables a human to intervene on the concept predictions at test time, as the predicted concept values can be modified and then passed back through the task predictor (Koh et al., 2020).

Despite the concept vector output, CBMs are unable to explain which input features are used to predict concept (this is known as *feature attribution*), or which concepts contribute to a task label. An XAI study for CBMs (Margeloiu et al., 2021) used saliency maps and suggests that CBMs do not learn concepts as humans would expect (where feature attribution is applied to distinct regions of the input), but instead feature attribution covers the entire input. However, the authors only looked at saliency maps for concepts and not task labels. Additionally, the authors did not provide a hypothesis or argument for what the models have learned to predict concepts, and instead, they attributed their findings to existing feature attribution techniques being “ill-equipped to study attribution for concept bottleneck”. Further, they also limited their study to a single dataset which does not account for all configurations of concept annotation possible in training datasets.

We define *semantically meaningful* as the prediction of concepts based on the minimum set of input features that share the same meaning. For instance, if the concept “has black bill colour” is predicted as present, then the pixels representing the birds bill should be the primary input features used. In contrast, if a CBM predicts the concept “has black bill colour” using input features from the entire body of the bird, its prediction is not *semantically meaningful*, as it includes features from other bird parts unrelated to the bill. This definition is based on the definition with the same name in (Margeloiu et al., 2021).

To explore CBMs interpretability thoroughly, this thesis introduces a new synthetic dataset that allows us to control the configuration of concepts in the dataset and how input features map to concepts, alongside using real-world datasets to verify the findings beyond a synthetic domain. In particular we show how CBMs

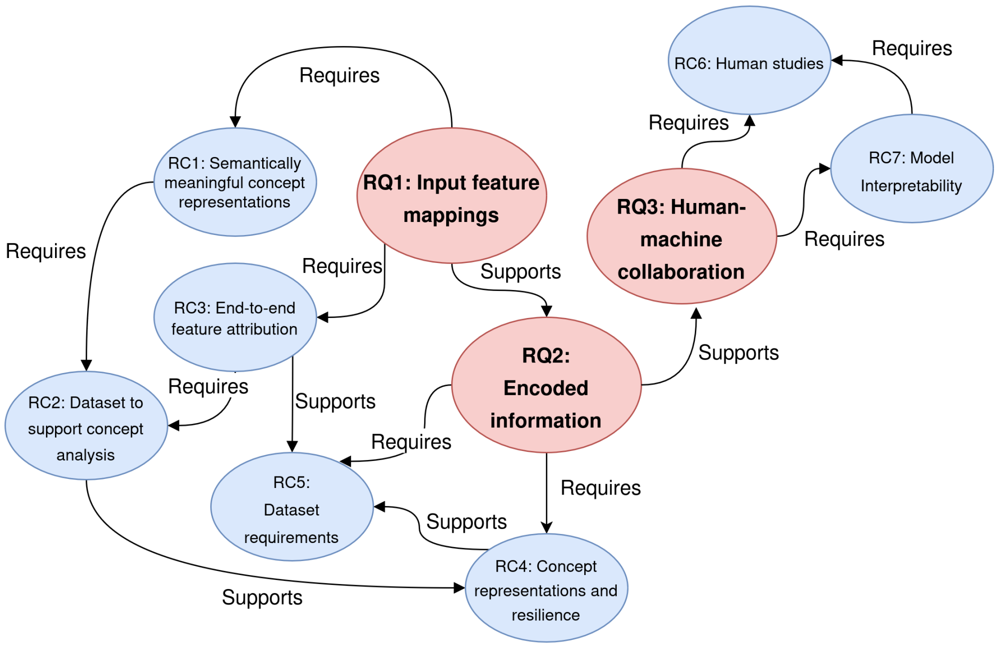

RC6: Human studies
Requires
Requires
RC7: Model Interpretability
Requires
RQ1: Input feature mappings
RC1: Semantically meaningful concept representations
RQ3: Human-machine collaboration
Requires
Requires
Requires
Supports
RC3: End-to-end feature attribution
Supports
Requires
Supports
RQ2: Encoded information
RC2: Dataset to support concept analysis
Requires
RC5: Dataset requirements
Requires
Supports
RC4: Concept representations and resilience
Supports

Figure 1.3: Arrows indicate required contributions to answer research questions or support other contributions, while support arrows represent contributions or questions that feed into linked elements.

can be trained such that concepts are predicted using semantically meaningful input features. We used multiple methods which provided a robust framework for assessing CBMs interpretability in human-machine collaboration, evaluating CBMs with qualitative metrics, quantitative metrics and human studies.

1.2 Research Questions

The research questions below outline the goals of this thesis, focusing on improving the training of CBMs and improving their interpretability for humans. These questions are referred to throughout the thesis using their identifiers (e.g. RQ1). A summary of how these relate to each other and to research questions is illustrated in Figure 1.3.

RQ1: How can we train a CBM to map semantically meaningful input features to concepts, and semantically meaningful concept predictions to task labels?

RQ2: How does the relationship between concepts and input features in the training dataset influence the information encoded in learned concepts and the model's reliance on input features for predicting those concepts?

RQ3: Do Concept Models improve task accuracy and model interpretability in a human-machine setting?

1.3 Contributions

RC1: We perform qualitative and quantitative analysis of CBMs, finding CBMs are capable of learning semantically meaningful concept representations from input features. This contribution partially addresses RQ1 and is covered in Chapter 3.

RC2: We introduce and publish a new synthetic image dataset with fine-grained concept annotations which we use to demonstrate instances when CBMs can learn semantically meaningful concept representations and when they fail to do so. This contribution partially addresses RQ1 and is covered in Chapter 3.

RC3: We expand on existing literature by looking at feature attribution both from the input to the concept vector and from the concept vector to the task output. This contribution partially addresses RQ1 and is covered in Chapter 3.

RC4: We perform an in-depth evaluation of CBMs revealing CBMs can be trained to minimise the encoding of extraneous information in concept representations, and concepts can be resilient to irrelevant input feature alterations. We demonstrate that CBMs generally learn underlying concept correlations present in the training data. This contribution partially addresses RQ2 and is covered in Chapter 4.

RC5: We conclude that two factors are critical for CBMs to learn semantically meaningful input features: (i) accuracy of concept annotations and (ii) high variability in the combinations of concepts co-occurring, that is, each concept in a dataset should appear alongside a variety of others to help the model distinguish between them. This contribution partially addresses RQ2 and is covered in Chapter 4.

RC6: We perform the first human studies using CBMs in a joint human-machine task setting which analyses the interaction between humans and the CBM. We find participants who performed interventions increased trust in a model, but this trust was sometimes misplaced. Additionally, the CBM decision-making process is not aligned to that of the humans. This contribution partially addresses RQ3 and is covered in Chapter 5.

RC7: We show providing concept explanations to humans increases both model interpretability and task accuracy. In addition, interventions can be used to reveal model bias. This upholds the model's promise of increasing interpretability from high-level concepts. This contribution partially addresses RQ3 and is covered in Chapter 5.

1.4 Thesis Structure

Chapter 2 provides an introduction to the background material of research relevant to CMs, XAI, AI, and human-machine collaboration.

Chapter 3 introduces our datasets and how we are using the configuration of dataset annotations to confine models during training. By doing so we demonstrate how the dataset can change the input features our models use for concept predictions. Additionally, we evaluate CBMs decision-making process from predicted concepts to task labels.

Chapter 4 evaluates CBMs with regards to how much extra information is encoded into concepts, concept prediction resilience, and concept correlation. Secondly, we identify required properties for datasets such that CBMs concept prediction is aligned to human expectations and reduces undesired properties.

Chapter 5 introduces two human studies where we asked participants to interact with a CBM to perform a task. The CBM in each of these tasks plays an assistant to the human participant. The first study evaluates the model and explanation abilities with expert users, while the second study evaluates the model and explanation capabilities with lay people.

Chapter 6 summarises the contributions of this thesis and proposes future research directions.

Chapter 2

Background

This thesis investigates how human-machine collaboration can be improved by (1) enhancing the interpretability of DNN-based models for human collaborators and (2) aligning the machine and human decision-making processes. Central to our research is the use of CBMs (Koh et al., 2020), designed to be inherently interpretable.

Throughout this thesis, we use two key terms: CMs and CBMs. CMs refer to a broad class of Machine Learning (ML) where, in addition to predicting the primary task, the model also detects subcomponents (concepts) related to the task. CBMs are a type of CMs and thus refers to a specific model architecture that constrain the model to predict concepts, and use these to predict a final downstream task. This chapter provides a comprehensive overview of key areas, including CMs, XAI, and human-machine collaboration.

2.1 Concept Models

Standard DNNs act in a black-box manner, meaning the decision-making process is opaque. In a human-machine collaboration setting this poses a challenge as the human will find it difficult to understand the machine's decision-making process for task predictions. CMs address this issue by predicting concepts that are connected to the task prediction. For instance, if we had a model that could predict the type of bird in an image, a CM might predict concepts for the birds wing colour and beak shape, all of which will be made available to a human collaborator. This added layer of interpretability allows for increased human

Category		Paper
Concept-grounded models	CBMs and functionally identical frameworks	Koh et al. (2020) Yuksekgonul et al. (2023) Dominici et al. (2024)
	Extended concept representations	Lockhart et al. (2022) Mahinpei et al. (2021)
	Data efficiency	Belém et al. (2021) Wang et al. (2023a) Losch et al. (2019) Chauhan et al. (2023)
	Robustness and extended capabilities	Marconato et al. (2022) Zarlenga et al. (2024) Xu et al. (2024) Kim et al. (2023b) Chen et al. (2020) Alvarez-Melis and Jaakkola (2018)
	Prototype-based models	Wang et al. (2023b) Chen et al. (2019) Wang et al. (2024a) Huang et al. (2024)
	Language model based models	Yang et al. (2023) Wang et al. (2024b) Moayeri et al. (2023) Oikarinen et al. (2023) Rao et al. (2024)

Table 2.1: Summary of CMs.

oversight, critical in high-stakes domains such as medical diagnosis.

Early versions of CMs were introduced by Kumar et al. (2009) and Lampert et al. (2009), although these introduced challenges compared to end-to-end DNNs. Kumar et al. (2009) required each concept to have a large number of positive and negative examples which makes it difficult to create a suitable training dataset. Lampert et al. (2009) achieved a classification accuracy of 40.5% compared to 65.9% on a standard supervised training model in their experiments. The idea of CMs was revisited by Koh et al. (2020) where they separated the concept prediction and downstream task prediction, achieving a competitive accuracy in comparison to end-to-end DNNs. Since the introduction of these models, subsequent designs have been introduced that aim to improve on one or more aspects of the original design.

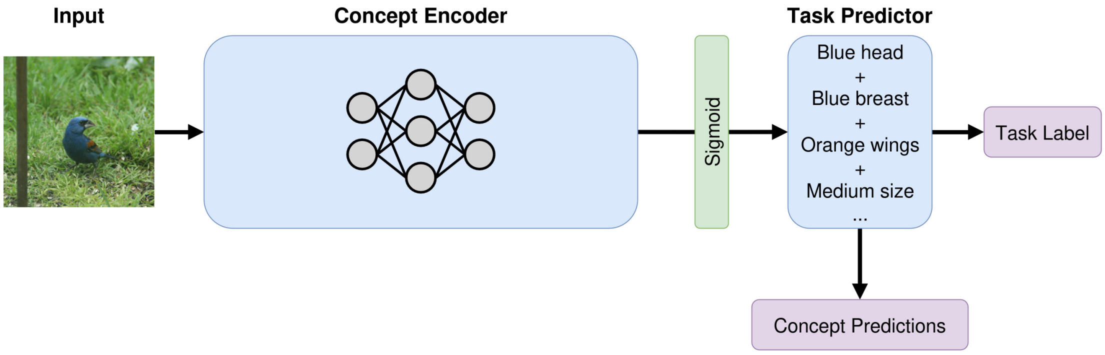


Figure 2.1: Concept Bottleneck Models first predict the presence and absence of a set of concepts that are then used to predict a task label.

We have introduced several CMs from the literature published after these early CMs, which are separated into three categories which we summarise in Table 2.1. Concept-grounded models are models that incorporate concepts as in intermediate part of a models decision making process. Prototype-based models learn concepts as prototypical parts, and language model based models incorporate Large Language Models (LLMs) into the model or training of a model.

Within Concept-grounded models, we have identified four subcategories: CBMs and functionally identical frameworks, extended concept representations, data efficiency, and robustness and extended capabilities. CBMs is a type of CM that is from the CBMs and functionally identical frameworks subcategory as as this type of model learns concepts as an intermediate step to predict task labels.

2.1.1 Concept Bottleneck Models

A CBM (Koh et al., 2020) takes an input which is passed through the *concept encoder* model part, predicting a vector of concepts. Concept predictions are then passed through the *task predictor* model part to predict a downstream task label. Concept predictions are in the range of 0 to 1 where 0 means the model is confident the concept is not present and a prediction of 1 means the model is confident the concept is present. Predictions of 0.5 and above are counted as present. Concept predictions can be viewed by a human in addition to a task label prediction. The vector of concept predictions is referred to as *concept explanations.* A CBM prediction can be intervened by adjusting the concept outputs with new values within the range 0 and 1 and then passing the new set of concepts back through the task predictor to get a new downstream task label prediction. An overview of the CBM architecture is shown in Figure 2.1.

CBMs are trained by supervising both the concepts and the downstream task. Formally, given the training set $\{x^{(i)}, y^{(i)}, c^{(i)}\}_{i=1}^{n}$ where we are provided with a set of inputs $x \in \mathbb{R}^d$, corresponding task labels $y \in \mathcal{Y}$ and vectors of k concepts $c \in \mathbb{R}^k$. A CBM in the form $f(g(x))$ maps the input space to the concept space $g : \mathbb{R}^d \rightarrow \mathbb{R}^k$ and maps concepts to task labels $f : \mathbb{R}^k \rightarrow \mathcal{Y}$. This is such that the task label prediction is made using only the predicted concepts. The function g refers to the prediction of c using the input x and the function f is the prediction of y with the input of c.

During training the loss function $L_{task} : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}$, measures the discrepancy between predicted and true task label, and the loss function $L_{concepts} : \mathbb{R}^k \times \mathbb{R}^k \rightarrow \mathbb{R}$, measures the discrepancy between the predicted and true concepts. The three training methods to train the model parts (the trained models are referred to as $\hat{g}$ and $\hat{f}$) are illustrated in Figure 2.2 and detailed as follows:

The independent training method learns $\hat{f}$ (equation 2.1) and $\hat{g}$ (equation 2.2) separately. During training $\hat{f}$ will take the true value of c as an input and $\hat{g}$ will take the true value of x. At test time the output of $\hat{g}$ will be the input for $\hat{f}$.

$$\hat{f} = arg\ min_f \sum\nolimits_{i=1}^{n} L_{task}(f(c^{(i)}); y^{(i)}) \tag{2.1}$$

$$\hat{g} = arg\ min_g \sum\nolimits_{i=1}^{n} L_{concepts}(g(x^{(i)}); c^{(i)}) \tag{2.2}$$

Sequential bottleneck training follows the same $\hat{g}$ as independent training but $\hat{f}$

(equation 2.3) is learned using the output of $\hat{g}$ instead of the true value of c.

$$\hat{f} = arg\ min_f \sum\nolimits_{i=1}^{n} L_{task}(f(\hat{g}(x^{(i)})); y^{(i)}) \tag{2.3}$$

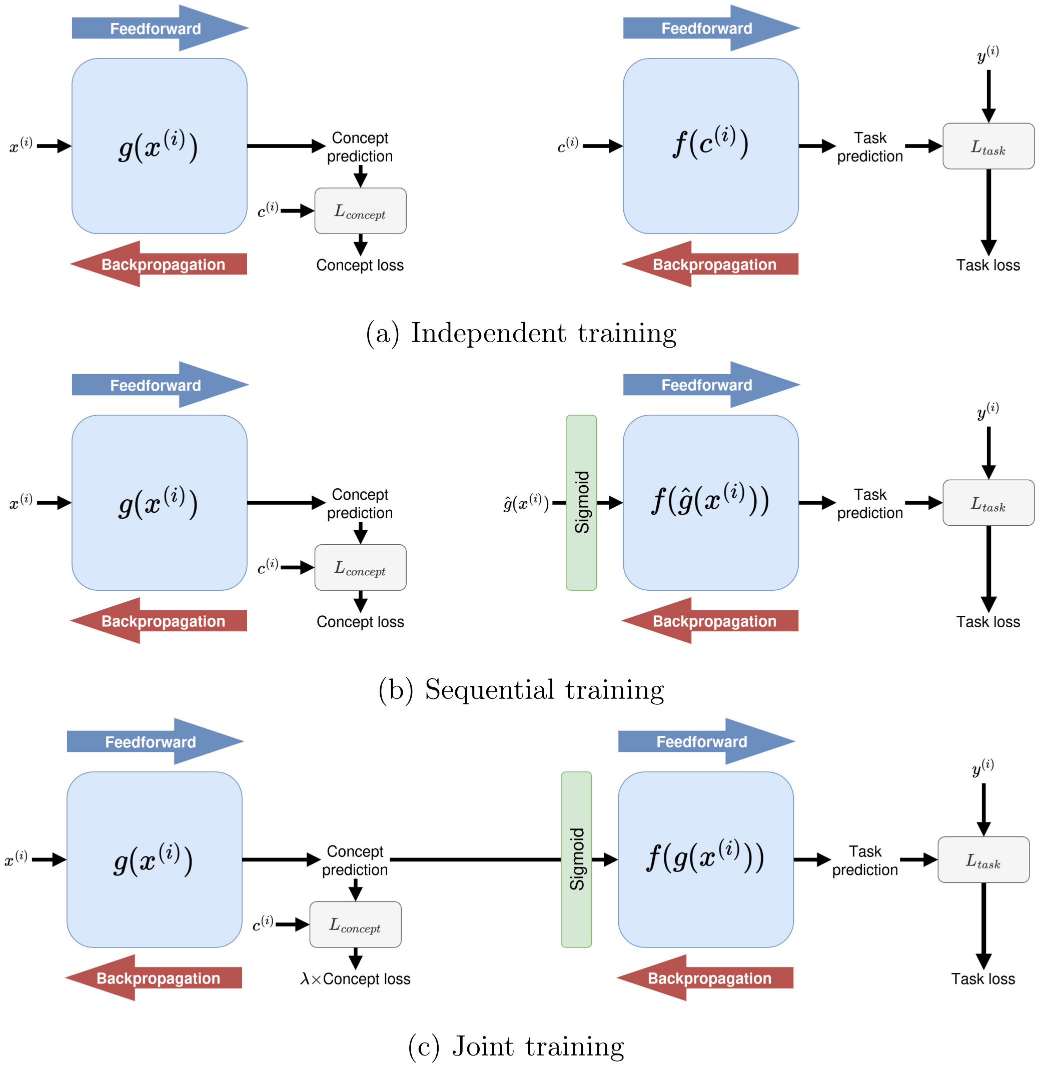
The joint training method minimises the weighted sum of $\hat{f}, \hat{g}$ for some value $\lambda > 0$ as seen in equation 2.4. λ is a hyperparameter used to prioritise the loss L_{task} or $L_{concepts}$. When λ approaches 0, the task prediction loss is prioritised and when λ approaches ∞, concept loss is prioritised.

$$\begin{aligned} \hat{f}, \hat{g} =& arg\ min_{f,g} \sum\nolimits_{i=1}^{n} L_{task}(f(g(x^{(i)})); y^{(i)}) \\ &+ \lambda L_{concepts}(g(x^{(i)}); c^{(i)}) \end{aligned} \tag{2.4}$$

Datasets used to train CBMs can either be configured with *class-level* concept annotations or *instance-level* concept annotations (Koh et al., 2020). Class-level concepts have concept annotations set to the classes, meaning all samples of one class have the same concept values no matter if each sample of a class should have the same concept values or not. Instance-level concepts have concept annotations set to individual samples. This means instance-level concept annotations can account for differences between samples within a class.

For example, let's consider a dataset of flowers where the task label is the flower species. With class-level concepts all roses might have the concept "is red" despite that roses can come in other colours. With instance-level concept the concept "is red" can be applied to only samples of roses of that colour while concepts for other colours of roses can be applied where appropriate. An illustration of this is provided in Figure 2.3.

Class-level concept annotations have the benefit of being cheap to add to a dataset as only one set of concepts is required for each class. Instance-level concepts are challenging to add to a dataset, especially if human annotation is required, or there are a large number of samples to annotate.



(a) Independent training

(b) Sequential training

(c) Joint training

Figure 2.2: CBM training methods. The input to the task predictor f for the sequential and joint training methods is optionally preceded with a sigmoid function to ensure all concepts are in the range 0 to 1.

As CBMs are only trained on task and concept labels they have no ground truth as to what features they should use from input samples and instead are left to discover this. This can lead to the model learning undesired representations of concepts from input features as explored by Margeloiu et al. (2021) who analysed a CBM trained on class-level concept annotations, and Marconato et al. (2022) and Espinosa Zarlenga et al. (2023) who show CBMs can encode more information for each concept than is required to predict themselves. This has the potential

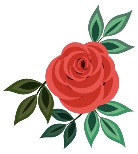
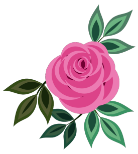
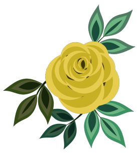
Input	Class-level concepts		Instance-level concepts	
	Concept	Value	Concept	Value
	Is red	Present	Is red	Present
	Is pink	Not present	Is pink	Not present
	Is yellow	Not present	Is yellow	Not present
	Is red	Present	Is red	Not present
	Is pink	Not present	Is pink	Present
	Is yellow	Not present	Is yellow	Not present
	Is red	Present	Is red	Not present
	Is pink	Not present	Is pink	Not present
	Is yellow	Not present	Is yellow	Present

Figure 2.3: Class-level concept annotations cannot account for visual differences between samples of a dataset with the same task label, in this case "rose", unlike instance-level concept annotations.

of allowing concepts to be predicted from one another, or the importance of concepts for task labelling to be unbalanced e.g. overly relying on one or more concepts. Returning to our flowers example this may be seen by the concept "has thorns" being predicted as present only when the concept "is red" is also predicted as present, or the class for rose only being predicted if the concept "is red" is predicted as present.

To the best of our knowledge, little attention has been given to how the configuration of concept annotations in a dataset affects how CBMs learn concept representations. Specifically, prior work does not explore the distinction between instance-level and class-level concept annotations, treating the learned concept representations of CBMs as if they are unaffected by the structure of concept annotations. As instance-level concept annotations provide finer-grained, per-sample attributes, how CBMs learn concept representations will be different to the concept representations learned with class-level concept annotations. Instance-level concept annotations may constrain CBMs to only learn desired concept representations. We discuss CM metrics and evaluation in Section 2.1.5 and revisit this issue in Chapters 3 and 4, where we examine the effects of different concept annotation configurations.

2.1.2 Concept-Grounded Models

Post-hoc CBMs (Yuksekgonul et al., 2023) and AnyCBM (Dominici et al., 2024) create a CBM by adding concept prediction and using the predicted concepts for task prediction in a pre-trained standard DNN. Post-hoc CBMs utilise Concept Activation Vectors (Kim et al., 2018) to learn concept representations. Meanwhile, AnyCBM trains a second model to translate a standard DNN embeddings to a set of supervised concepts and then back to the embeddings. Both approaches enable model interpretability without reducing task accuracy and have the advantage of allowing concept selection independently of the downstream task thus reducing some of the challenges of acquiring a suitable dataset for training. Post-hoc CBMs and AnyCBM keeps the same inherent interpretability and intervention capability as CBMs. However, as the final model architecture is similar to CBMs, they also have the same limitations as CBMs.

Several models add additional components to the standard CBM architecture to handle situations where dataset concept annotations are inadequate to accurately predict a downstream task. Sidecar CBMs (Lockhart et al., 2022) adds a component which can bypass the concept vector when a set criterion is met, meaning concepts are not suitable for task prediction. If this criterion is not met then task predictions are made using concept predictions. Similarly, Hybrid CBMs (Mahinpei et al., 2021) combines the concept vector with unsupervised outputs, ensuring that the model can capture information that does not fit into the supervised concept labels. While these models improve model accuracy in scenarios when concept annotations are not complete, the downside is a potential reduc-

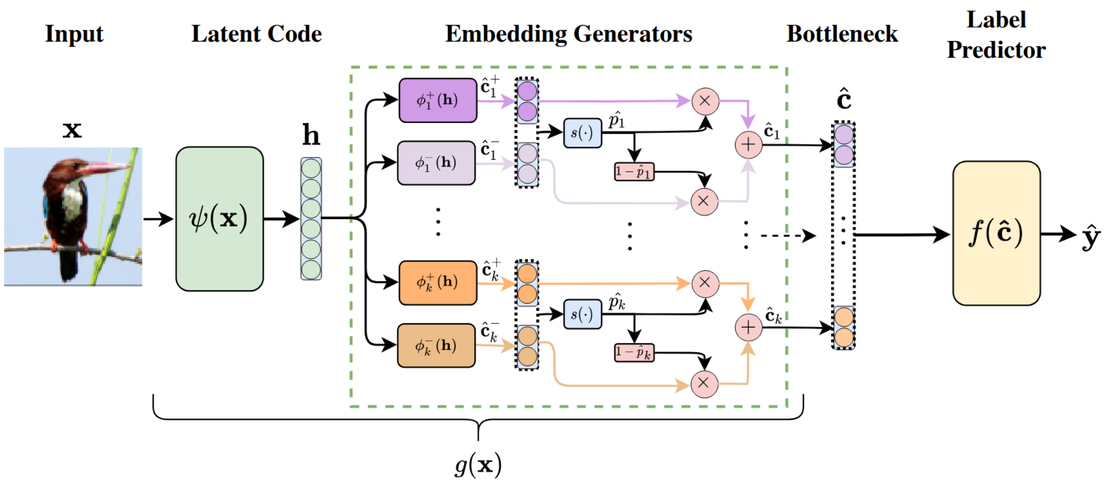


Figure 2.4: Concept Embedding Models learns two embeddings for each concept: one ($\hat{c}_i^+$) for when a concept is present, and another ($\hat{c}_i^-$) for when a concept is absent. Only one embedding is active at a time. A human can intervene on a CEM by changing which embedding is active. This figure is from (Zarlenga et al., 2024).

tion in interpretability since not all model predictions will be explained through concepts and thus will exhibit the same black-box nature CBMs were intending to reduce.

Concept Embedding Models (CEMs) (Zarlenga et al., 2024) also identify rich concept annotations in datasets are hard to create in addition to there often being a trade-off between accuracy, robust explanations and effective intervention. Their proposed model architecture (illustrated in Figure 2.4), CEMs, includes two embeddings for each concept: one for a concept being present, and another against a concept being present. If a human wishes to intervene on a concept prediction they can set the model to use only the concept embedding for the desired concept presence rather than a weighted mix of the two embeddings. CEMs are demonstrated to show similar or better accuracy to CBMs while also showing strong intervention ability and robustness to incorrect concept intervention. This is primarily done by including random interventions during training that update predicted concepts with their ground truth values. Human testing would be required to show if this has a significant difference in a real-world use case.

Chen et al. (2020) proposed a separate approach from CBMs by introducing a *concept whitening* module into a DNN. This module is trained to align predefined concepts in the latent space, arranging them in orthogonal directions. This makes DNNs interpretable by identifying training samples that are most activated along a particular concept's axis in the latent space, without lowering the model's performance on the downstream task. Concept Whitening also can be introduced after a model is trained with little additional training required. Compared to CBMs, Concept Whitening does not have the same dataset requirements as concepts are introduced from a separate data source, but these models do not allow user feedback such as the CBM intervention capability.

Losch et al. (2019) introduced Semantic Bottleneck Networks that are conceptually similar to CBMs, using a bottleneck to represent semantics extracted from input features. Semantic Bottleneck Networks can be created by adding a bottleneck layer to an existing model that already segments input features, thus reducing the complexity that would otherwise be required to create a suitable dataset. Unlike CBMs, Semantic Bottleneck Networks are created such that concepts represent semantic segmentation of input features instead of the presence of concepts. This means their applicability may be limited to domains where semantic segmentation is not the primary concern.

Continuing with tackling the potential challenge of requiring a dataset to have concept annotations for CBM training, weakly supervised multi-task learning (Belém et al., 2021) aims to address the dataset challenge by first training a model on a noisy dataset and then fine-tuning on noise-free samples. The noisy dataset used by Belém et al. (2021) generates concepts based on rules for payment transactions. This approach has shown significant improvements in performance over models trained only on noise-free data, which is to be expected if there are few noise-free samples. However, this method is dependent on the availability of rules or heuristics to create the noisy labels, which may not be applicable in all domains. This is especially restrictive for image-based datasets where such rules without analysing the content of an image may only apply rules at the class-level. Alternatively, generating fine-grained concept labels would require a concept detector similar to the one we aim to train, creating a circular dependency.

Interactive-CBMs (Chauhan et al., 2023) adds a human-in-the-loop element during training, allowing the model to query a human collaborator for concept labels. This approach reduces reliance on fully annotated datasets by introducing new information during training, though it comes with increased training costs and the need for domain expertise which may not be feasible in all domains.

Probabilistic CBMs (Kim et al., 2023b) tackle the problem of ambiguity in concept predictions (e.g. concepts that do not maintain the same visual appearance between samples) by adding uncertainty estimates to each concept prediction. This allows the model to provide uncertainty predictions that can help distinguish between present concept predictions that look similar to training samples, and present concept predictions that are not visually similar (e.g. hidden in the input image). As this architecture is still similar to standard CBMs, they keep the same limitations including concept annotation requirements.

Another approach that removes the need for concept supervision is Concept Bottleneck Learners (Wang et al., 2023a), where a CM is equipped with an extractor that identifies concept prototypes without ground-truth labels. As no concept annotations are required for training, these models have the main advantage of supporting a greater number of datasets compared to standard CBMs. When trained on the Caltech-UCSD Birds-200-2011 (CUB) (Wah et al., 2011) dataset, Concept Bottleneck Learner identified concepts that were consistently detected using visually similar input features across samples in the dataset. However, since the learned concepts are not directly supervised, they may not align with human-understandable features, which would limit their interpretability.

Energy-based CBMs (Xu et al., 2024) enhance the interpretability of CBMs by combining CBMs with energy-based models (Lecun et al., 2006). During training, energy-based CBMs learns concept embeddings (similar to CEMs), a task embedding, and three energy networks: one that measures the compatibility between an input and a task label, one between an input and a set of concepts, and one between a set of concepts and a task label. All energy networks are optimised to assign lower energy to compatible pairs. During inference, all embeddings and energy functions are frozen. The model then predicts concept and task probabilities by minimising the three energy functions. This helps capture inter-concept interaction and provides a richer interpretation of how concepts contribute to predictions than predictions made by a standard CBM. For example, when one concept is intervened with a standard CBM the accuracy of related concept predictions may not change. Energy-based CBMs address this by assigning a low energy value to concept configurations similar to those observed during training and higher energy values to unobserved configurations. However, this approach still suffers from the same dataset dependency as standard CBMs.

GlanceNets (Marconato et al., 2022) tackle the issue of concept representation alignment, namely, alignment is the extent to which models learn to use human-understandable features of data to predict concepts. A model is considered aligned when the features it uses to predict concepts can be clearly and simply mapped to real-world concepts recognisable by humans. These models include a decoder, encoder and classifier in their model architecture. GlanceNets use concept supervision for training, but during test time they can reject samples where concepts do not fit into the learned concept latent space (Sun et al., 2020). This makes them more robust against information leakage which is a measurement of information encoded in each concept above that which is required to accurately predict that concept.

Self-explaining neural networks (Alvarez-Melis and Jaakkola, 2018) satisfy the interpretability criteria of explicitness, faithfulness, and stability. Like CBMs, these models use a concept encoder. However, self-explaining neural networks go a step further by also introducing relevance scores that explain the contribution of each concept to the final prediction. The concepts and relevance scores form an explanation for a given sample input. This makes explanations both explicit and faithful, as these relevance scores are part of the model's prediction mechanism and not generated post hoc. Self-explaining neural networks also satisfy stability through the use of regularisation by adding a penalty to large changes in the relevance scores when small changes are made to the model's input. By providing both the concepts and their relevance scores, Self-explaining neural networks offer richer explanations than CBMs. However, Only CBMs allow for human intervention by adjusting concept values.

2.1.3 Prototype-Based Models

Prototype-based models such as ProtoPNet (Chen et al., 2019), HQ-ProtoPNet (Wang et al., 2023b), and MCPNet (Wang et al., 2024a) take a different approach from CBMs, while also predicting a task label in a two-step process. Instead of predicting the presence and absence of concepts, they learn a set of prototypical parts from their training data. These prototypes are directly comparable to patches from the input, making the decision-making process interpretable. However, as these models do not predict a task label from the predicted presence of prototypes, the ability to intervene on concept predictions is not possible. This means a human collaborator will not be able to ask the "what if" questions that CBMs enable.

A key advantage of prototype models is that they do not require concept labels for training as a set number of prototypes are learned by minimising the latent space between patches from the same class and maximising the latent space between patches of other classes. This makes them compatible with larger datasets where concept annotation might be expensive or infeasible to create. However, ProtoPNet and similar models do not support interventions.

Recently CBMs and Prototype-based models have been combined to create an enhanced CBMs (Huang et al., 2024). This architecture integrates prototypes from prototype-based models and concept predictions from CBMs. This results in a model that does not have the same concept annotation requirements as CBMs while keeping the intervention capability from CBMs.

2.1.4 Language Model Based Models

LLMs, such as GPT-3 (Brown et al., 2020), are a type of DNN based on the transformer architecture (Vaswani et al., 2017). These models utilise self-attention mechanisms to encode relationships across large sequences of data which can be used to generate coherent text outputs. With sufficient training on large datasets, LLMs demonstrate the ability to encode a substantial amount of world knowledge (Jiang et al., 2020).

Recently CBMs have been integrated with LLM and Contrastive Language-Image Pretraining (CLIP) (Radford et al., 2021). Models such as Label-Free CBMs (Oikarinen et al., 2023), Language in a bottle (Yang et al., 2023), Text-to-Concept models (Moayeri et al., 2023), and Align2Concept (Wang et al., 2024b) leverage the ability to generate labels for concept annotations, and thus automate the annotation process of creating training data (Oikarinen et al., 2023). These methods, for the most part, keep the same architecture as CBMs and therefore have the same capabilities as their methods focus on the training data. Label-free CBMs and Language in a bottle go as far as keeping the same CBM training methods.

Text-to-concept is trained differently by using an off-the-shelf DNN as a fixed vision encoder alongside a pre-trained CLIP image encoder. By passing an image dataset through both encoders, we can collect output pairs, which are then used to train a linear alignment layer $h(z)$ that maps the image encoding output from the DNN to the CLIP space. To embed concepts, text prompts (e.g. “a red apple”) are converted into individual concept vectors c_k using a CLIP text encoder. Finally, to build the CBM, each image in a training dataset is passed through the DNN and aligned to the CLIP space. Then, cosine similarities between the aligned image representation and each concept vector are computed as: $s_k(x) = cos(h(f(x)), c_k)$, where $f(x)$ is the DNN image encoder output and h is the linear aligner. The similarity scores $s_k(x)$ are concatenated into a vector, which serves as an input to train a linear classifier task predictor. As this approach can use an off-the-shelf image encoder, the only parts requiring training are the alignment layer and the task predictor.

Finally, Discover-then-Name (Rao et al., 2024) trains a model in three steps. (1) using sparse autoencoders (Bricken et al., 2023), concepts are extracted from a model that has not been trained on pre-specified concepts. (2) These concepts are named. (3) a task predictor is trained on the named concepts.

While these methods reduce the dependency on concept annotations, they introduce new challenges. Primarily, the generated concepts are not guaranteed to align with human intuition, which can reduce the interpretability of the model. Although this approach offers greater scalability, it may prove to not be suitable for environments where guaranteed interpretability is required.

2.1.5 Concept Model Analysis

In the literature CMs have been analysed w.r.t. the input features used for concept predictions and the information encoded into concepts. Measuring the input features used for concept predictions has been previously explored by Margeloiu et al. (2021) where they find their CBM does not predict concepts using semantically meaningful input features. However, they only use a single dataset and thus one dataset configuration. Specifically, they only used a model trained on class-level concepts and do not cover models trained on datasets with instance-level concepts, or other dataset configurations. Figure 2.5 shows one of the results from (Margeloiu et al., 2021) where it is clear the input features used for concept predictions are distributed over the entire bird in the image, and is not contained to just the pixels representing the bird wing.

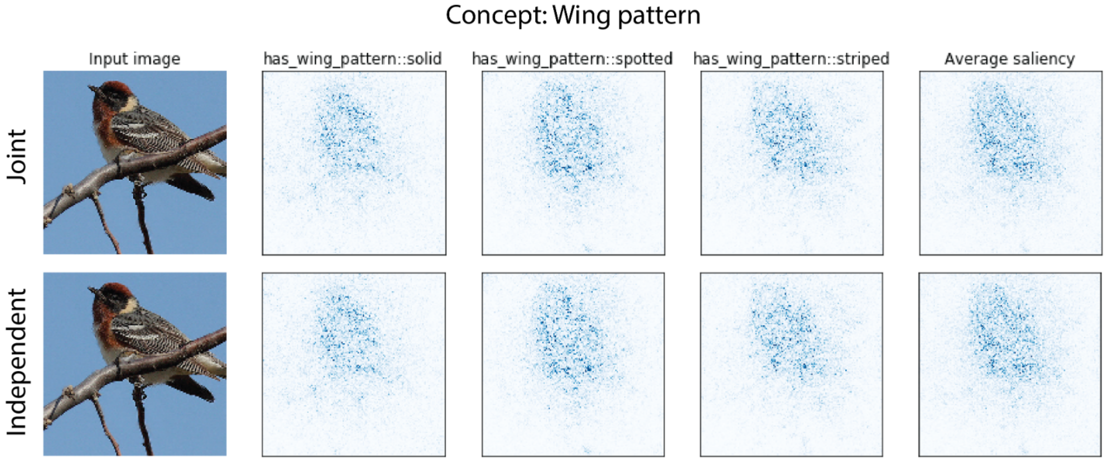


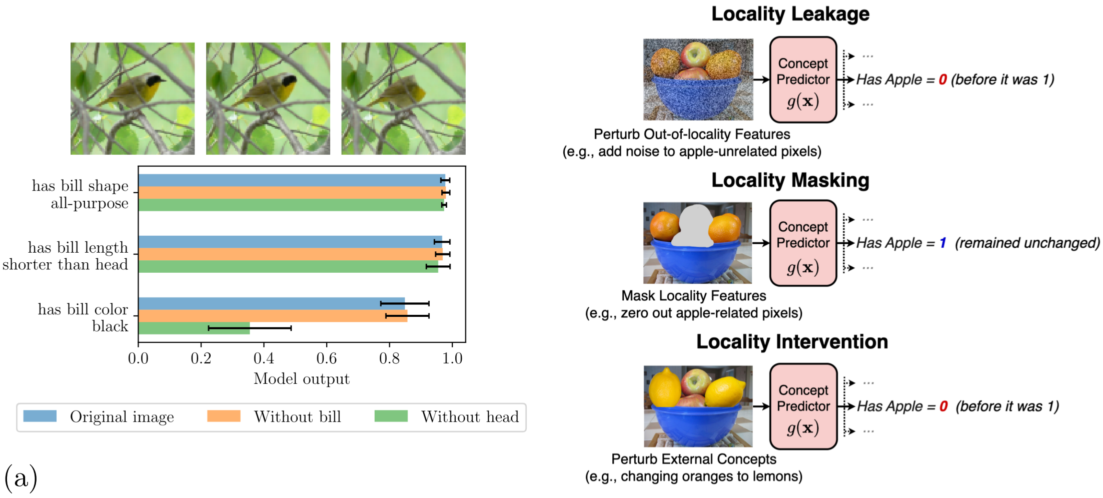
Figure 2.5: Post hoc saliency maps using the IG technique showing a CBM trained on the CUB dataset does not predict the concept representing the wing pattern of the bird using semantically meaningful input features. This figure is from (Margeloiu et al., 2021).

Metrics proposed in the literature to analyse CMs without feature attribution techniques measure either analyse *information leakage* (Mahinpei et al., 2021) or *concept feature sensitivity*. Information leakage evaluates how independent or orthogonal concepts are to one another. For instance, it may be desired for concepts to be learned such that they are accurately predicted without predicting other concepts (Bengio et al., 2013). Concept feature sensitivity, on the other hand, measures how *spatially localised* concepts are (Raman et al., 2024). This is to say whether concept predictions depend on specific input features, semantically meaningful input features, and whether the predictions are robust to the addition or removal of irrelevant input features. However, high concept feature sensitivity does not imply high discriminatory power. A model can be sensitive to the same input features across multiple concepts if those input features are shared. For example, the concept for wing shape and wing colour may depend on the same input features.

Information leakage has been found to occur when CBMs are trained using the independent and sequential methods (Mahinpei et al., 2021), and the joint training method (Margeloiu et al., 2021). Measuring concept leakage can be achieved with a verity of metrics. First of all (Mahinpei et al., 2021; Margeloiu et al., 2021) measured task accuracy after either some or all concepts required for accurate task prediction were removed form the training data. Mahinpei et al. (2021) also introduced *concept purity* which is a measurement of whether concept predictions can be used to predict the labels of other concepts. Concept purity was extended by Espinosa Zarlenga et al. (2023) who introduces the Oracle Impurity Score (OIS) and the Niche Impurity Score. These scores measure inter-concept predictability w.r.t. the expected predictive performance of the dataset. OIS is a measurement of whether a learned concept representation has the predictive power to predict other concepts compared to the expected predictability from ground truth labels. The Niche Impurity Score is the predictive power of multiple concepts. Finally, Marconato et al. (2022) evaluates models according to the metric Disentanglement, Completeness and Informativeness (DCI) (Eastwood and Williams, 2018). Using DCI Marconato et al. (2022) trained models with varying amounts of concept supervision. Marconato et al. (2022) observed less entanglement as concept supervision increased during training.

CBMs have been found to suffer from information leakage with all of these metrics. However, as with feature attribution methods, most of these methods have been evaluated with single dataset configurations, and have not compared models explicitly on the training methods and dataset configuration combined.

Heidemann et al. (2023) introduced the metric *Concept Removal Accuracy (CRA)* to analyse *concept feature sensitivity.* They define this metric as the number of samples for which the model's concept prediction changes from present to not present when the input features for an unrelated concept are removed over the total number of all true positive concept predictions. An example of this is shown


has bill shape all-purpose
has bill length shorter than head
has bill color black
0.0
0.2
0.4
0.6
0.8
1.0
Model output
Original image
Without bill
Without head
Locality Leakage
Concept Predictor $g(\mathbf{x})$
Has Apple = 0 (before it was 1)
Perturb Out-of-locality Features
(e.g., add noise to apple-unrelated pixels)
Locality Masking
Concept Predictor $g(\mathbf{x})$
Has Apple = 1 (remained unchanged)
Mask Locality Features
(e.g., zero out apple-related pixels)
Locality Intervention
Concept Predictor $g(\mathbf{x})$
Has Apple = 0 (before it was 1)
Perturb External Concepts
(e.g., changing oranges to lemons)

(a) acsCRA measures the change in concept predictions as input features are removed. In this example the concept for "has bill color black" changes from a present prediction to a non-present prediction with the removal of semantically meaningful input features. This figure is from (Heidemann et al., 2023)[1].

(b) Illustration of locality leakage, locality masking, and locality intervention. This figure is from (Raman et al., 2024)

.

Figure 2.6: CRA and concept locality examples.

in Figure 2.6a[1] where the removal of input features for the bird beak and head changes the values of predicted concepts. As this metric requires knowledge of ground truth input features for each concept removed from an input Heidemann et al. (2023) also defined the metric "difference in test accuracy" where the accuracy of two concepts are compared between two different subsets of a dataset: one where both concepts are present or absent, and one subset where only one of the concepts are present. Heidemann et al. (2023) found that a high correlation of concept annotations in a dataset may lead a model to use one concept as a proxy to predict others.

Alternatively, Raman et al. (2024) defined similar metrics that measures the ease

[1] Permission has been granted to use this figure from ©2023 IEEE

with which a concept prediction can be modified by changing the input features. They introduce three metrics; locality leakage, locality masking, and locality intervention. Locality leakage captures a score which details how easy it is to change a concept prediction by changing irrelevant input features. Locality masking is very similar to CRA but masks input features that are both semantically meaningful and irrelevant to concepts. Finally, locality intervention aims to understand if the inter-concept correlation of a model's training data is relied upon for concept predictions. Examples of these metrics are shown in Figure 2.6b. Raman et al. (2024) found locality leakage increased with both more layers in a DNN, and as dataset complexity increased. They find locality masking made almost no change with real-world datasets concept predictions and relevant vs irrelevant masking differed by at most 5%. This seems to suggest their models use the whole input to make predictions and do not rely on semantically meaningful input features. Locality intervention showed that when CBMs were trained on an increased number of concept combinations seen during training it improved how robust a model was when making concept predictions. Similar to other metrics, this was only evaluated with one dataset and thus their results may not represent models trained on other datasets.

Huang et al. (2024) evaluates CBMs in regards to the trustworthiness of the models predictions. Trustworthiness is a measurement of whether a model is using the intended input features to make predictions or not. Their metric, called *concept trustworthiness score* works by first predicting which region of an input that a concept is predicted from and then comparing the predicted region to the ground truth region. The metric computes the average number of instances where the ground truth input feature region is inside the predicted region. This has some similarities to The Pointing Game (Zhang et al., 2018), but instead of evaluating whether the predicted input features are within the ground truth region, the opposite is done. Huang et al. (2024) used their concept trustworthiness score to compare various models, including CBMs, using the CUB dataset for which their

CBMs performed poorly.

2.1.6 Concept Model Summary

Overall we have introduced individual model architectures from the different categories of CMs in the literature. A number of these models are either based on or similar to CBMs with similar capabilities. A different class of models, but with some similar goals are prototype-based models although these do not offer interventions on predicted concepts. Finally, we have seen the start of LLM enhanced models and training procedures that show promise for future research.

We have identified most papers tend to introduce architectural improvements that increments on the CBM training methods and model capabilities, but this leaves the training data unchanged. The term “garbage-in-garbage-out” could be used here meaning if we are training CBMs on poor quality training data then the trained models will also have lower performance for the metrics being evaluated. CBMs, to the best of our knowledge, has not been analysed w.r.t. the configuration of training data. In addition, we did not identify any papers that looked at the representations CBMs learn end-to-end. It is currently unknown how CBMs predict task labels from concept predictions.

In this thesis, we focus on the CBM architecture due to its intervention capability which enables counterfactual explanations and thus helps to make a model’s decision-making process interpretable. Prototype-based models and some Concept-grounded models do not have this capability. Concept-grounded models, such as Sidecar CBMs and hybrid CBMs, reduce interpretability by adding additional unsupervised concept outputs. Other approaches, such as CEMs, share a similar architecture and capabilities with CBMs, allowing our research findings to be applicable to them as well. Finally, Post-hoc CBMs and AnyCBM only introduce new methods to create a CBM, and not a change in model architecture.

LLM approaches and enhanced CBMs were published after we had completed a significant amount of the research in this thesis. For instance, Enhanced CBMs was published in March 2024 which was after most of the research in Chapter 3 and Chapter 4 had been concluded. Although we discuss them, we have not actively focused on them in our methods and results.

2.2 Explainable Artificial Intelligence

DNNs are considered black boxes, meaning it is not feasible to know exactly why a particular prediction was made. DNNs are black boxes as the interaction between neurons is non-linear Benitez et al. (1997). DNNs are trained on data with the goal of minimising or maximising a value, e.g. minimising error loss. This means that although the internal representations learned by DNNs have been proven to be highly accurate in many situations, they may contain undesired properties. One example of this is where a model learned to predict huskies based on snow appearing in the background of an image (Ribeiro et al., 2016b). In addition, as DNNs are often trained without humans-in-the-loop, and as such, the representations they learn from their training data may not be compatible with humans Chattopadhyay et al. (2017).

In human-machine collaborative settings with a DNN-based agent, the black box nature of the DNN will lead to challenges for humans to build an accurate mental model of the DNN. A fundamental attribute of a successful human-machine team is the ability of a human to recognise if the DNN will succeed or fail given an input (Bansal et al., 2019). Avoiding the formation of an inaccurate mental model, or to correct incorrectly formed ones, DNN-based agents need to be equipped to make their decision-making process transparent such that humans can understand it (Druce et al., 2021).

We use two primary terms regarding understanding the output from a DNN: *interpretability* and *explainability*. Interpretability is the degree to which a human can determine the cause of a decision Miller (2019); Doshi-Velez and Kim (2017).

Interpretability helps a human build trust in the DNN, and can help move an agent from environment to environment (e.g. from training to a real-world task) Lipton (2018). Explainability is the ability of an artificial agent to reveal underlying causes for output predictions. This is known as eXplainable Artificial Intelligence (XAI) Miller (2019). Interpretability and explainability are often used interchangeably and have a large overlap between their definitions (Molnar, 2022).

Interpretable AI and XAI aim to answer "why" a prediction was made by an AI, and not just what was predicted (Miller, 2019). Analysing AI solutions with metrics such as accuracy (the percentage of correct vs incorrect predictions an AI makes) creates an incomplete picture of a model's performance in real-world tasks. Interpretability and XAI can expand our understanding of the underlying decision boundaries behind a model's predictions, and thus we can verify if a model is suitable for real-world tasks (Doshi-Velez and Kim, 2017). Essentially, interpretability and XAI enable us to look deeper into the underlying causes behind a model's predictions.

2.2.1 Techniques

Just because a model is described as being interpretable or explainable does not guarantee all methods used to make these claims are made to the same standard. Interpretability, for instance, may be evaluated without human input. This means the degree with which a model is interpretable may be an argument by the researchers who claimed it (Doshi-Velez and Kim, 2017), or just that an explanation is "good" Miller (2019), without verification beyond their automated metrics. In fact, automated evaluation of explanation techniques has been shown to not correlate to real human-machine collaborative performance (Nguyen et al., 2021).

In addition, if we are evaluating a DNN we should recognise they are trained to find patterns in their training data which may not align with a human's own beliefs (Geirhos et al., 2019) and thus an explanation for a prediction may not be relevant to the human or go against what they already know (Miller, 2019). This calls into question whether models evaluated without human involvement truly meet the definitions of interpretability and explainability previously outlined in this section.

Explanations from the social sciences have been extensively researched with a focus on humans giving and receiving them. Miller (2019) highlighted several important findings for XAI that were not believed to be a current focus at the time. These being (1) explanations are contrastive as a human will want to know why an event happened over another event, (2) explanations are selected out of possibly an infinite number of causes to just a few (3) probabilities probably don't matter as the most likely explanation may not always be the best for a human, and (4) explanations are social as they are explained relative to the beliefs of the receiver. In the time since this paper was published it has shown to have made a significant impact on XAI and influenced new research (Liao and Varshney, 2022).

We can separate models for interpretability and XAI into two main groups: *intrinsic interpretability* and *post hoc interpretability*. Intrinsically interpretability is a class of models where the structure of the model provides the interpretability capabilities (such as decision trees (Breiman et al., 1984)). Post hoc interpretability models rely on techniques that are applied to a model after training.

Interpretability and explainability techniques can be split into two groups: *model-specific* (such as Layer-wise Relevance Propagation (LRP) (Bach et al., 2015) and Integrated Gradients (IG) (Sundararajan et al., 2017)) and *model-agnostic* (Ribeiro et al., 2016a). Model-specific techniques can only be applied to certain model classes (e.g. can only be applied to DNNs) whereas model-agnostic can be applied to any model no matter the underlying technology. For this reason, model-agnostic techniques are limited to only evaluating models based on the input and output data, and not any internal values or information about the structure of a model, unlike model-specific methods which also have access to a model's internal operations.

Interpretability and explainability techniques can also be split into *global* and *local* techniques. Global techniques are those that describe the behaviours and decision boundaries on an entire model (Lipton, 2018). Local methods are limited by only explaining why a model made a single output. We cannot generalise a single local method output to an entire model (Arrieta et al., 2019; Lipton, 2018).

As this thesis focuses on DNNs, we pay particular attention to model-specific methods and local explanation techniques. Current XAI techniques for DNN-based agents have provided ways of examining models in modalities including text, images, and explanation-by-example (Lipton, 2018).

One such XAI technique we used through this thesis to explain a model's output is with *feature attribution* where a value is applied to each input feature to indicate its contribution to the models task label prediction. With images these can be visualised using *saliency maps* (we provided an example of a saliency map in Figure 1.1). model-specific feature attribution techniques include LRP Bach et al. (2015) and IG Sundararajan et al. (2017) which uses the gradient of a DNN to produce explanations. LRP in particular only redistributes feature attribution from an output prediction to input features such that feature attribution is not created or reduced. This allows the calculation of how much each input feature contributed to the output label. This was demonstrated in (Taylor et al., 2020).

As feature attribution techniques are visualised with saliency maps, expanding beyond this qualitative metric, we can combine feature attribution techniques with other metrics to reveal additional insights. This includes Intersection over Union (IoU) which was used by Saporta et al. (2022), and The Pointing Game (Zhang et al., 2018). IoU evaluates how many feature attribution values are within a defined region of an input, while The Pointing Game measures the number of times the highest feature attribution value is within a defined region of an input.

Despite CBMs being a type of DNN, they include intrinsic interpretability capabilities. Specifically CBMs first predict a vector of human-understandable concepts, which is then used to predict downstream task labels. This architectural design enables the models to be interpretable. Returning to the desired XAI focuses highlighted by Miller (2019), we believe CBMs align with point 1 (explanations are contrastive) and point 4 (explanations are social). Interventions provide counterfactual explanations (Koh et al., 2020) which can be used to reveal why one task label was predicted over another, and the concept vector confines the model decision-making process to use the presence and absence of concepts. This is argued as being easier for a human to understand as it's inline with their beliefs (Koh et al., 2020).

2.2.2 Trust

Jacovi et al. (2021) defines human-machine trust as the perception that a machine is trustworthy for a task, combined with a human being vulnerable to the machine's actions. Trust is not binary but exists on a scale (Jacovi et al., 2021). For instance, if a human consistently believes that an AI can complete a task to the human's expectations, the human has placed high trust in the AI. On the other hand, if the human is not confident in the AI's ability to complete tasks to the human's satisfaction, trust is low. Trust can also be given for specific situations. If an AI is perceived to be capable of completing a task for certain input conditions but not others, trust will be high when the input conditions match (Jacovi et al., 2021).

Building appropriate levels of trust requires time and interaction between a human and AI to enable humans to understand an AI's accuracy and decision boundaries. This is achieved by equipping an AI with interpretability capabilities (Tomsett et al., 2020). Trust is generally increased when an AI output is explicitly explained (Miller, 2019). Even in cases where an AI is not consistently accurate, transparency can help humans develop an appropriate level of trust (Jacovi et al., 2021).

However, we should be careful not to purely maximise trust as it should be given for the right reasons and not misplaced. Misplaced trust can eventually be lost, which is challenging to regain (Ososky et al., 2013). Equally, overly high trust may lead to users relying on AI systems even when the systems are unable to complete tasks accurately (Jacovi et al., 2021). Insufficient trust can result in underutilisation of an AI.

AIs with XAI capabilities can have varying levels of soundness and completeness (Kulesza et al., 2013). A sound AI is accurate in completing its trained tasks, while a complete AI reveals all underlying causes for its actions. Getting the right balance between soundness and completeness is important for effective human-machine interaction. Kulesza et al. (2013) demonstrated that sound and complete models are ideal for building accurate mental models, while complete but unsound models can result in accurate mental models but with reduced trust. Finally, sound but incomplete models often lead to increased requests for clarification on the AI's actions. Additionally, increasing completeness can overwhelm humans, highlighting the importance of designing AIs with outputs with the correct level of detail such that they remain comprehensible and keep humans engaged (Kulesza et al., 2015).

To increase the interpretability of an AI agent, therefore helping to build, trust can be achieved by designing them to behave in a human-like way (Fel et al., 2022). One approach to achieve this could be through the use a CM. Concept explanations and interventions, such as those used by CBMs, improve completeness in comparison to a standard DNN as they reveal the model's decision-making process (Koh et al., 2020).

2.3 Human-machine Collaboration

In this section, we focus on humans and machines working together in a collaborative setting. We start with mental models which can be considered a social capability (Miller, 2019) as it allows a human to gain an accurate understanding of an AI agent. We also look at humans-in-the-loop and existing human studies in the literature. Analysing human-machine collaboration with real humans is important as AI-AI teaming does not guarantee the same performance will translate over to human-machine teams (Chattopadhyay et al., 2017).

2.3.1 Mental Modelling

The main factor to consider in human-machine collaboration is enabling human agents to build a *mental model* of AI agents. A mental model is a cognitive representation of an object's internal mechanics. This allows the human to make predictions about the object's future states and thus aid in future interactions (Johnson-Laird, 1986; Craik, 1943; Halasz and Moran, 1983; Norman, 1983). Craik (1943) was the first to introduce the idea of internal models with Johnson-Laird (1986) coining the name. Rouse and Morris (1986) later defined a mental model as "mechanisms whereby humans generate descriptions of systems purpose and form, explanations of a system functioning and observed system states, and predictions of future system states" which encapsulates many different definitions. We can use this definition of mental models for how humans perceive DNN-based models.

As humans build mental models of objects they interact with, including DNN-based agents, if a human is unable to build an accurate representation of a DNN decision boundaries, the human could be misled to either accept misclassifications or disregard its output entirely (Bansal et al., 2019). To improve the accuracy of a mental model of a DNN-based agent a human will need the agent to be capable of explanations Akula et al. (2019); Miller (2019). This means the DNN agent should be designed to enable both agents to complement each other (Bansal et al., 2019) instead of just maximising the accuracy of the DNN agent. It has been shown that human-machine performance can be lower than the DNN agent on its own, suggesting humans may not trust the DNN agent, but introducing XAI methods can improve team performance over just showing predicted labels (Lai and Tan, 2019).

In this thesis, we primarily focus on human-machine collaborative settings where the DNN agent is advising the human on what action should be performed, but it is up to the human to make the final decision. Using an AI agent in an advisory role, Bansal et al. (2019) evaluated human mental models of some AI agents. They found that over time humans learned the error boundaries of the AI agents, although explanations should not be overly complex as this can make evaluating the AI just as much work as completing the task without the AI's input.

2.3.2 Human-in-the-Loop

Although advances with DNNs in recent years have led to greatly improved accuracy without the addition of humans (Brown et al., 2020; Silver et al., 2016; Redmon et al., 2016), certain domains, such as healthcare, have higher stakes and, as such, complete automation is undesired in case the DNN-based agent makes a mistake. Introducing a human-in-the-loop provides the best of both worlds, allowing the machine to enhance a human and has been shown to improve the performance of DNN and human agents working individually (Reverberi et al., 2022), although many studies show otherwise which may be attributed to worse team cognition or a lack of trust (Schmutz et al., 2024).

Adding a human-in-the-loop has been evaluated during training (Chauhan et al., 2023) and post-training. During training adding a human-in-the-loop can increase the quality of training data by allowing the model to query the human about its current ability (Russakovsky et al., 2015). For post-training human-in-the-loop, we are referring to a human and artificial agent working on a shared goal. A major challenge however is ensuring efficient communication between the team members (Miller, 2019; Schmutz et al., 2024). We can also not assume a well-performing DNN agent is suitable for human collaboration as it has been shown that an AI-AI team can outperform a human-machine team (Chattopadhyay et al., 2017; Schmutz et al., 2024). A simple method to improve the human-machine performance is to train the human on the DNN before starting a task (Chandrasekaran et al., 2017).

As already discussed in Section 2.2.1, evaluating DNN performance with automated metrics does not always correlate to higher performance with a human-in-the-loop. The only suitable option to verify any automated metrics is with a human-in-the-loop (Yadav et al., 2019). Some metrics include the Visual Turing Test (Geman et al., 2015) for visual AI and System Causability Scale (SCS) (Holzinger et al., 2020) for systems with XAI capabilities.

Doshi-Velez and Kim (2017) proposed a taxonomy for the evaluation of interpretability. This starts from functionally-grounded, but automated metrics to application-grounded evaluations with real humans in real-world applications. This taxonomy provides the groundwork to methodically evaluate an AI on automated metrics before verifying interpretability holds with real humans in a real-world setting.

2.3.3 Human Studies

We have summarised the literature of human studies evaluating artificial agents and XAI. XAI is evaluated in regards to trust, model understanding and Human-machine team performance.

To robustly analyse the use of AI agents and XAI a taxonomy was proposed by (Doshi-Velez and Kim, 2017) where they identified studies can be carried out in three categories: application-grounded studies, human-grounded studies, and functionally-grounded studies. Application-grounded studies are studies evaluating humans in a real-world task (e.g. medical diagnosis). Human-grounded studies are studies that use real humans in a simplified task (e.g. multi-choice questionnaires). Finally, functionally grounded studies are any study using automated metrics. This taxonomy acknowledges finding participants is more challenging than automated metrics, but you cannot fully understand how an AI will be used in a real application without using real humans.

Trust of an AI agent can be measured as self-reported trust or observed trust (Papenmeier et al., 2019). Both measurements can be used in the same study. Self-reported trust can be collected with questionaries, whereas observed trust may be quantified by measuring human and model agreement (Rong et al., 2024; Wang and Yin, 2021; Lai and Tan, 2019). However, this metric does not account for if that trust is deserved. To measure this the model's accuracy should also be considered (Wang and Yin, 2021). Trust in existing studies has been shown to be largely dependent on the accuracy of the model (Yin et al., 2019), and overall higher if the model is perceived to be more accurate than a human user.

Model understanding is the degree to which a human creates an accurate representation of an AI agent decision boundaries. As previously mentioned this is commonly discussed in regards to a human building a mental mental of the AI, and XAI aids the creation of this. We may measure model understanding subjectively or objectively (Cheng et al., 2019). Objective metrics require a suitable task or subtask for humans to solve e.g. predicting a model output (Wang et al., 2023a; Doshi-Velez and Kim, 2017), although this is not the only suitable metric to evaluate model understanding (Rong et al., 2024). Subjective analysis can be achieved with a questionnaire or otherwise by asking humans how well they understand the model.

The overall aim of human-machine collaboration is to improve performance, efficiency, or some other metric important for a given task. As we are evaluating a human-machine team where the DNN-based agent is an assistant to the human, we will primarily use the accuracy of the human as they are the final decision-maker. As accuracy is only objective, measuring it is achieved by evaluating the desired metric for a task e.g. accuracy in a medical diagnosis task (Lai and Tan, 2019).

Some human studies has shown XAI to be beneficial to find model bias (Ribeiro et al., 2016b; Adebayo et al., 2020), while others have found little benefit to users (Kaur et al., 2020; Chandrasekaran et al., 2017). It seems that the use of current XAI techniques individually does not fully facilitate their designed intentions.

Studies evaluating saliency maps have produced mixed findings. While some work shows no benefit to including saliency maps, other studies suggest otherwise. For example, Alqaraawi et al. (2020) found that saliency maps generated using the LRP technique helped participants learn which image features their model was sensitive to, enabling them to better predict the model's output. However, Nguyen et al. (2021) highlights that saliency maps can sometimes harm human-machine collaboration, particularly for tasks requiring specialist knowledge. Their study, along with (Jeyakumar et al., 2020), which involved a larger participant count, and (Cai et al., 2019) found explanation-by-example to be a more effective technique for developing model understanding. Despite the mixed evidence, XAI techniques appear to improve human-machine teaming and mostly do not harm performance.

Human studies using CBMs and similar model architectures can be placed into a few categories; human concept preference (Barker et al., 2023; Ramaswamy et al., 2023), concept explanations (Jeyakumar et al., 2023, 2022; Wang et al., 2023a; Sixt et al., 2022; Dubey et al., 2022), human-in-the-loop (Mysore et al., 2023; Nguyen et al., 2024) and bias discovery (Yuksekgonul et al., 2023; Midavaine et al., 2024).

In studies on concept preference, Barker et al. (2023) investigated the concepts humans identified in sample images, finding that human-selected concepts varied widely and performed worse when used by a CM on downstream tasks compared to those chosen by the model. Similarly, Ramaswamy et al. (2023) found that participants preferred smaller sets of concepts. They identified participants preferred using 32 or fewer concepts. This is consistent with completeness (Kulesza et al., 2013), as discussed earlier in this chapter, where the model should not overwhelm a human.

Next, concept explanations preference is a common theme for CM human studies (Jeyakumar et al., 2023, 2022; Sixt et al., 2022; Dubey et al., 2022). Jeyakumar et al. (2022) demonstrated that participants favoured concept-based explanations for a model trained on activity recognition. Participants were asked to select the explanation they preferred from multiple options. However, Sixt et al. (2022) reported that concept explanations performed poorly for bias discovery, although their model was not a CBM. Similarly, Dubey et al. (2022) found that concept explanations underperformed by approximately 5% compared to their proposed method when participants were asked to predict the model's downstream task. Additionally, Jeyakumar et al. (2023) observed that the CBM explanations were the least preferred among participants in a study involving time-series data, though this result may not generalise to other modalities.

Beyond concept preference, some studies have investigated how humans interact with CM in a collaborative task. Mysore et al. (2023) introduced a CBM inspired recommender system that combined user provided concepts with automatically generated ones to suggest relevant text documents. Their study included interventions, leading to improvements of 20–47% in recommendation accuracy compared to distance based approach. Nguyen et al. (2024) examined the effectiveness of explanations in a visual correspondence model, CHM-Coor (Taesiri et al., 2022). Participants interacted with static (could not adjust the models prediction) or dynamic explanations (participant could select parts of the input image for the model to focus on), finding little difference in performance (73.57% compared to 72.68% accuracy). Additionally, participants often agreed with the model's predictions regardless of if the model was correct or incorrect. Both of these studies look at humans updating a models prediction, similar to interventions with CBMs. As Mysore et al. (2023) model has more similarities to CBMs, their findings suggest similar could be observed in an image modality.

For bias discovery, Yuksekgonul et al. (2023) used CBM-like architectures to study human-guided pruning on a model where input samples had shifted (e.g. the correlation of concepts co-occurring is changed after training). Participants selected concepts to prune based on input samples and model predictions, outperforming random pruning and only slightly less effective than fine-tuning or greedy performance. This study was repeated by Midavaine et al. (2024) which found similar results. Considering user pruning does not require access to the training data, this technique shows its potential as a human-in-the-loop approach for addressing data shifts and biases in similar settings.

From these studies we have identified no papers which look at CBMs or similar model architectures that evaluate the capabilities of CBMs in a real-world tasks. Most importantly it has not been shown how participants intervene on concept predictions and whether these models are more interpretable than standard DNNs. In (Koh et al., 2020) they show the effectiveness of interventions and how concepts and interventions can be used for counterfactual explanations, but both of these points are yet to be validated in a human study. We expand on CM human studies in Chapter 5.

2.4 Gap Analysis

In this section, we analyse the existing literature and highlight several gaps that we discovered and will address in this thesis. While the interpretability of CBMs have been extensively discussed, how their learned concept representations are influenced by the configuration of concepts in their training data remains poorly understood. Furthermore, although CBMs are considered inherently interpretable due to their ability to reveal the model's decision-making process through concept explanations and counterfactual explanations, these claims have not been validated with human users. This lack of validation represents a significant gap in the literature that requires further investigation.

We identify that in a human-machine collaborative setting a DNN need to communicate their decision-making process in a human-compatible way, and as discussed, CBMs are positioned to achieving this. However, we have also discussed how existing literature analysing CBM's feature attribution has only been completed with a limited set of training dataset configurations. For instance Margeloiu et al. (2021) only analyses the feature attribution of a CBM trained on class-level concepts. To the best of our knowledge there is no prior work which looks at feature attribution of input features that contribute to a model's output(s) with CBMs trained on other datasets, such as ones with instance-level concepts *[Gap 1]*. To analyse this gap a dataset with with multiple configurations of concept annotations, and ground truth segmentations of the corresponding input features is required *[Gap 2]*.

In addition, there are no papers that analyse which concept predictions CBMs use to predict task labels *[Gap 3]*. As the interpretability of CBMs comes from the addition of concept outputs and interventions, it is highly desired to understand how concepts are used for task label predictions.

We address both input feature attribution and concept feature attribution in Chapter 3, verifying our results with both qualitative and quantitive metrics. We introduce a new dataset better suited to CBM training and evaluation, and include an additional real-world image dataset which uses instance-level concept annotations in our analysis.

As with feature attribution of input features from concepts predictions, information leakage metrics are often used to compare CBMs to other model architectures. We have identified the need to compare CBMs trained on different dataset configurations using this class of metric to get an overall picture of how CBMs learn to represent concepts and the information they encode *[Gap 4]*. We complete this in Chapter 4. This has allowed us to conclude how datasets for CBM training should be configured to achieve concept prediction from semantically meaningful input predictions, minimise the encoding of extraneous information in concept representations, and show concept predictions can be resilient to irrelevant input feature alterations *[Gap 5]*.

Finally, We have looked at both human studies with CMs, and with XAI. This has shown XAI does not always translate to large improvements in model interpretability. Regarding CMs, Previous studies cover a range of model capabilities and human preferences. However, we are left with a few questions. Firstly, Barker et al. (2023) shows CBMs may use a different subset of concepts for a downstream task prediction than a human completing the same task. This raises the question of whether this occurs with our models and if it is important for models and humans to predict tasks with the same concepts. The main concern is interventions may not aid a human if the concepts the CM uses compared to a human are significantly different. Next, most studies looked at concepts without a task. The only exceptions to this was (Mysore et al., 2023) and (Nguyen et al., 2024). Currently, there are no studies that explore human interaction with CMs in a real-world task *[Gap 6]*. How humans interact with these models remains unknown, including the use of interventions. Finally, with the introduction of CBMs, the authors made claims about the benefit of human-machine teaming and interpretability. These have not been verified with a human study *[Gap 7]*.

Table 2.2 shows a summary of the research contributions (RCs), research questions (RQs), gaps, and the corresponding chapter where we contribute to the literature.

A brief description of the items in Table 2.2 are as follows:

- RQ1 - Input feature mappings
- RQ2 - Encoded information

- RQ3 - Human-machine collaboration
- RC1 - Semantically meaningful concept representations
- RC2 - Dataset to support concept analysis
- RC3 - End-to-end feature attribution
- RC4 - Concept representations and resilience
- RC5 - Dataset requirements
- RC6 - Human studies
- RC7 - Model interpretability
- Gap 1 - Dataset variation for feature attribution analysis
- Gap 2 - Configurable dataset for CBM analysis
- Gap 3 - Concept contribution to task label predictions
- Gap 4 - Encoded information in concept representations over multiple CBMs
- Gap 5 - Dataset configurations to achieve semantically meaningful concept predictions, minimise extra encoded information and sensitivity to unrelated input features
- Gap 6 - Human-machine interaction with a CM
- Gap 7 - Unverified human-machine teaming and interpretability claims

2.5 Summary

In this chapter, we have outlined current CMs in the literature which aims to extend standard DNN architectures with additional capabilities to make them

Contribution	Question	Gap	Addressed
RC1	RQ1	Gap 1	Chapter 3.6
RC2	RQ1	Gap 2	Chapter 3.5.1
RC3	RQ1	Gap 3	Chapter 3.6
RC4	RQ2	Gap 4	Chapter 4.10
RC5	RQ2	Gap 5	Chapter 4.7
RC6	RQ3	Gap 6	Chapter 5
RC7	RQ3	Gap 7	Chapter 5

Table 2.2: Mapping of research questions, contributions and gaps.

more interpretable. We have discussed how XAI aims to aid in the human ability to build a mental model of an AI agent, and finally AI, XAI and CM human studies. Important to this thesis we have identified gaps with regards to understanding the representations of concepts CBMs learn, and their end-to-end decision-making process. We have highlighted that we should not conclude our evaluation with singular metrics, and thus we also evaluate CBMs in regards to information leakage and input feature dependency. Finally, we have identified the need for human studies to look at CBMs and their stated capabilities in a real-world task, and to evaluate human interaction.

Chapter 3

Feature Attribution in Concept Bottleneck Models

3.1 Introduction

In this chapter, we present a comprehensive analysis of how CBMs feature attribution is applied to input features from concept predictions and concept predictions from task label predictions w.r.t. the configuration of concepts in the models training datasets. CBMs have been positioned as improving human-machine collaboration as they are inherently interpretable (Koh et al., 2020). This capability is enabled by the model predicting a vector of human-defined concepts which are then used to predict a task label. Concept predictions can be inspected to reveal the decision-making process of a CBM task label prediction since the user can probe the CBM with various combinations of concepts. However, concept predictions may be misleading to humans interpreting the machine's outputs if the model does not predict concepts based on their expected input features, but the human assumes it does.

As previously mentioned in the Section 2.1.1, feature attribution values from concept predictions have been found to be distributed over the entire input image for a CBM trained on a dataset with class-level concept annotations. To address if this finding is restricted to certain training configurations, or is applicable to all trained CBMs, this chapter answers **RQ1** ("*How can we train a CBM to map semantically meaningful input features to concepts, and semantically meaningful concept predictions to task labels?*"). This question can be broken down into two sub-questions:

1. What dataset configurations, in particular concept annotations and concept correlation, are required to train CBMs to learn semantically meaningful mappings from input features to concept predictions, and from concept predictions to predicted task labels?
2. What is the most effective CBM training method?

We focus on training CBMs on images containing visual features that depict concepts. Expanding beyond the single model limitation of existing research we train and evaluate CBMs on three distinct datasets with different constraints applied to concept annotations. These include two real-world image datasets and one synthetic image dataset, covering both class-level and instance-level concept annotations (an example of the differences between the difference concept annotation methods was illustrated in Figure 2.3). By answering RQ1 we make the following contributions:

- **RC1**: We perform qualitative and quantitative analysis of CBMs, finding CBMs are capable of learning semantically meaningful concept representations from input features.
- **RC2**: We introduce and publish a new synthetic image dataset with fine-grained concept annotations which we use to demonstrate instances when CBMs can learn semantically meaningful concept representations and when they fail to do so.
- **RC3**: We expand on existing literature by looking at feature attribution both from the input to the concept vector and from the concept vector to the task output.

This chapter contains work in our papers "*Towards a Deeper Understanding of Concept Bottleneck Models Through End-to-End Explanation*", and "*Can we Constrain Concept Bottleneck Models to Learn Semantically Meaningful Input Features?*"

3.2 Motivation

This chapter looks at the research gap regarding how different configurations of concepts in training datasets influence CBM's ability to learn concept representations from input features, as well as task labels from concept annotations. For example, we examine which dataset configurations allow a model to predict concepts using semantically meaningful input features. This gap was discussed in Section 2.1.1 and Section 2.4. To achieve this we used gradient-based XAI techniques, primarily LRP (Bach et al., 2015), as this uses the model architecture and gradients of the models forward pass to work out feature attribution of input features w.r.t. concept and task label predictions. We used this approach rather than a proxy model, such as employed by the technique Local Interpretable Model-agnostic Explanations (LIME) (Ribeiro et al., 2016b), as this does not guarantee to show exactly which input features contributed to a model's output. In addition LRP groups attribution values to objects and not just individual pixels (Samek et al., 2021), which we see as a useful property as we may assume pixels representing concepts are grouped together in an input image. Groups of attribution values should be easier to interpret for an end user as there is less information to process compared to pixel-by-pixel attribution values.

CBMs are described as inherently interpretable, but as discussed in Chapter 2, this leads us to assume concepts are predicted using the same input features as a human performing the same task, e.g. it may be assumed a model will use the pixels for a wing of a bird in an image to predict the concept for the wing colour. CBMs inherent interpretability arises from concept explanations and the ability to intervene on concept predictions (Koh et al., 2020). Performing interventions allows humans to explore task predictions with counterfactual concept explanations. For instance, a human may correct a concept they believe the model got wrong and inspect any changes in the predicted task label. Although task predictions are made using concepts, and not input features from the original input sample, task predictions still rely on the accuracy of concept predictions. If concepts are not predicted using semantically meaningful input features then it's also possible that predicted task labels will use a different set of concepts than a human would, e.g. require the inclusion of concepts that are not visible. The main concern this creates is concepts could be learned such that there is a correlation between concepts that are not observed in the real world, and thus a model may be inaccurate outside of the models training data.

In Section 2.1.5 we discussed (Margeloiu et al., 2021) as the only work in the literature that explored how feature attribution is applied to input features from concept predictions. In particular, they found feature attribution values are distributed over the entire input image for concept predictions, and not localised to a small area of the image (e.g. feature attribution values cover the entire image of a bird instead of localised to the pixels representing the bird wing). However, their findings are narrow as they do not explore other class-level concept datasets, or datasets with instance-level concepts. In addition they hypothesise existing feature attribution methods are ill-equipped for CBM evaluation. We do not have any indication if their findings hold for other datasets, and if existing feature attribution techniques are indeed the limiting factor.

As discussed in Section 2.4, we have not identified any papers that evaluate feature attribution from predicted task labels to predicted concepts. Without this analysis it is unclear which concept predictions models use to predict task labels, and whether these are aligned to human decision-making.

We have identified most research looking at CMs focus on the training method or model architecture, as identified in Section 2.1, but it remains unclear if CBMs and similar models are incapable of representing concepts using human aligned input features. Therefore we see a need to investigate how CBMs learn concept representations when trained on datasets with concept configurations other than class-level concepts. In this chapter we focus on exploring learned concept representations with datasets configured with class and instance-level concepts, in addition to the accuracy and correlation of concept annotations. We also evaluate which concepts are used for downstream task label predictions, and how these align to ground truth concept values. Further, we utilise additional metrics to provide a quantitive evaluation of the otherwise qualitative results saliency maps provide.

3.3 Feature Attribution

In its simplest form when a DNN performs a forward pass input features, such as pixels from images, are passed from layer to layer in the model where the input of each layer is the output of the previous layer (Krizhevsky et al., 2012). As part of this process a gradient will be computed which we can utilise with XAI techniques such as LRP (Bach et al., 2015) or IG (Sundararajan et al., 2017) (as introduced in Section 2.2.1) to produce a local explanation to reveal how much each input feature contributed to the task prediction. We can visualise the explanation as a saliency map.

As discussed in Section 2.1, CBMs are expected to predict concepts using input features with the same meaning. However, Margeloiu et al. (2021) indicates this is not the case with models trained on datasets using class-level concept annotations. Margeloiu et al. (2021) demonstrated their model assigned feature attribution values over the entire input image instead of confined to the semantically meaningful input features. Margeloiu et al. (2021)'s model was trained on CUB (Wah et al., 2011), a popular dataset for CBM research. Most concepts in CUB represent bird parts, however, if the image is cropped, or the bird changes appearance with gender or age, concepts may no longer match the visual appearance in the image.

As detailed in Section 2.4, we have not identified prior work that analyses if the configuration of concepts in a dataset can confine a CBM to predict concept using semantically meaning input features. For instance, datasets with instance-level concepts avoid the inaccurate concept annotations seen with CUB as concepts can be fine-grained, only marking concepts to present when their visual representations can be identified in the input. We cannot jump to the conclusion that instance-level concepts are all you need. The original CUB dataset (before (Koh et al., 2020) modified it with class-level concepts) had instance-level concepts, but these were noisy (Koh et al., 2020) which itself may restrict a CBM from learning to map semantically meaningful input features to concepts.

In addition to concept explanations, we can also look at which concepts contributed to task predictions. Concept predictions from the output of a CBM concept encoder are used as the input features to the model's task predictor. Using XAI techniques, we can identify the contributions of concepts for the task prediction. This may be interpreted as the rules the model has learned to map concepts to the downstream task. For instance, CBMs may learn that for the task class "Mallard", the concept for a green head and orange feet need to be present. With some XAI techniques, such as LRP, feature attribution values that are propagated through a model are conserved (Bach et al., 2015). As such, we can take this a step further and convert the attribution values to show the proportion of the contribution of each concept w.r.t. the predicted downstream task label.

3.4 Methods

Before we move onto the experiment set-up we must remind ourselves of RQ1: "How can we train a CBM to map semantically meaningful input features to concepts, and semantically meaningful concept predictions to task labels?"

RQ1 fundamentally ask whether a CBM can predict concepts and task labels in alignment with the presence of semantically meaningful features in the input data. These features may include visually identifiable elements in an image corresponding to concepts, or concept annotations indicating the presence of concepts in the dataset. Sub-question 1 and 2 then asks what the requirements are for a CBM to learn to make predictions using semantically meaningful features. These requirements could be related to the dataset configuration in Sub-question 1 or the training method in Sub-question 2.

To address RQ1, we need to evaluate if different dataset configurations and each CBM training method (independent, sequential and joint) enable a model to learn to map semantically meaningful input features to concepts. For dataset configurations specifically, we will need to include datasets with class-level concept annotations (e.g. CUB (Wah et al., 2011)) where all samples of a class shares the same concept vector, and datasets with instance-level concept annotations (e.g. in our synthetic Playing Cards dataset) where concepts are annotated on a per sample basis. Concept annotations must include ground truth knowledge of the semantically meaningful input features associated with each concept. This ground truth information is necessary to validate that the learned feature mappings align with the ground truth input features that represent each concept. Given the challenges of defining ground truth features in some domains (e.g. emotions associated with images of faces), we limit our datasets to images with concepts representing visually identifiable attributes. A complete list of the datasets we've used for evaluation is detailed in Section 3.5.1.

We have used feature attribution techniques to evaluate which input features a CBM uses for concept and task predictions. As we detailed in Section 2.2.1, feature attribution techniques can reveal the contributing input features for a CNN prediction and have been useful to highlight model bias (Ribeiro et al., 2016b). As we intend to reveal the input features used for concept and task predictions, this capability aligns with our requirements to answer RQ1. Additionally, we used model-specific techniques such as LRP (Bach et al., 2015) as these use the model itself instead of a proxy model to produce feature attribution values. Compared to model-agnostic methods this will be a better representation of the true input features used for a model prediction. For consistency, we display positive attribution values in red and negative attribution values in blue.

Answering RQ1 requires quantitative results. As discussed in Section 2.2.1, feature attribution techniques, produce local explanations of model predictions. While visualising explanations as saliency maps provide a qualitative evaluation of model behaviour. For quantitive evaluation we measured the alignment between the feature attribution applied to input features and ground truth concept locations, and then averaged the individual results across testing dataset samples. This aggregation helps assess whether model predictions rely on semantically meaningful features and concepts.

We produced quantitive results with three metrics:

1. We adapted The Pointing Game evaluation technique (Zhang et al., 2018) to measure the distance between the highest feature attribution value and the ground truth input feature point for concepts.
2. We evaluated the proportion of feature attribution values that overlapped ground truth concept locations.
3. We used IoU to evaluate which concepts were used for task predictions.

These measurements are necessary as they allow us to quantify how effective a CBM is able to make predictions using input features that are semantically meaningful to a prediction.

3.5 Experiment Set-up

As discussed in the literature review, we have identified CBMs have not been explored in regards to how they represent concepts, and how concepts are used to predict task labels. To evaluate CBMs we need to train models on multiple datasets with suitable evaluation techniques that reveal the learned concept representations and decision-making process. We discuss the datasets and evaluation setup based on XAI techniques in this section.

3.5.1 Datasets

We trained and evaluated our models on three datasets: CUB (Wah et al., 2011), a bird image dataset with class-level concepts showing visual attributes, playing cards, a new synthetic image dataset we've introduced to evaluate CBMs with accurate concept annotations, and CheXpert (Irvin et al., 2019), an image dataset which has instance-level concepts representing visual attributes of each input image. These datasets are summarised in Table 3.1.

These three datasets are necessary for our evaluation of CBMs as they represent a number of different configurations of datasets that a CBM can be trained from. Starting with CUB, this dataset has class-level concepts and thus demonstrates cases when concepts may not have a visual representations in the input images. Next, Playing cards show situations when concept annotations are always accompanied by a visual representation in the input image. Playing cards include variations of class and instance-level concept annotations and vary the correlation of concepts present at the same time. Finally, CheXpert represents a real-world dataset where concept annotations are accompanied by a visual representation in the input image.

These datasets are sufficient for our evaluation as they contain all required variations of concept annotations to align with our motivations. Specifically, they

	Dataset	Concept annotation type	Number of samples	Number of Concepts	Number of Classes
	CUB	Class-level	11,788	112	200
Playing cards	Poker cards	Instance-level	10,000	52	6
	Random cards	Instance-level	10,000	52	6
	Class-level Poker cards	Class-level	10,000	11	6
CheXpert	Instance-level CheXpert	Instance-level	224,316	13	2
	Class-level CheXpert (three concepts)	Class-level	44,974	13	2
	Class-level CheXpert (four present concepts)	Class-level	21,760	13	2
	Class-level CheXpert (five present concepts)	Class-level	636	13	2

Table 3.1: Summary of datasets.

include instance and class-level concepts, a variance in co-occurring concepts (quantified using the Pearson correlation coefficient, as detailed in Section 4.5), instances where concepts are accompanied by a visual representation, and instances where they are not. These align with our motivations as they allow us to separate individual changes in concept configurations and thus measure their effect to train CBMs to learn to predict concepts using semantically meaningful input features.

3.5.1.1 CUB

*CUB (Wah et al., 2011) is a dataset containing 11,788 images of birds. Each image is accompanied by attributes representing the visual features of the bird. We have provided an example of a sample from the dataset in Figure 3.1. For

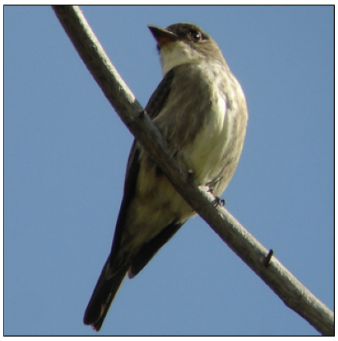
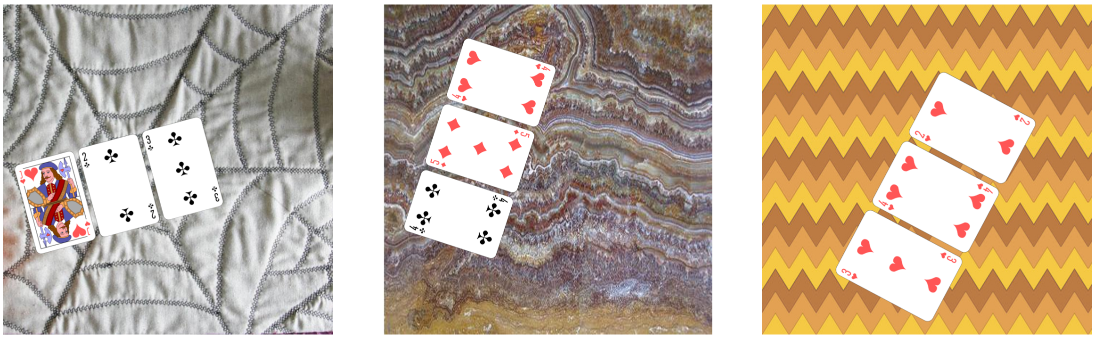
Input	Present concepts
	has_bill_shape::all-purpose has_wing_color::grey has_upperparts_color::grey has_underparts_color::grey has_back_color::grey has_tail_shape::notched_tail has_head_pattern::plain has_breast_color::grey has_eye_color::black has_bill_length::shorter_than_head has_forehead_color::grey has_under_tail_color::grey has_belly_color::grey has_size::small_(5_-_9_in) has_shape::perching-like has_tail_pattern::solid has_primary_color::grey has_leg_color::black has_bill_color::black has_crown_color::grey has_wing_pattern::multi-colored

Figure 3.1: Example CUB sample with the task label "olive_sided_Flycatcher", with concept that are annotated as present. Any concept not listed from the full 112 available in the dataset are not present in this sample

our study, we are using a modification by (Koh et al., 2020) which altered the attributes to be set at the class-level which were selected using majority voting. This was to remove noise in the original annotations. In total, there are 112 concepts and 200 task labels. Images were centre-cropped and resized to 299 x

299 pixels for training. The official splits were modified with 20% of the original training samples moved to a new validation set. The test set was not modified.

3.5.1.2 Playing cards



(a) Random cards

(b) Poker cards

(c) Class-level poker cards

Figure 3.2: Samples from the Playing cards dataset.

Playing cards is a synthetic image dataset we introduced to analyse CBMs free of inaccurate concept annotations. The dataset consists of multiple image variations, of which we use three in this thesis: *Random cards*, *Poker cards*, and *Class-level poker cards*. Example samples can be seen in Figure 3.2. Each variation consists of 10,000 images where concepts represent playing cards, and task labels represent hand ranks in the game Three Card Poker (of Odds, 2024). In our dataset cards are placed onto a random background image as we did not want to introduce an unintentional bias the models could learn.

In the card game hand ranks are formed by holding three playing cards at the same time in what is called a card hand. Players aim to beat the dealer by having a card hand with a lower probability than the dealer. For instance, the card hand with the rank “flush” has a 4.96% probability of occurring and thus beats the card hand with the rank “pair” which has a probability of 16.94%. Players place bets on if they believe their hand ranks higher than the dealer.

Concepts for Random cards are selected at random from the full 52 possible playing cards present in a standard deck of playing cards, with no repeats. This ensures that there is no correlation between which concepts occur together; however, it introduces a class imbalance in the dataset. For instance, the class "high card" has 5191 training samples, while the class "straight flush" has 20 training samples.

For Poker cards, task classes are balanced, with concepts selected from the sets of triplets available based on the class used in each sample. This results in some concepts appearing together more often than others. For example, there are 48 unique triplets of concepts for the class "straight flush," while this class has 1166 training samples.

Class-level poker cards have the same task classes as Random cards and Poker cards but use only 11 concepts instead of 52, with one triplet of concepts used for each task class. Some concepts are only used for one task class, while others are used for many. We have listed the full concepts and relations to task classes in Table 3.2, and show which concepts appear together in Figure 3.3 using a Chord diagram. Each concept is linked to other concepts that appear together, with the thickness of each link representing the number of samples where concepts co-occur.

Each image variation has a 70%-30% split between training and validation images. Random cards and Poker cards use instance-level concepts, while Class-level poker cards use class-level concepts. In all cases, if a concept is annotated as present, then it is visible in the corresponding image.

During training we transform training sample images using a random flip (both horizontal and vertical), apply a colour jitter to the brightness, contrast, saturation, and hue, and randomly convert them to a greyscale image. Samples are scaled to 299 by 299 pixels.

The dataset is publicly available[1] along with the code to generate the dataset[2].

[1] Playing cards dataset: `https://huggingface.co/datasets/JackFurby/playing-cards`

[2] Playing cards dataset generator: `https://github.com/JackFurby/playing-card-`

Task label	Concepts
Straight Flush	2 of ♡, 3 of ♡, and 4 of ♡
Three of a Kind	4 of ♣, 4 of ♢, and 4 of ♠
Straight	3 of ♡, 4 of ♣, and 5 of ♢
Flush	4 of ♢, 6 of ♢, and 9 of ♢
Pair	5 of ♣, 5 of ♢, and 10 of ♡
High Card	4 of ♠, 5 of ♢, and 10 of ♡

Table 3.2: Concepts for class-level poker cards are arranged such that some are used for one task class while others are used for many task classes

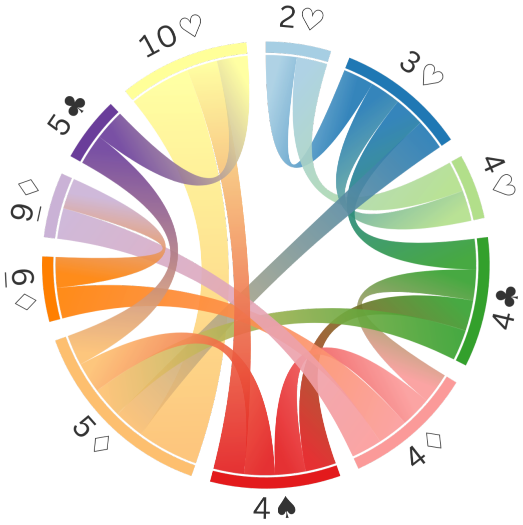


Figure 3.3: Class-level poker cards has concepts organised so that certain concepts consistently appear together (e.g. Two of Hearts, Three of Hearts, and Four of Hearts), while others co-occur with a range of concepts (e.g. Five of Diamonds will co-occur with the Five of Clubs in some sample images, and the Four of Spades in other sample images)

`concept-generator`

3.5.1.3 CheXpert

CheXpert (Irvin et al., 2019) is a real-world image dataset with visually represented observations. Each sample has 14 observations such as “fracture” and “edema”, of which 12 are pathologies. The other two observations are “support devices” (e.g., a pacemaker) and “no_findings.” We use 13 of these observations as concepts, while the observation “no_findings” is used as the task label. “No_findings” is positive if all pathologies are not annotated as present. We have provided an example sample with concept annotation in Figure 3.4.

CheXpert has instance-level concepts and contains 224,316 chest X-ray images. We use the official dataset splits from (Irvin et al., 2019), which include 223,414 training images, 234 validation images, and 668 test images. Training annotations were automatically generated from radiology reports. Observations were labelled as 1 when confidently present, 0 when confidently not present, and -1 when uncertain. To translate these labels into binary annotations, we used U-ones annotations, which set any missing values to 0 and any uncertain annotations to 1. Validation images were labelled by three board-certified radiologists, while test images were labelled by eight board-certified radiologists. Both validation and test images include only binary annotations.

We also created a modified version of the dataset with class-level concepts, using the most common concept vector for samples with three, four, and five concepts present. Class-level CheXpert has 44,974 samples, 21,760 samples, and 636 samples for three, four, and five concepts present, respectively. We refer to the original version of CheXpert as *instance-level CheXpert* and the modified version as *class-level CheXpert*.

During training, samples are randomly rotated by up to 15 degrees, translated by up to 5% of the overall image width and scaled by up to 5%. All samples are resized to 512 by 512 pixels.

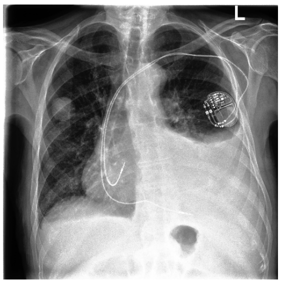
Input	Concepts	Class label
	Enlarged cardiomediastinum: present Cardiomegaly: present Lung opacity: present Lung lesion: present Edema: not present Consolidation: not present Pneumonia: not present Atelectasis: present Pneumothorax: not present Pleural effusion: present Pleural other: not present Fracture: not present Support devices: present	No Findings: True

Figure 3.4: Example CheXpert sample with concept annotations and class label.

3.5.2 Models

All of our models use a similar structure and the same training methods specified in (Koh et al., 2020). Keeping the overall model structure consistent between models allows us to measure the effect the dataset has on training a CBM, and thus answer RQ1. The models structure starts with the concept encoder which receives an input and outputs a concept vector with one value for each concept. This is followed by an optional sigmoid function which receives the concept vector as an input and outputs concept predictions. The sigmoid function increases the independence of concept representations (Espinosa Zarlenga et al., 2023). Finally, this is followed by the task predictor which receives concept predictions if a sigmoid function is used, or the concept vector if it is not, as an input and outputs a task prediction. We use Binary Cross Entropy loss to train the concept

Training method	LR	Optimizer	Batch size	λ	Epochs
Independent & sequential concept encoder	0.01	SGD	32	N/A	500
Independent task predictor	0.001	SGD	32	N/A	500
Sequential task predictor	0.001	SGD	32	N/A	1000
Joint	0.001	SGD	32	0.99	1000

Table 3.3: CUB models training hyperparameters.

encoder, and Cross Entropy loss for the task predictor. Concept accuracy is the average binary accuracy of concept predictions with a 0.5 threshold. We repeated training 5 times for Playing cards and CheXpert models, and 3 times for CUB models. For some experiments, such as those involving saliency maps, we use the model with the highest concept accuracy.

CUB models were trained using the three CBM methods; independent, sequential and joint, where independent models have a sigmoid layer between the two model parts, while both sequential models and joint models either had the sigmoid layer or passed the output from the concept encoder model part directly to the task predictor model part. We used the same hyper-parameters used in (Koh et al., 2020) which are detailed in Table 3.3, and a modified repository than that used in (Koh et al., 2020) for training the models[3]. The averaged concept and task accuracies are shown in Table 3.4. Each model trained on CUB used a Visual Geometry Group (VGG)-16 architecture (Simonyan and Zisserman, 2015) for the concept encoder model part and a single linear layer for the task predictor model part.

Playing cards models use a VGG-11 architecture with batch normalisation (Simonyan and Zisserman, 2015) for the concept encoder and two linear layers with a ReLU activation function for the task predictor. We trained these models to minimise the concept and task loss. We used Weights and Biases Sweeps (Biewald,

[3] CUB model training repository: `https://github.com/JackFurby/VGG-Concept-Bottleneck`

Training method	Concept accuracy	Task accuracy
Independent	96.85 (±0.1)	77.51% (±0.4)
Sequential without sigmoid	96.85% (±0.1)	75.35% (±0.08)
Sequential with sigmoid	96.72% (±0.1)	77.28% (±0.59)
Joint without sigmoid	96.12% (±0.08)	78.75% (±0.65)
Joint with sigmoid	94.87% (±0.03)	75.35% (±0.31)

Table 3.4: Summary of CUB models.

2020) to find optimal hyper-parameters for training each of our models which is summarised in Table 3.5. This was configured with a Bayesian search method to optimise the parameters. The parameters were starting Learning Rate (LR) (between 0.1 and 0.001), optimizer (between Adam (Kingma and Ba, 2014) and Stochastic Gradient Descent (SGD)), LR patience (between 3, 5, 10 and 15 epochs of no improvement in loss) and λ value between 0.9 and 1.0. Each sweep ran until we stopped seeing improvements in the model accuracy, about 30 iterations per sweep. To compare CBMs with standard DNNs we also trained models on Poker cards using the same model architecture but without concept loss. All average model accuracies are shown in Table 3.6.

The standard DNNs model shows significantly lower task and concept accuracies compared to all other models. This difference is because the model shares the same architectural design, including the bottleneck layer, yet it is not trained with any concept supervision. While the CBMs are trained to learn meaningful concept representations, the standard DNN is trained end-to-end only with task supervision. As a result, the model is restricted by the bottleneck layer but lacks the necessary concept-level guidance to make efficient use of it, leading to both reduced concepts and task accuracies.

CheXpert models use a Densenet121 architecture (Huang et al., 2017) for the concept encoder which is initialised with pre-trained weights from ImageNet and two linear layers with a ReLU activation function for the task predictor which

Training method	Dataset	LR	Optimizer	Batch size	LR patience	λ	Epochs
Independent & sequential concept encoder	Random cards	0.03	SGD	32	15	N/A	200
Independent & sequential concept encoder	Poker cards	0.02	SGD	32	15	N/A	200
Independent & sequential concept encoder	Class-level poker cards	0.0825	SGD	32	3	N/A	100
Independent task predictor	Random cards	0.01	Adam	32	5	N/A	200
Independent task predictor	Poker cards	0.01	Adam	32	5	N/A	200
Independent task predictor	Class-level poker cards	0.064	Adam	32	5	N/A	100
Sequential task predictor	Random cards	0.059	Adam	32	5	N/A	200
Sequential task predictor	Poker cards	0.046	Adam	32	15	N/A	200
Sequential task predictor	Class-level poker cards	0.0846	Adam	32	10	N/A	100
Joint	Poker cards	0.025	SGD	32	15	0.98	300
Joint	Class-level poker cards	0.0398	SGD	32	15	0.867	100
Standard DNN	Poker cards	0.088	SGD	32	15	0	300

Table 3.5: Playing cards training hyperparameters.

is not pre-trained. We trained these models to maximise the Area Under the receiver operating characteristic Curve (AUC) of concept predictions, following previous work (Ye et al., 2020; Chauhan et al., 2023), and minimise the task loss. We trained our models with the hyper-parameters in Table 3.7. CheXpert models were trained using the independent/sequential method for the concept encoder, and the sequential method for the task predictor. Models trained with the joint method used a sigmoid layer between the two model parts. Average model metrics are shown in Table 3.8.

Training method	Dataset	Average concept accuracy	Average task accuracy
Independent	Random cards	99.94% (±0.01)	99.17% (±0.09)
Sequential	Random cards	99.92% (±0.04)	97.46% (±0.76)
Independent	Poker cards	99.96% (±0.01)	99.42% (±0.03)
Sequential	Poker cards	99.92% (±0.05)	98.80% (±0.28)
Joint	Poker cards	99.87% (±0.05)	96.01% (±0.21)
Independent	Class-level poker cards	99.98% (±0.01)	99.96% (±0.04)
Sequential	Class-level poker cards	99.98% (±0.014)	99.95% (±0.05)
Joint	Class-level poker cards	100% (±0)	100% (±0)
Standard DNN	Poker cards	50.25% (±1.31)	67.14% (±0.58)

Table 3.6: Playing card models averaged accuracy and standard deviation. All values are rounded to 3 decimal places

Training method	LR	Optimizer	Batch size	λ	Epochs
Sequential concept encoder	0.001	Adam	14	N/A	3
Sequential task predictor	0.001	Adam	14	N/A	3
Joint	0.001	Adam	14	0.99	3

Table 3.7: CheXpert models training hyperparameters.

3.5.3 LRP Configuration

A prominent XAI technique we use to calculate input feature attribution is LRP. LRP uses the term *relevance* to describe the contribution of individual input features or neurons to a specific output prediction, as redistributed through the layers of a model. Relevance is directly comparable to feature attribution. In this thesis, we will use the term relevance specifically to describe the contributions calculated using LRP, while referring to the final outputted values as feature

Training method	Dataset version	Concept accuracy	Task accuracy
Sequential	Instance-level	75.77 (±1.12)	84.70 (±0.63)
Joint	Instance-level	74.70 (±1.22)	85.15 (±0.62)
Sequential	Class-level with 3 present concepts	59.18% (±8.80)	95% (±0.69)
Sequential	Class-level with 4 present concepts	63.90% (±9.75)	95.71% (±1.43)
Sequential	Class-level with 5 present concepts	65.28% (±9.05)	96% (±1.33)
Joint	Class-level with 3 present concepts	56.47% (±2.89)	96.76% (±1.16)
Joint	Class-level with 4 present concepts	60.39% (±4.16)	97.14% (±2.67)
Joint	Class-level with 5 present concepts	61.95% (±4.17)	95.33% (±1.63)

Table 3.8: CheXpert models averaged accuracy and standard deviation. All values are rounded to 2 decimal places

attribution.

LRP supports rules which change how relevance is propagated from a prediction back to the input (Bach et al., 2015; Montavon et al., 2019). With rule selection, we aim to produce saliency maps that accurately explain the input features leading to concept predictions. Using a single uniform LRP rule across the entire model yielded misleading results where, no matter which concept we attempted to visualise, we were presented with a saliency map which is seemingly similar to the next. This was most noticeable with the LRP-$\alpha\beta$ rule. However, alternative rules also had the added drawback of appearing noisy, making the explanation less understandable to a human collaborator (Montavon et al., 2019). In general, singular LRP rules applied to an entire model results in explanations that are not class-discriminative (Gu et al., 2019; Kohlbrenner et al., 2019). Changing to composite rules rectifies this issue, allowing us to visualise feature attribution values of input features that both contributed positively and negatively to

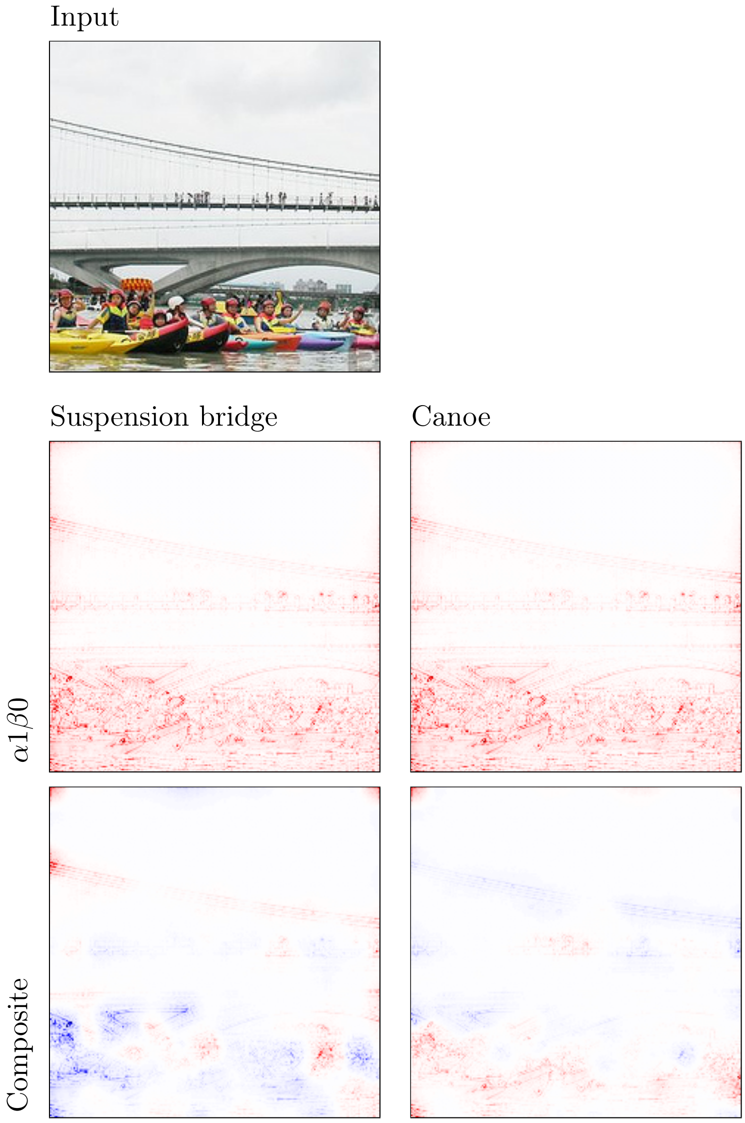


Figure 3.5: Singular LRP rules can result in saliency maps that are distinctly similar between output classes. This similarity does not occur with composite rules. Positive attribution values are shown in red and negative attribution values are shown in blue.

the concept prediction, with each concept saliency map being distinctly different from the others. Composite rules allow LRP rules to be applied to individual layers in a DNN (Montavon et al., 2019). An example of singular and composite rules can be seen in Figure 3.5 where, with an input of people in canoes and a

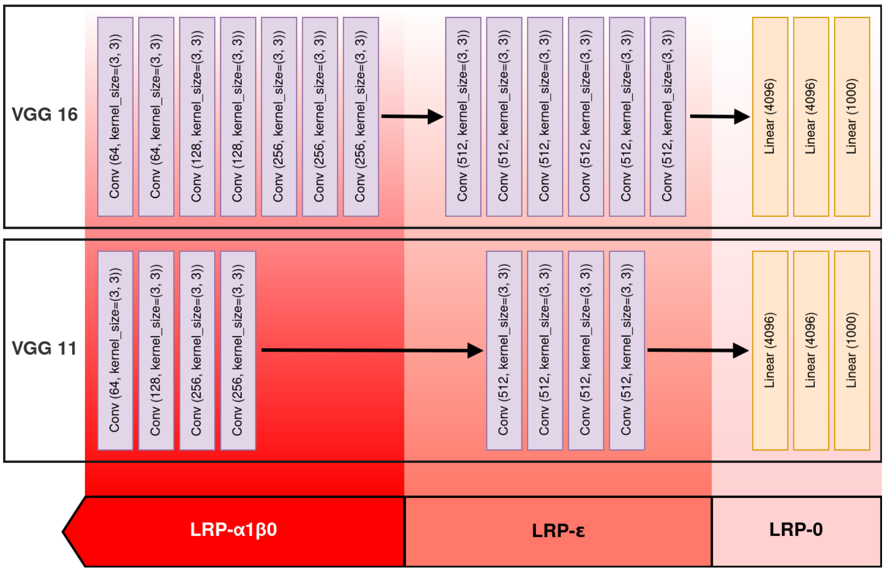

VGG 16
Conv (64, kernel_size=(3, 3))
Conv (64, kernel_size=(3, 3))
Conv (128, kernel_size=(3, 3))
Conv (128, kernel_size=(3, 3))
Conv (256, kernel_size=(3, 3))
Conv (256, kernel_size=(3, 3))
Conv (256, kernel_size=(3, 3))
Conv (512, kernel_size=(3, 3))
Conv (512, kernel_size=(3, 3))
Conv (512, kernel_size=(3, 3))
Conv (512, kernel_size=(3, 3))
Conv (512, kernel_size=(3, 3))
Conv (512, kernel_size=(3, 3))
Linear (4096)
Linear (4096)
Linear (1000)
VGG 11
Conv (64, kernel_size=(3, 3))
Conv (128, kernel_size=(3, 3))
Conv (256, kernel_size=(3, 3))
Conv (256, kernel_size=(3, 3))
Conv (512, kernel_size=(3, 3))
Conv (512, kernel_size=(3, 3))
Conv (512, kernel_size=(3, 3))
Conv (512, kernel_size=(3, 3))
Linear (4096)
Linear (4096)
Linear (1000)
LRP-α1β0
LRP-ε
LRP-0

Figure 3.6: Different LRP rules are applied to individual layers in the VGG model architecture we use for some concept encoders.

bridge in the background. Using just the rule $\alpha 1 \beta 0$ the saliency maps for "canoe" and "suspension bridge" predictions are the same, whereas using composite rules the prediction for "suspension bridge" applies positive attribution values to the bridge and negative values to the people in canoes. For the class "canoe" the feature attribution values are reversed.

Throughout the thesis, we have used LRP rules similar to (Montavon et al., 2019) for the concept encoder. These rules are: LRP-$\alpha\beta$, where $\alpha = 1$ and $\beta = 0$, where α and β controls the contribution of positive and negative relevance respectively, for the first convolutional layers, LRP-ϵ for the middle convolutional layers and LRP-0 for the top linear layers (Bach et al., 2015), as seen in Figure 3.6. These rules avoided feature attribution being applied on a pixel-by-pixel basis as observed with single LRP rules and instead attribution values are applied to regions of the input Samek et al. (2021).

In addition to the concept encoder, we can also utilise LRP for the task predictor to analyse feature attribution applied to predicted concepts. As the task predictor for our models is only comprised of one or two layers we have opted to use LRP-

0 for the entire model part as this will propagate both positive and negative relevance. LRP ensures relevance is conserved as it is propagated backwards through a model (Bach et al., 2015; Montavon et al., 2019) which we can use to compute the proportion each concept contributed to the task prediction. We calculated the proportion of contribution P for k concepts in the concept input using the relevance R that is propagated backwards to each concept C from the task classification, as shown in equation 3.1. Each value of P_k is weighted similarly to the method used by (Taylor et al., 2020) but instead of applying the weighting to two modalities, we apply the weighting to k concepts. As each concept is a single value we do not need to account for imbalance in concept proportions.

$$P_k = \frac{X_k}{\sum_{n=1}^{k} X_n} \tag{3.1}$$

where

$$X_k = \frac{R_k}{C_k} \tag{3.2}$$

3.6 Results

With our CBMs, training data, and XAI based evaluation techniques detailed we now show qualitative results for feature attribution as saliency maps and quantitative results where we average the relevance attribution w.r.t. the ground truth values, where the datasets allow.

These results help to answer Sub-question 1 as we evaluate each of our CBMs, paying particular attention to how the configuration of concepts in the training data changes how a CBM applies feature attribution values to input features from concept predictions.

3.6.1 Input Feature Attribution

3.6.1.1 CUB

We start by producing saliency maps for our CUB models. Figure 3.7 shows the feature attribution applied to input features for a range of concepts which a human would expect to map to distinct regions of the input. Specifically, these are concepts that identify a particular part (e.g. wing or beak) of a bird instead of the bird as a whole (e.g. bird size). Regardless of the training method used, the saliency maps indicate that the models have not learned how to map distinct regions in the input to concept labels. Feature attribution is generally distributed over the entire bird although, an observation with our models is the eyes of the bird appear to be the most common group of input features where feature attribution values are either highly positive or negative.

Concepts with similar predictions also appear to share similar saliency maps. This is evident in Figure 3.7 with the independent and sequential models and concepts "has_crown_color::brown" and "has_wing_shape::pointed-wings" which have a predicted concept value of 0.9973 and 0.9980 respectively to four decimal places. For the joint-without-sigmoid model, "has_back_color::brown" has a predicted concept value of 0.9918 and "has_breast_pattern::solid" has a predicted concept value of 0.9975. The similarity between saliency maps likely means that each model has learned the same input features can accurately predict different concepts.

Our results confirm CBMs trained on the CUB dataset do not learn distinct regions from the input to concepts, as Margeloiu et al. (2021) showed. This is likely due to the training data or training methods not constraining the model to do so. Like regular bottleneck models (Grezl et al., 2007), CBMs will typically only keep the most important input features, in this case, to fit the concept vector, but leave the CBM to select which input features to use. In addition, by using class-level concepts the model learns the concept vector but not if a concept is

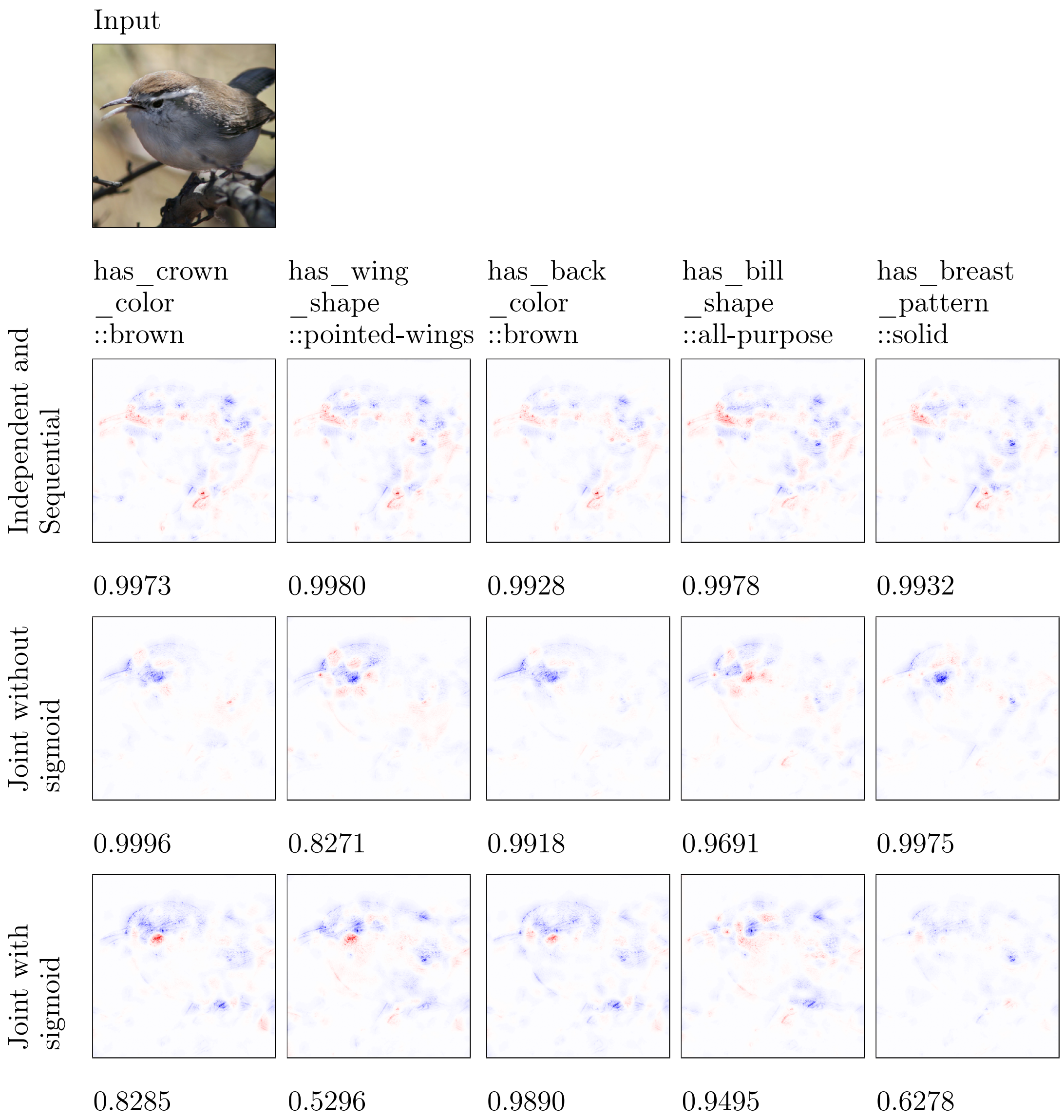


Figure 3.7: Concept saliency maps for the input image of a Bewick Wren where concepts are correctly predicted. Positive attribution values are shown in red, negative attribution values are shown in blue and the predicted concept value to four decimal is placed below each saliency map (a value of 0.5 or higher means the concept was predicted as present). In general, feature attribution is not applied to input features that a human would associate each concept with.

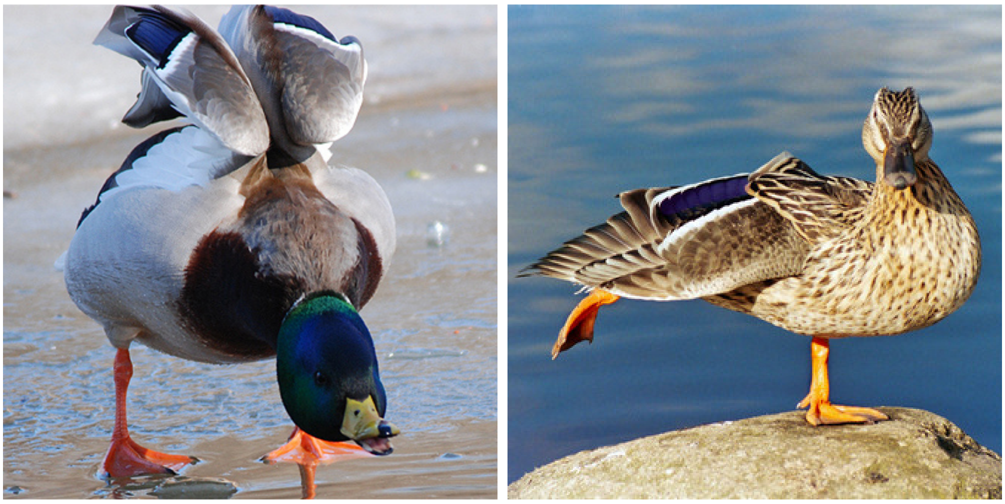


Figure 3.8: The version of the CUB dataset used in this thesis has class-level concept annotations. This results in some samples having concept annotations that do not represent the visual representation of concepts in the sample. In this example the downstream task class Mallard has concepts for both male (left image) and female (right image) ducks that are not shared by both genders.

present and visible in a given sample. Koh et al. (2020)'s version of CUB also has incorrect concepts. For example, the class "Mallard", as seen in Figure 3.8, has the same concept vector for males and females despite the visual differences between them. For example, the concept "has_wing_color::white" is correct for male Mallard ducks but not females, while the concept "has_upperparts_color::brown" is true for female Mallard ducks but not males. Concept ambiguity has also been identified by (Kim et al., 2023b). If concepts are not carefully considered when designing a dataset then there could be concepts that always appear together, potentially causing unintentional concept correlations, concepts that only appear for one downstream task, opening a shortcut the model may use for downstream task prediction, or concepts that do not have a visual representation in the input. We hypothesise that using a dataset with accurate and well-defined concepts, a CBM can learn concepts such that feature attribution values are applied to semantically meaningful input features.

Beyond individual saliency maps, we also evaluated our CUB models with a modified version of The Pointing Game (Zhang et al., 2018) which we have named The Distance Pointing Game. The Pointing Game counts hits and misses of whether the point with the highest feature attribution value of a given sample explanation is placed in a defined region, the ground truth, resulting in an accuracy measurement. Our version measures the distance between the point with the highest attribution value and the ground truth point. This was necessary because CUB includes bird part locations, but does not provide bird part bounding boxes. We present the results as an average distance. Our technique does not replace The Pointing Game, but instead, it satisfies a different situation; when you have ground truth points. By using our evaluation technique, we can quantify whether an explanation technique for a given model's output is primarily focusing on a ground truth point. We can also rank feature attribution techniques or models, which enables us to analyse whether our CUB models are applying feature attribution to semantically meaningful input features. We used the explanation technique IG with a SmoothGrad noise tunnel (Smilkov et al., 2017) using a batch size of 25 and a standard deviation of 0.2, similar to (Margeloiu et al., 2021), LRP, and a baseline gradient method (Simonyan et al., 2014).

We measured the average distance using our independent model, due to that model having the highest concept accuracy, using the validation dataset split. Results are shown in Figure 3.9. Lower average distance shows increased alignment between saliency maps and ground truth bird part locations. IG has around a 3rd higher average distance compared to both LRP and the baseline gradient for most bird parts while LRP and the baseline have similar average distances. To remove noisy saliency maps we also show the average distance of the shortest 10% of distances which follows the same story as the overall distance averages. While IG assigns attribution values to individual pixels, LRP with our rules groups feature assigns attribution values to regions of input features. As a result, LRP saliency maps are filtering out noisy attribution values from the input image.

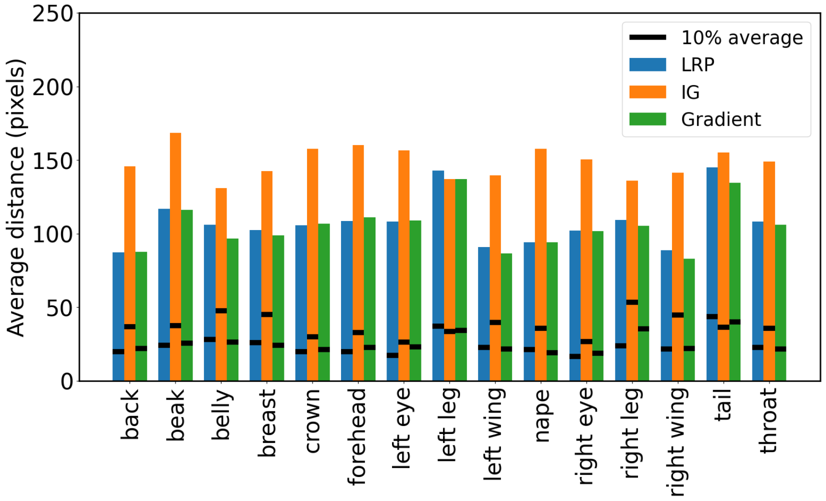


Figure 3.9: Distance Pointing Game results comparing LRP, IG and a baseline gradient method. LRP and gradient has a shorter average distance for most bird parts compared to IG. This remains the same for when averaging the shortest 10% of distances.

However, the average distance hovers around 100 pixels away from the ground truth point with LRP and, considering the input images are 299 by 299 pixels in size, this could still fall outside of the concept in the input image, adding to what we observed in Figure 3.7 with attribution values generally covering the entire bird.

3.6.1.2 Playing cards

CBMs trained on CUB only explores a single configoration of concept annotations in a dataset. To conclude whether CBMs can learn to predict concepts using semantically meaningful input features we also need to look at other datasets. For the first of these, we used our dataset Playing cards. Playing cards is a synthetic dataset where we can ensure concept annotations always describe the visual representations of concepts in each sample. Figure 3.10 shows concept saliency maps using the XAI technique LRP for Random cards and Poker cards

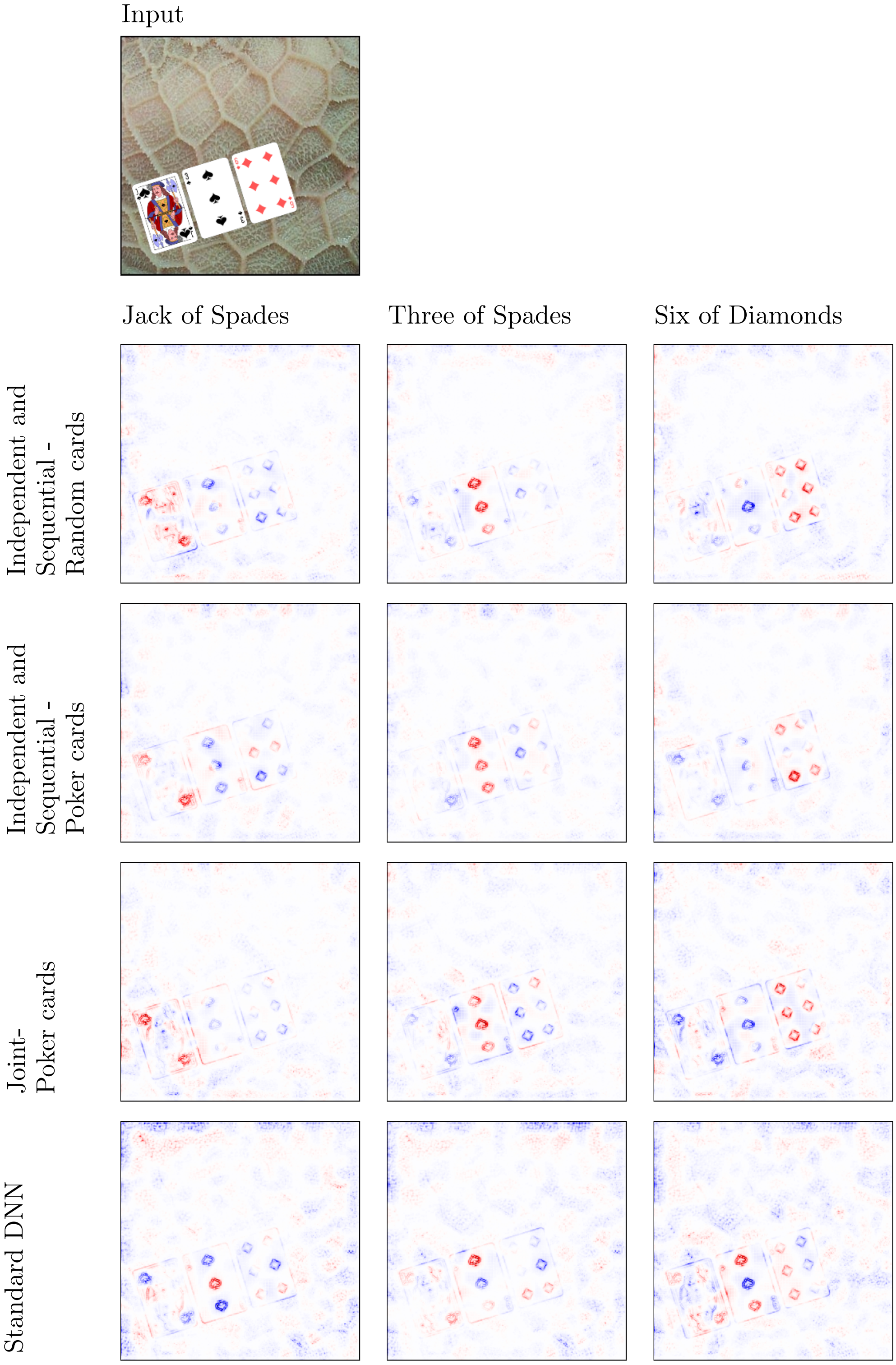


Figure 3.10: Concept saliency maps show positive feature attribution values are applied to the expected input features for CBMs while distributed over all playing cards for the standard DNN.

CBMs. The input features with the highest feature attribution values are symbols on the playing cards with negative feature attribution distributed over the other playing cards. Specifically, positive feature attribution values are distributed to the input features that correctly represent concepts. As we do not specify which input features the model should use for each concept, and we can see the models have selected features within the boundaries of each specified playing card, we consider these reasonable input features for the model to use. Our standard DNN was unable to localise feature attribution values to individual playing cards and instead distributed attribution values over all three cards present.

In Figure 3.11 and Figure 3.12 we show saliency maps for models trained on Class-level poker cards. Similar to (Margeloiu et al., 2021), and as shown by us with CUB, most concepts from our Class-level poker cards models did not apply attribution values to semantically meaningful input features. However, a few concepts were an exception. Namely, the concepts “Four of Clubs” and “Four of Spades” as seen in Figure 3.11 with our independent and sequential models are observed to apply high amounts of attribution to the corresponding semantically meaningful input features. As Class-level poker cards assigns some concepts to a single task label, while others appear for many, the input features the model should use for each concept prediction may be ambiguous. For instance, the concepts Two and Four of Hearts always appear together and therefore the model has no way of separating the pixels for one concept from the other, while the concept “Four of Clubs” may appear with the “Four of Diamonds” and “Four of spades”, or with the “Three of Hearts” and “Five of Diamonds”. In this case, the model has a far better chance of learning the semantically meaningful input features for the concept “Four of Clubs”.

RQ1, and specifically Sub-question 1, cannot be answered with qualitative evaluation techniques. To answer RQ1 we need to quantitatively analyse whether our playing card models are applying feature attribution to semantically meaningful input features. We measured the proportion of feature attribution applied

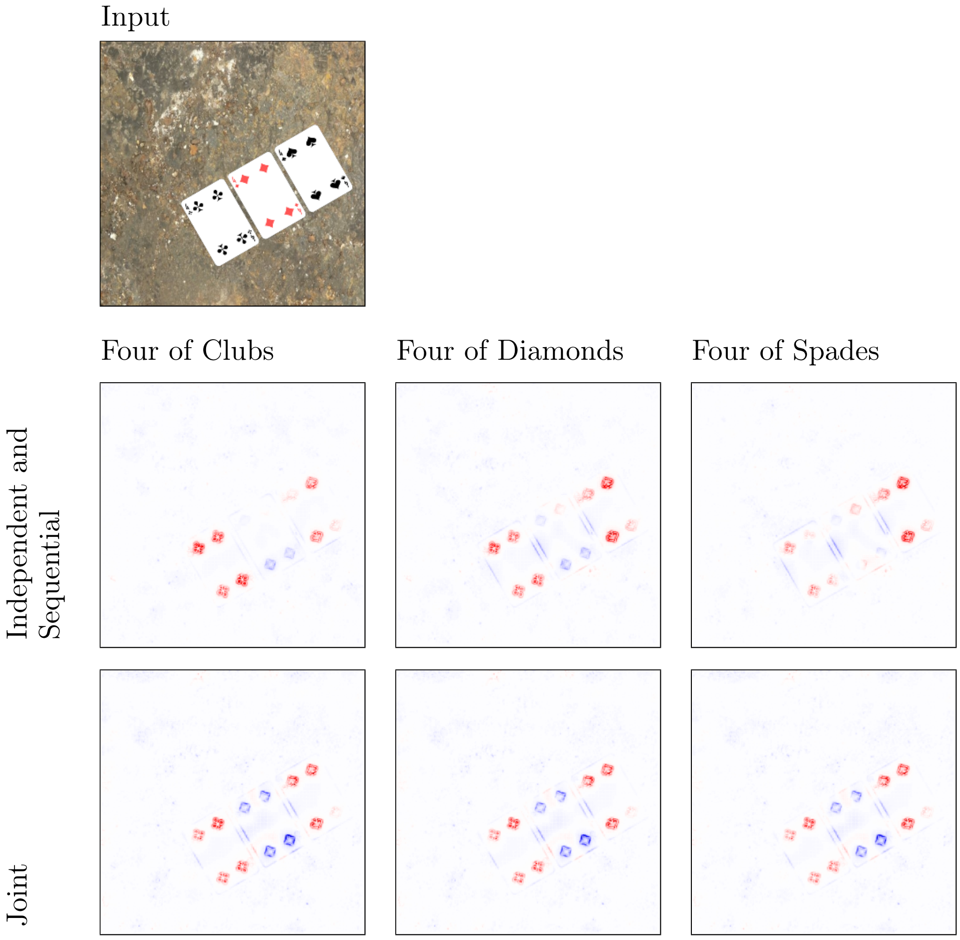


Figure 3.11: For most concepts, such as "Four of Diamonds", the model's saliency maps do not align with semantically meaningful input features. However, for some concepts, like "Four of Clubs" and "Four of Spades", the same CBM uses input features aligned with semantically meaningful input features.

to concepts' visual representations in comparison to the total feature attribution applied to all concept input features. For Playing cards, this is the feature attribution applied to one playing card compared to all three playing cards in a given sample. If the proportion of positive feature attribution is high, and the proportion of negative feature attribution is low, then the model has learned to predict concepts using semantically meaningful input features. We repeated this measurement using the 5 training repeats for each dataset variant and presented

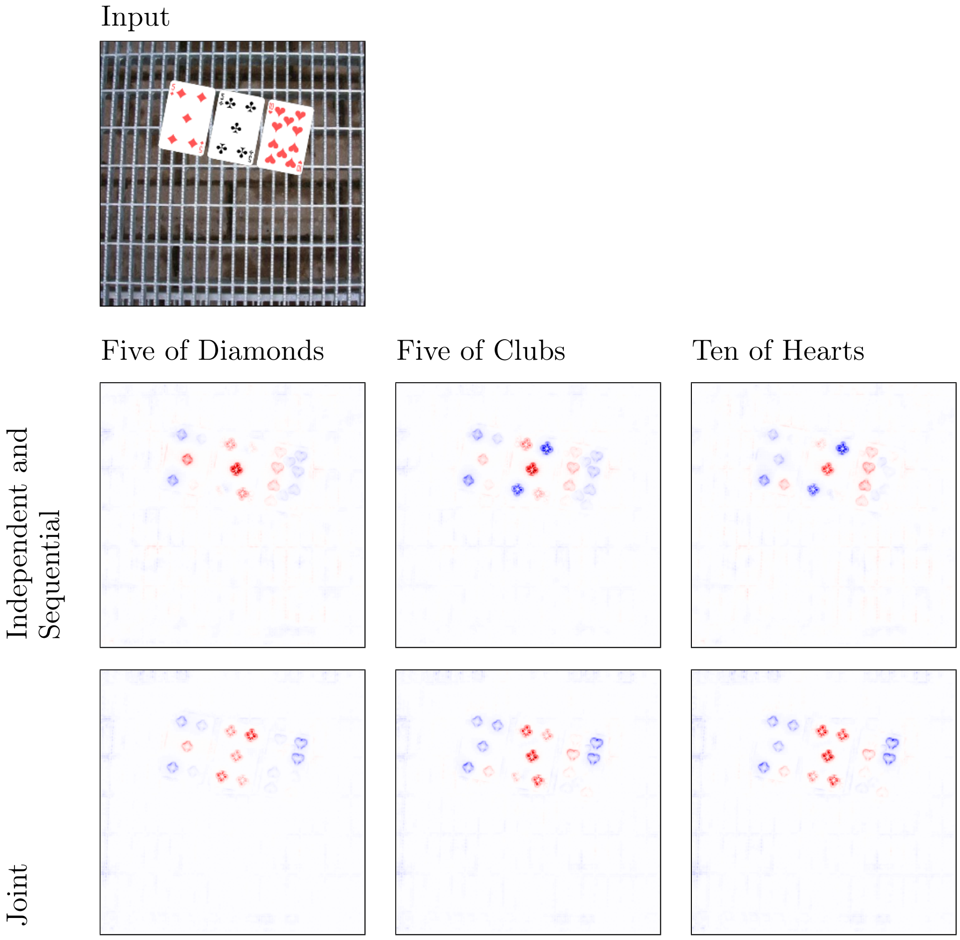


Figure 3.12: Some input features can be used to predict multiple concepts such as the ground truth pixels for the concept "Five of Clubs".

the averaged result.

The plot in Figure 3.13 shows the proportion of feature attribution for our independent and sequential models where each point represents a single concept. There are two distinct clusters, one for the standard DNN and one for both Random cards and Poker cards. Random cards and Poker cards cluster has the highest proportion of positive feature attribution, with most points being between 70% and 80%, while the lowest proportion of negative feature attribution falls between 10% and 30%.

The standard DNN has far less positive feature attribution values and slightly more negative feature attribution values, both around 30% to 40%. As the feature

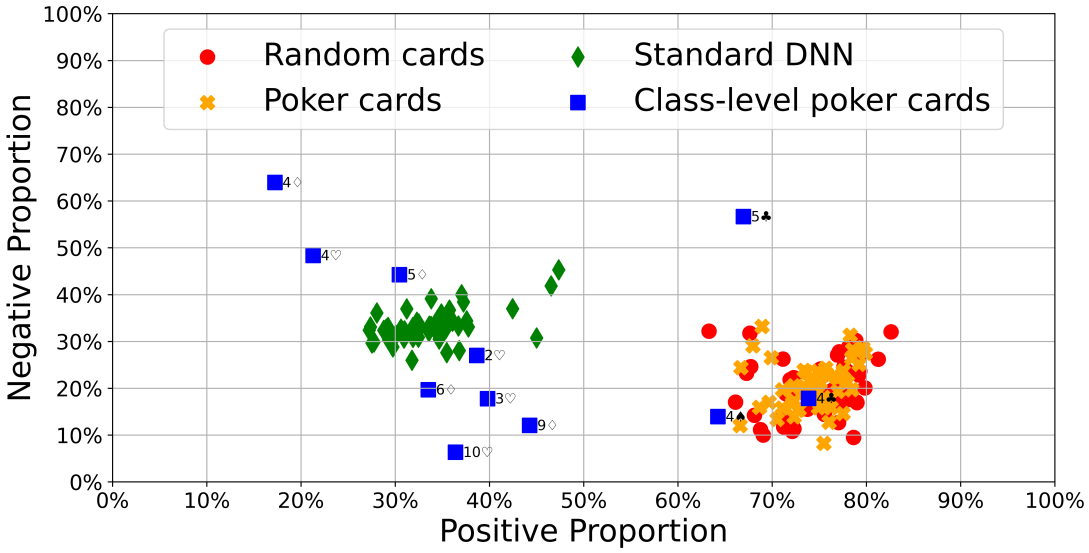


Figure 3.13: Random cards and Poker cards both have a high positive proportion of feature attribution, indicating the models have learned a semantically meaningful mapping from input features to concepts, unlike standard DNN and most concepts for class-level poker cards.

attribution for the standard DNN saliency maps appears distributed over all ground truth concept input features, the proportion being close to 30% is expected as 33% would show feature attribution has been applied evenly to all concept input features. The standard DNN points are positioned close to the baseline Playing cards CBMs will need to pass before we can consider them to have learned to map input features to semantically meaningful concept outputs. Combining this plot with the saliency maps we saw in Figure 3.10 confirms the CBMs have learned to apply feature attribution values to semantically meaningful input features for both Random cards and Poker cards.

The points for Class-level poker cards are not clustered together. Most concepts have a low proportion of positive feature attribution, meaning feature attribution values are not applied to semantically meaningful input features. However, a few concepts, “Four of Spades”, “Four of Clubs” and “Five of Clubs” have a high proportion of positive feature attribution. For the concepts “Four of Spades” and

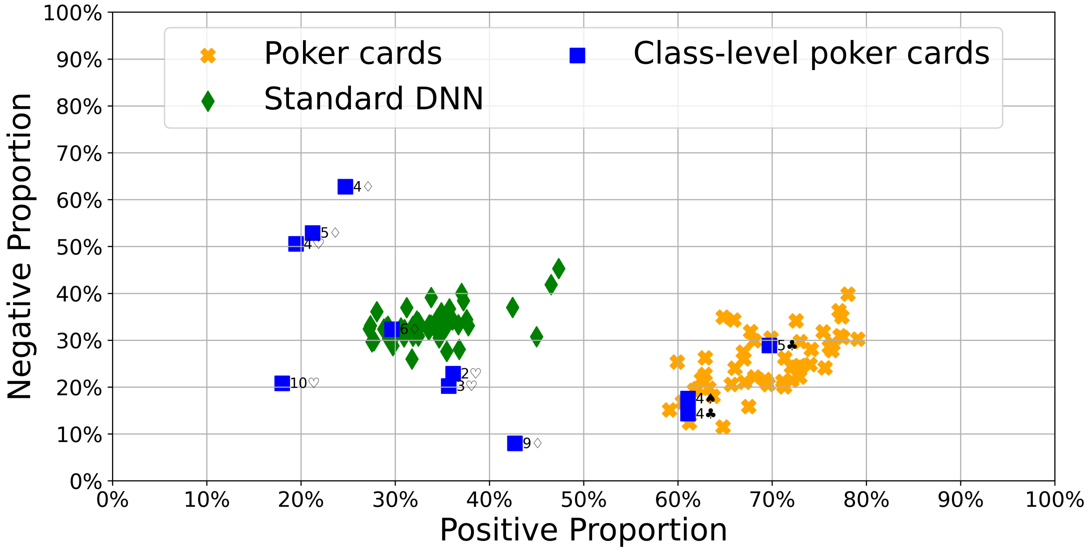


Figure 3.14: Poker cards has a high positive proportion of feature attribution, indicating the models have learned a semantically meaningful mapping from input features to concepts, unlike standard DNN and most concepts for Class-level poker cards.

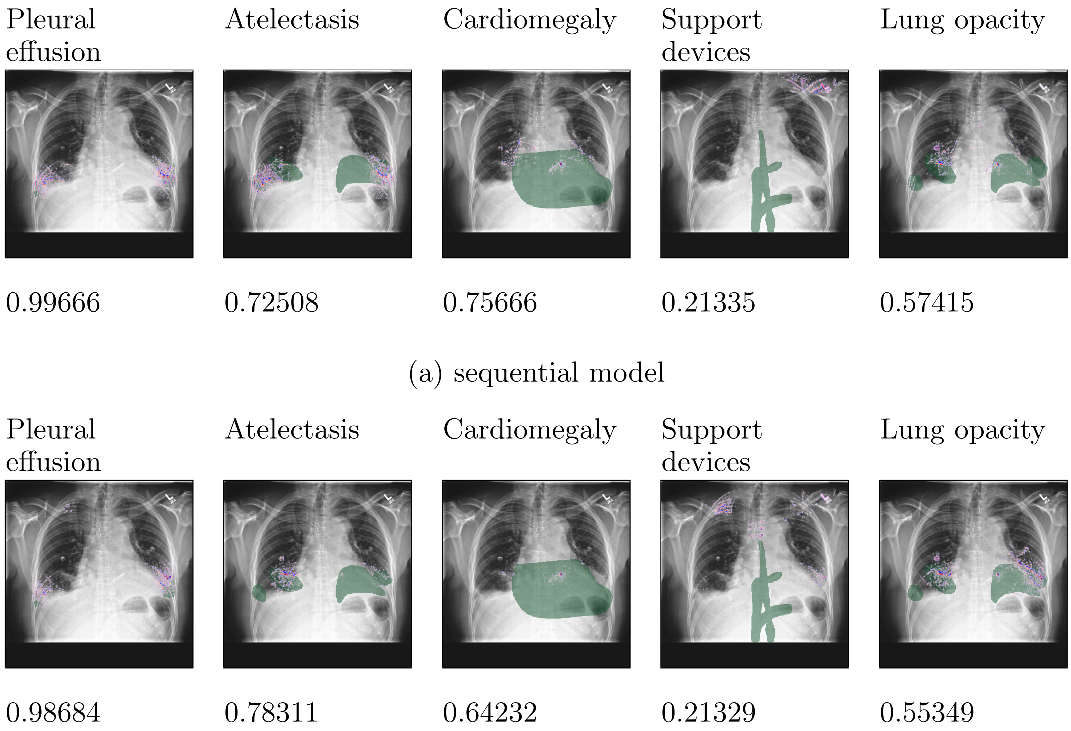
"Four of Clubs", this confirms what we saw in Figure 3.11, that a semantically meaningful concept mapping has been learned. The same cannot be said for the concept "Five of Clubs" as other concepts apply positive feature attribution values to the pixels representing this concept, such as the concepts "Five of Diamonds" and "Ten of Hearts" as seen in Figure 3.12, which inflates the positive proportion of feature attribution seen. Class-level poker cards reveal the challenge of creating a dataset with enough constraints for the model to learn semantically meaningful concept mappings. Even though the concepts "Four of Spades" and "Four of Clubs" show it possible for a CBM to learn semantically meaningful concept mappings with class-level concepts, the consistency in feature attribution proportions with Random cards and Poker cards shows the advantage instance-level concepts can provide.

We repeated the proportion of feature attribution for joint models in Figure 3.14. As with independent and sequential playing card models, our joint models continue to apply a high proportion of feature attribution to semantically meaningful input features for Poker cards, while Class-level poker cards show a high proportion of feature attribution to semantically meaningful input features for the same concepts that received a high proportion in Figure 3.13. As Random cards are only used to train the concept encoder for independent and sequential models, it has not been included in Figure 3.14. The same data is used for the standard DNN across both plots to indicate the baseline when our models no longer use semantically meaningful input features to predict concepts.

3.6.1.3 CheXpert

Unlike a synthetic domain, perfect concept annotations are not guaranteed in a real-world setting. Applying what we have learned from the Playing cards dataset, the question that now stands is does the same hold for a real-world image dataset with similar concept annotation properties? To answer this we have trained CBMs on CheXpert. This dataset contained chest X-ray images with concepts representing visual observations. As uncertain or missing values in the dataset are set to present, some concept annotations will be inaccurate and we may assume there are some additional inaccurate concept annotations caused by the annotations originally being generated using an automated labeller (Irvin et al., 2019). For our results in Figure 3.15, we used our models trained on Instance-level CheXpert and the saliency mapping technique Guided Grad-CAM (Selvaraju et al., 2017) as Grad-CAM techniques have been shown to outperform other XAI techniques with this dataset (Saporta et al., 2022). To validate that the models have mapped concepts to semantically meaningful input features we used ground truth segmentations from (Saporta et al., 2022). These were created by two board-certified radiologists and ensure our conclusions are made w.r.t. expert opinion.

Our results show concepts trained on Instance-level CheXpert can map concepts to semantic input features for models trained with both the independ-

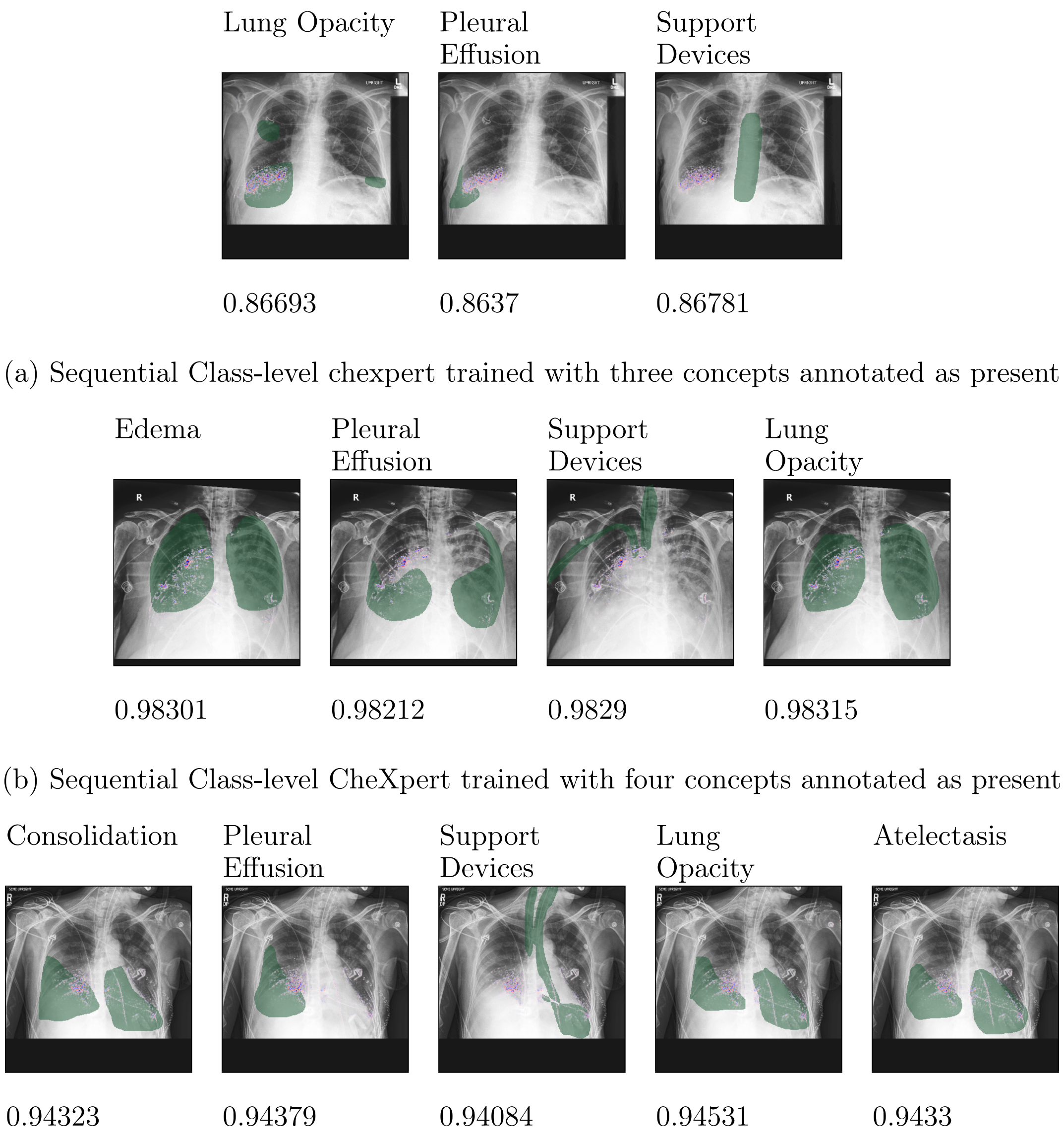


(a) sequential model

(b) Joint model

Figure 3.15: Concept saliency maps for chest X-rays with Instance-level CheXpert shows reasonable localisation of concepts to ground truth segmentations of the input image. The number beneath each saliency map is the concept prediction made by the model where a value of 0.5 or above means the model predicted the concept as present.

ent/sequential method in Figure 3.15a and joint method in Figure 3.15b. The concepts for "lung opacity", "atelectasis" and "pleural effusion" all should be observable with observations in the lung, "cardiomegaly" observed by an enlarged heart, and "support devices" by the observation of an object that is not part of the body (e.g. a pacemaker). From the samples we show, most concepts map to features within the ground truth segmentation such as with the saliency maps for the concepts "atelectasis" and "pleural effusion". The concept "cardiomegaly" is localised to a portion of the segmentation, while "support devices" missed the ground truth segmentation. In the case of "support devices", the model may have



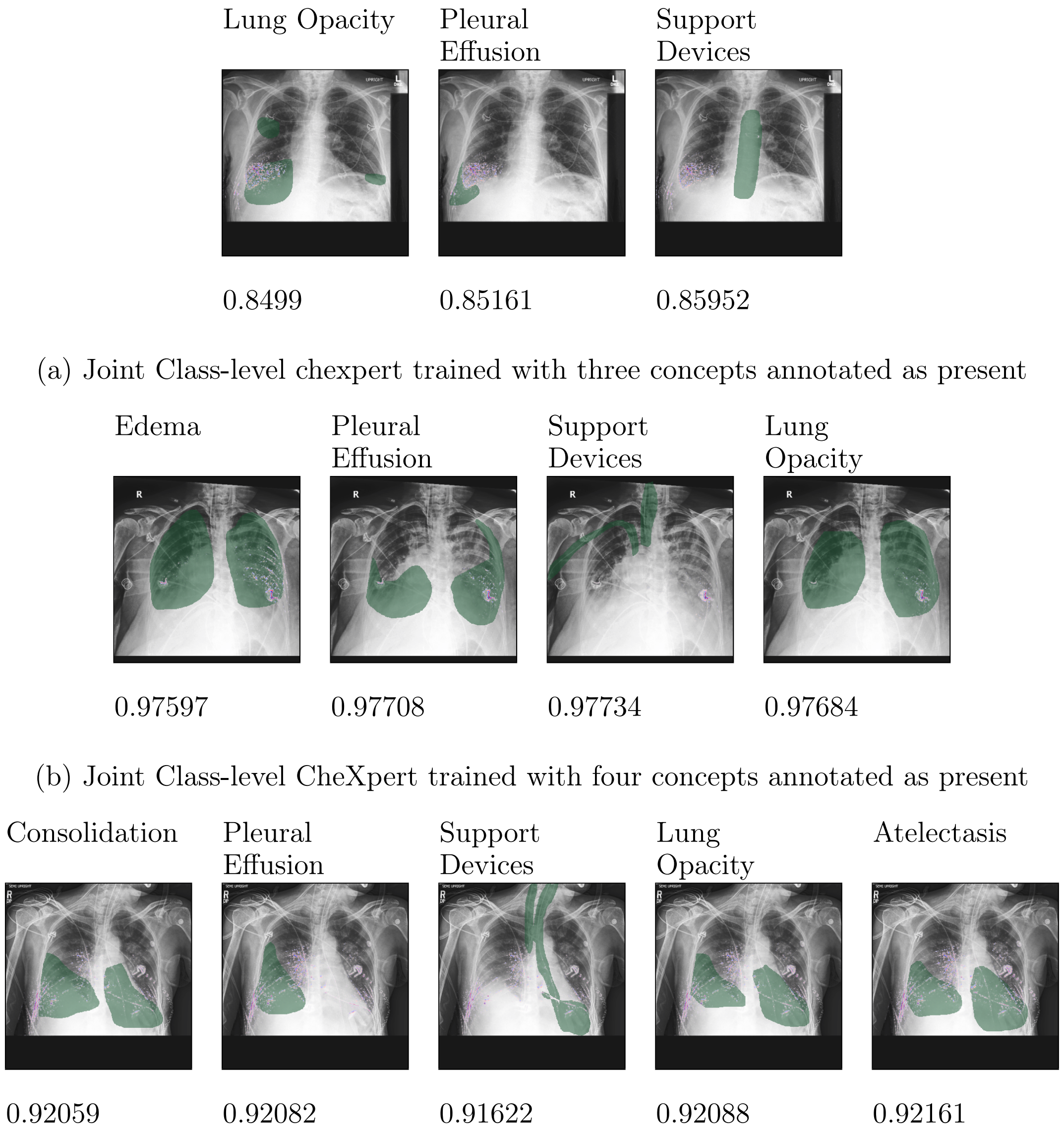
(a) Sequential Class-level chexpert trained with three concepts annotated as present

(b) Sequential Class-level CheXpert trained with four concepts annotated as present

(c) Sequential Class-level chexpert trained with five concepts annotated as present

Figure 3.16: Concept saliency maps for a CBMs trained with the sequential method on Class-level CheXpert (all three versions) shows each model use similar input features to predict all present concepts for a given input image. Concept predictions are beneath each saliency map where 0.5 or above means concept as is predicted as present.

missed the semantic input features as they are hard to spot in this sample compared to a pacemaker which would also be annotated as the same concept. It's also worth pointing out that the concept "support devices" was not predicted as



(a) Joint Class-level chexpert trained with three concepts annotated as present

(b) Joint Class-level CheXpert trained with four concepts annotated as present

(c) Joint Class-level CheXpert trained with five concepts annotated as present

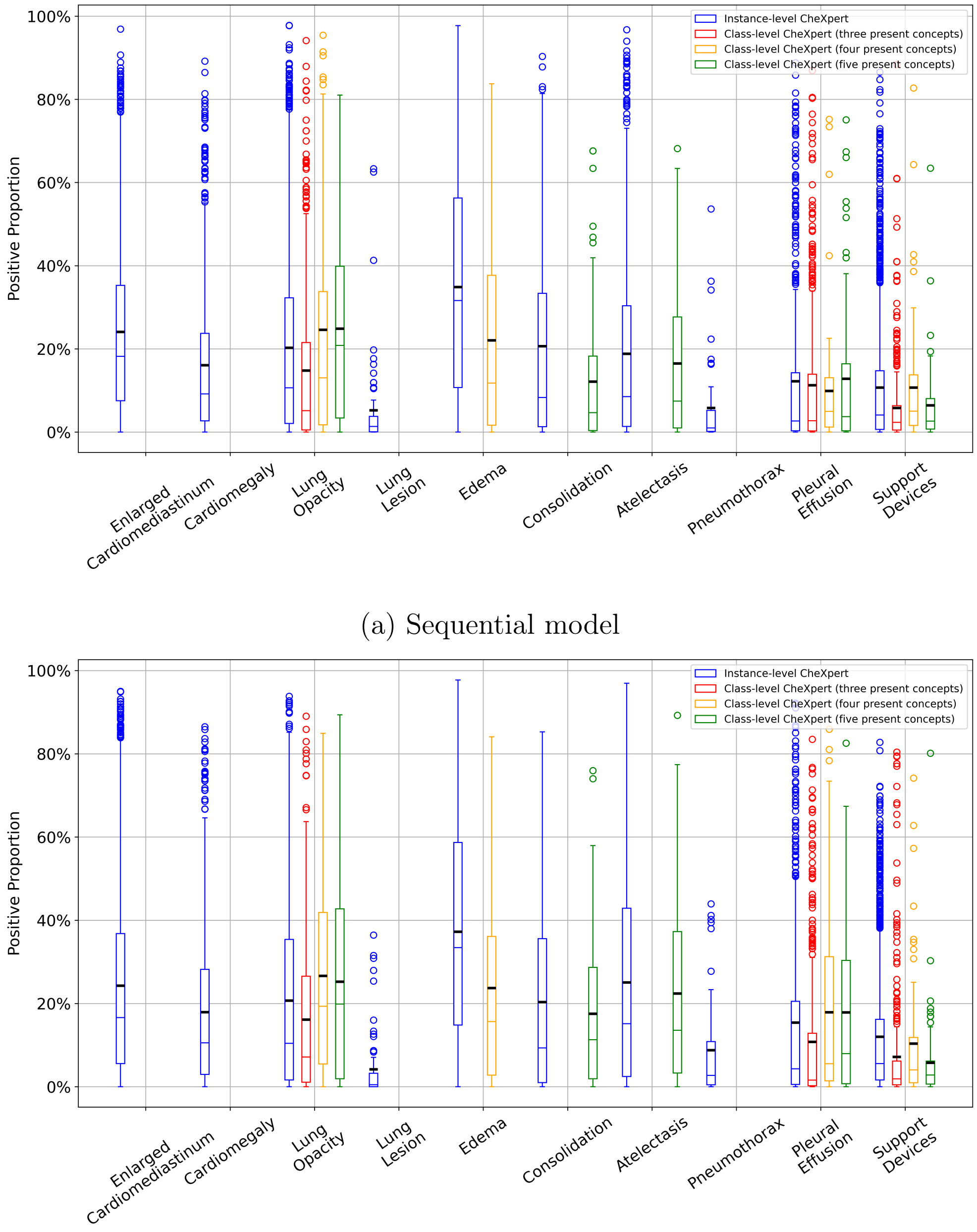
Figure 3.17: Concept saliency maps for CBMs trained with the joint method on Class-level CheXpert (all three versions) shows each model use similar input features to predict all present concepts for a given input image. Concept predictions are beneath each saliency map where 0.5 or above means concept as is predicted as present.

present as seen by the value beneath the saliency map being lower than 0.5. The main takeaway from Figure 3.15 is the saliency maps for Instance-level CheXpert are distinctly different to each other and do not appear to be using the same input features for every concept prediction, unlike Class-level CheXpert as we see in Figure 3.16 and Figure 3.17 where all concept saliency maps highlight similar input features irrespective of the concept being predicted.

The proportions of positive feature attribution in our sequentially trained CheXpert models are shown in Figure 3.18a, and for joint trained models in Figure 3.18b. These figures compare positive feature attribution applied to ground truth segmentations versus the entire image. Since positive and negative feature attributions were nearly identical in proportion, we excluded negative proportions from the plots. Unlike LRP, Guided Grad-CAM assigns feature attribution on a pixel-by-pixel basis which means both positive and negative feature attribution values are placed close together, as seen in the saliency maps for our CheXpert models.

On average, the positive proportion of feature attribution (indicated by black lines in the figures) tops out just below 40% for Instance-level CheXpert and between 20% to 30% for Class-level CheXpert. For concepts such as “consolidation” and “atelectasis”, Instance-level CheXpert shows a higher alignment between feature attribution and ground truth segmentations than Class-level CheXpert, whereas Class-level CheXpert performs better for the concept “lung opacity”.

The lower proportion of alignment for Class-level models is caused by the same input features being used to predict multiple concepts. For instance, if a model primarily uses pixels associated with “lung opacity” to predict all concepts, the feature attribution for “lung opacity” will result in a high alignment, while other concepts will have a lower alignment of attribution values. Saliency maps for our Class-level CheXpert models confirm that these models rely on the same input features across multiple concepts, indicating the dataset does not constrain the CBM to use semantically meaningful features. In contrast, Instance-level CheXpert does not consistently predict concepts using semantically meaningful features.



(a) Sequential model

(b) Joint model

Figure 3.18: Instance-level CheXpert has more positive feature attribution applied to ground truth segmentations than Class-level CheXpert for most concepts, thus demonstrating models trained on Instance-level CheXpert used semantically meaningful input features to predict concepts more often than models trained on Class-level CheXpert.

Returning to RQ1, class-level CheXpert uses the same input features for all concept predictions, as with our models trained on Class-level poker cards and CUB. Therefore, we conclude that a dataset must include a clear link between input features and concept annotations for a model to predict the presence of concepts based on semantically meaningful input features. Instance-level concept annotations promotes this. Using Instance-level playing cards we demonstrated the scenario where concept annotations perfectly align with ground truth input features. On the other hand, the assumed inaccuracies in CheXpert's ground truth concept annotations emphasise the need for accurate alignment between concept annotations and visually identifiable input features. This also highlights the limitation factor for training CBMs that the dataset is difficult to produce, as previously discussed in Chapter 2.1.

3.6.2 Concept Feature Attribution

So far we have focused on explaining concept encoder predictions and assessing whether CBMs use semantically meaningful input features. Beyond concept predictions, CBMs also predict task labels, which can be evaluated using methods similar to those applied in our input feature attribution analysis. For this evaluation, we used LRP, as we can utilise its conservation property to work out the proportion of contribution for each concept prediction.

These results contribute RQ1 by analysing the requirements for CBMs to predict task labels using semantically meaningful concepts labels. These results specifically answer Sub-question 2 as the three CBM training methods changes whether a models task predictor model part receives ground truth concept value as an input (independent training methods), or concept predictions (sequential and joint training methods).

3.6.2.1 CUB

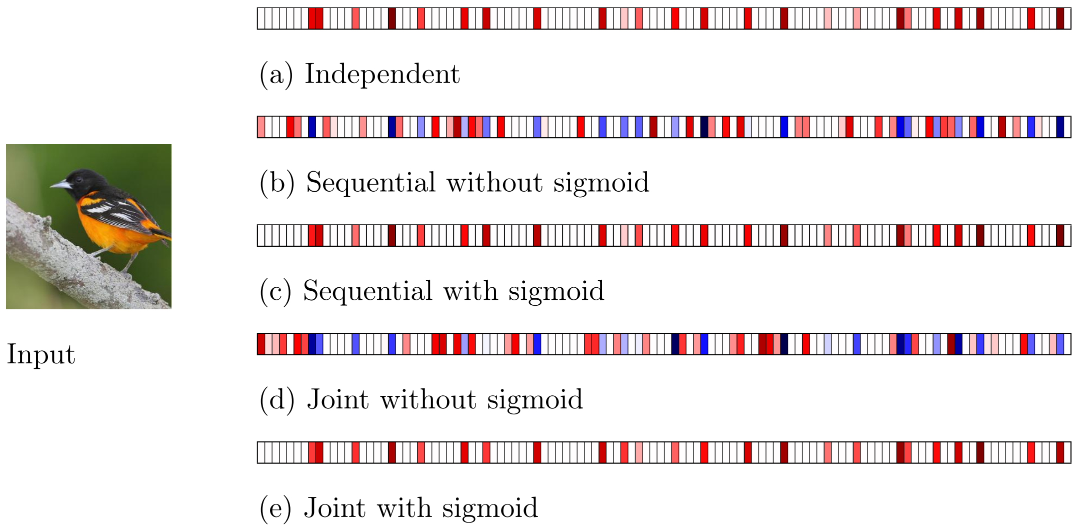
Figure 3.19 shows the saliency maps for our CUB models concept predictions. In this figure, the saliency maps are represented as a segmented line where the leftmost segment corresponds to the first concept, and the rightmost segment corresponds to the last, with concepts ordered by their index.

The results highlight samples for the independent, sequential with sigmoid, and joint with sigmoid models often apply positive feature attribution values to concepts predicted as present, with almost no feature attribution applied to concepts predicted as absent. This indicates that these models rely primarily on the presence of predicted concepts to predict task labels, and applied little weighting to the absence of concepts.

The sequential without sigmoid and joint without sigmoid models both show a different pattern of feature attribution. These models often assign negative attribution values to concepts predicted as present and positive values to concepts predicted as not present. This suggests the learned mapping of concept predictions to task labels uses the absence of concepts when predicting task labels, instead of the presence of concepts. Feature attribution values are not flipped for all samples in the test dataset, although it occurs most of the time.

The key difference between these two groups of models is the use of the sigmoid function. Models with a sigmoid function between the model parts have feature attribution patterns aligned with concept presence. Models without a sigmoid function use concept absence in their task label predictions.

As previously discussed, LRP enables us to calculate the contribution of each predicted concept w.r.t. the predicted task label. For the same input as used in Figure 3.19, the top three concepts contributing to the final class predicted with the independent model are as follows: *has_ upperparts_ color::white* at 6.04%, *has_ primary_ color::yellow* at 5.83%, and *has_ tail_ pattern::multi-colored* at 5.39% with a total of 38 concepts contributing to the final classification. By calculat-



Input

(a) Independent

(b) Sequential without sigmoid

(c) Sequential with sigmoid

(d) Joint without sigmoid

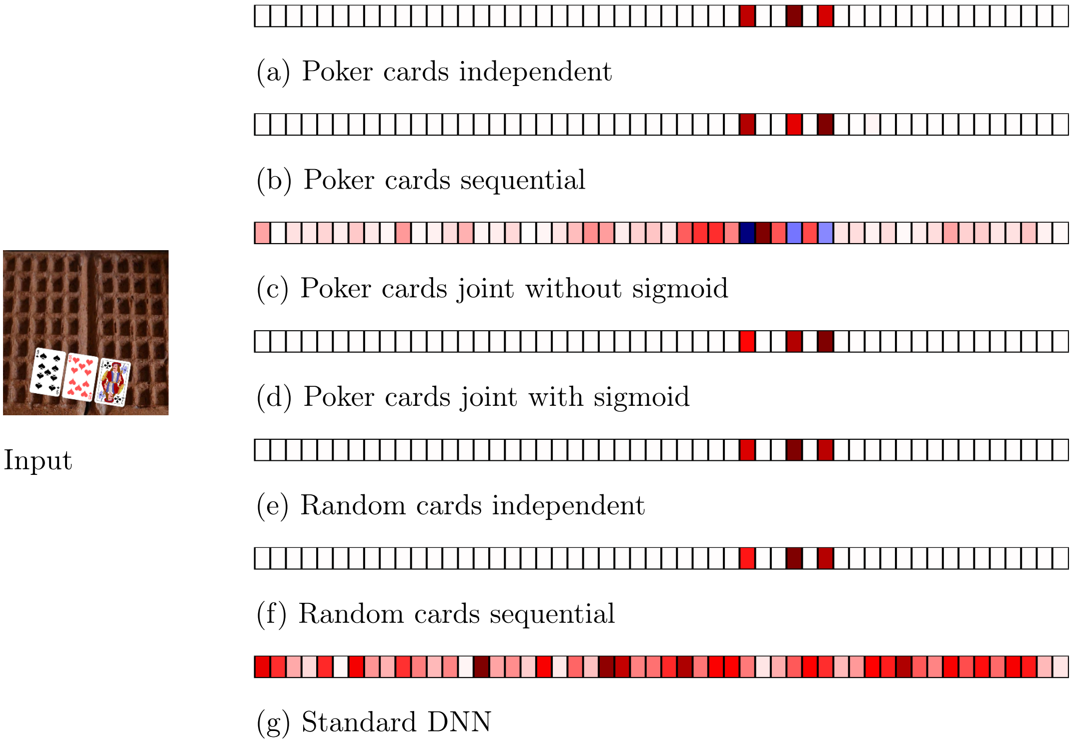
(e) Joint with sigmoid

Figure 3.19: Task label saliency maps for a correctly predicted Baltimore Oriole input. Each vector has 112 segments, one for each concept input. Positive feature attribution values are shown in red and negative attribution values are shown in blue. The independent, sequential with sigmoid and joint with sigmoid models only apply positive attribution values to concept predicted as present. The joint without sigmoid and sequential without sigmoid models apply positive attribution values to concepts predicted as not present and negative attribution values to concepts predicted as present.

ing the concept contributions we are revealing the decision-making process of the task predictor such that a human can take this into their decision-making when interacting with a CBM.

3.6.2.2 Playing cards

Task prediction saliency maps for Playing cards continue the same story as CUB where if a sigmoid function is part of the CBM then feature attribution is only applied to the concepts predicted as present. If a sigmoid function is not part of the model then concepts predicted as present receive a negative feature attribution while concepts not predicted as present receive positive feature attribution values.



Input

(a) Poker cards independent

(b) Poker cards sequential

(c) Poker cards joint without sigmoid

(d) Poker cards joint with sigmoid

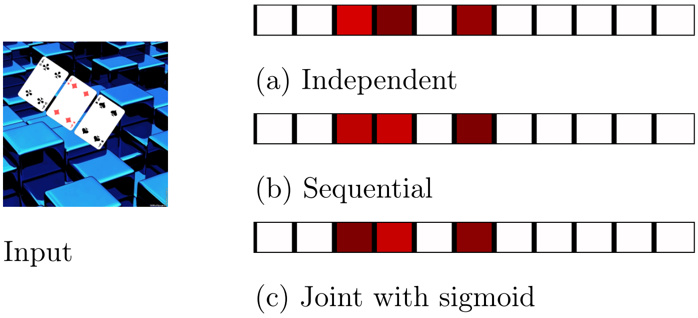
(e) Random cards independent

(f) Random cards sequential

(g) Standard DNN

Figure 3.20: Task predictor saliency maps for models trained on playing cards. Each vector has 52 segments, one for each concept input. Positive feature attribution values are shown in red and negative attribution values are shown in blue. Models that use a sigmoid function between the two models parts apply attribution values to concepts predicted as present, while models without a sigmoid function applies negative attribution values to concept predicted as present and positive attribution values to concepts predicted as not present. The standard DNN does not apply attribution values to any concept in particular as it was trained without concept loss.

These observations can be seen in Figure 3.20 and Figure 3.21. For our Playing card models, we use a sigmoid function in all models. The only exception is our joint models which have a version with a sigmoid function, and a version without one. As the saliency maps for models with and without sigmoid are consistent across models trained on different datasets, it is clear a sigmoid activation should be used to ensure a positive attribution values are applied to concepts predicted



Input

(a) Independent

(b) Sequential

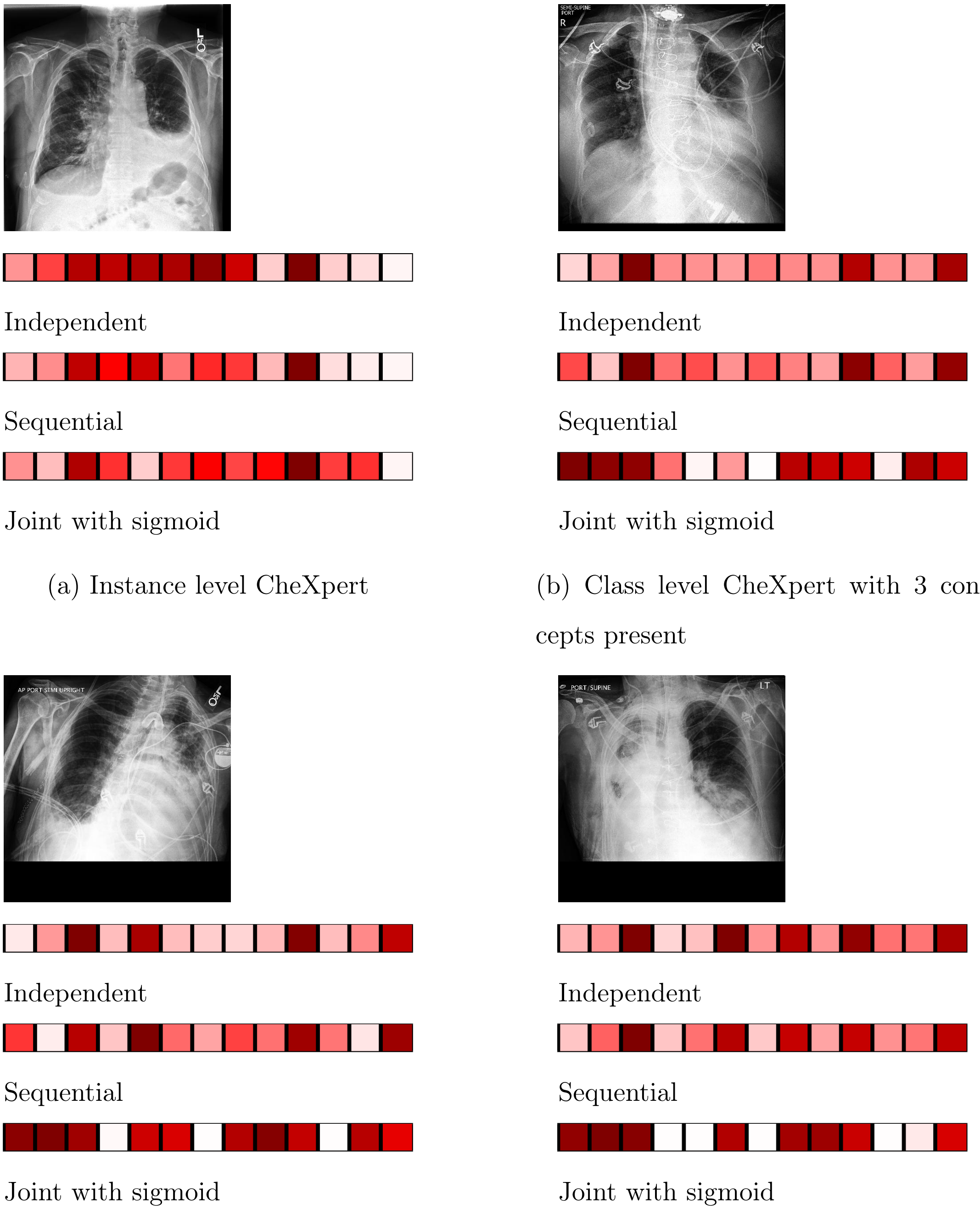
(c) Joint with sigmoid

Figure 3.21: Task predictor saliency maps for models trained on class-level poker cards. Each vector has 11 segments, one for each concept input. Positive feature attribution values are shown in red and negative attribution values are shown in blue. As all models use a sigmoid function between the two models parts they all applied attribution values to concepts predicted as present.

as present. As the standard DNN did not use concept loss during training, and therefore was left to set the weights of the concept vector to best fit the task output. Feature attribution values are distributed over the entire concept vector for this model.

3.6.2.3 CheXpert

The CheXpert task predictor saliency maps continue some of the same story as CUB and Playing Cards. As with these datasets, only positive feature attribution values are applied to concepts. All CheXpert models use sigmoid functions. However, CheXpert differs in that nearly all concepts receive a higher-than-expected feature attribution value as seen by the saliency maps showing most segments as red. Concepts predicted as present have the highest attribution values (darker red segmentations), with concepts predicted as not present lower values (light red segmentations). The lower distinction of feature attribution may be explained by the lower model accuracy of the CheXpert models.



(a) Instance level CheXpert

(b) Class level CheXpert with 3 concepts present

(c) Class level CheXpert with 4 concepts present

(d) Class level CheXpert with 5 concepts present

Figure 3.22: Task prediction saliency maps for CheXpert apply positive feature attribution values to most concepts in the concept vector, although the highest attribution values are those that are also predicted as present.

3.6.2.4 Concept Feature Attribution Alignment

To quantitatively measure task prediction feature attribution, we used the Intersection over Union (IoU) metric, also known as the Jaccard similarity coefficient. Like the proportion of feature attribution metric applied to the concept encoder, IoU evaluates how closely feature attribution values align to ground truth concepts annotation from the dataset. We have defined how we're using IoU in Equation 3.3, where A represents ground truth concepts annotated as present and B represents predicted concepts with a feature attribution value between the maximum value observed in a sample and 10% of that maximum. The 10% threshold ensures that only concepts contributing meaningfully to task predictions are included while excluding those with very low attribution.

$$IoU = \frac{|A \cap B|}{|A \cup B|} \tag{3.3}$$

IoU values are displayed in Table 3.9 for all models trained with CUB, Playing cards and CheXpert using a sigmoid function. As we did not train models using the joint training method with the Random cards dataset, this value is blank in the table. The values in the table are averaged by performing the IoU calculation with each set of model weights from the repeated training runs for each dataset. The highest IoU values are for models trained on Playing cards with only small difference between Random cards and Poker cards. CUB also has a high IoU value while CheXpert models has low IoU values. As CheXpert is a challenging dataset for the models to learn (the concept accuracy for all CheXpert models does not exceed 76%), a low IoU value is to be expected. Comparing the different training methods, joint models generally have a lower IoU, apart from CheXpert models. Independent models almost consistently have the highest IoU values, but this is only narrowly ahead of sequential models.

In addition to our CBMs that includes a sigmoid function, we evaluated our models without a sigmoid function: our standard DNNs trained on Poker cards,

Training method	CUB	Random cards	Poker cards	Class-level poker cards	Instance-level CheXpert	Class-level CheXpert with 3 present concepts	Class-level CheXpert with 4 present concepts	Class-level CheXpert with 5 present concepts
Independent	**0.849**	**0.991**	**0.988**	**1.0**	0.155	0.122	0.177	0.21
Sequential	0.841	0.935	0.969	0.966	**0.168**	0.12	0.168	0.199
Joint	0.781	-	0.887	**1.0**	0.164	**0.15**	**0.219**	**0.261**

Table 3.9: IoU values for models and datasets. All models have a sigmoid function between the model parts. The independent training method achieves the highest IoU value for models trained on datasets with instance-level concepts.

sequential models trained on CUB, and joint models trained on both CUB and Poker cards. These configurations achieved IoU values of 0.057, 0.102, 0.248, and 0.008, respectively. These results are far lower than those of models that include a sigmoid function, corroborating our earlier observations from saliency map comparisons. Although the non-zero IoU values indicate some overlap in feature attribution applied to predicted concepts, the overall low scores suggest that such occurrences are rare.

As the only change between our models is regarding how they are trained, we can answer Sub-question 2. The best training method to train a CBM task predictor is with the independent method, and a sigmoid function should not be used with the CBM. As independent models are trained using ground truth values that are either 0 or 1, the models have a clearer training signal. The sequential and joint training methods will provide concept values between 0 and 1. Training models on ground truth concept values also ensures all concepts are accurate, which may not be the case if the task predictor is trained using the output of the concept encoder.

3.7 Discussion

The findings presented in this chapter address RQ1: "How can we train a CBM to map semantically meaningful input features to concepts, and semantically meaningful concept predictions to task labels?".

Through our analysis, we demonstrate that it is possible to train CBMs to predict concepts using semantically meaningful input features, and task labels using concept predictions aligned with ground truth concept annotations. This discussion expands RQ1 by addressing the two sub-questions:

1. What dataset configurations, in particular w.r.t. concept annotations, are required to train CBMs to learn semantically meaningful mappings from input features to concept predictions, and from concept predictions to predicted task labels?
2. What is the most effective CBMs training method?

3.7.1 Dataset Configurations Requirements

Starting with training a CBM to map semantically meaningful input features to concept predictions, the configuration and annotations of the dataset play a crucial role. Our experiments revealed two primary factors that influence the success of this mapping:

- Concept variance: The variance in which concepts appear together in the training data significantly affects a CBM's ability to learn semantically meaningful mappings from input features to concept predictions. In this chapter, we highlighted this with class-level vs. instance-level concept annotations. Class-level concepts often result in multiple concepts appearing together, sometimes exclusively, across all training samples. CUB is a clear example of this, where the model learned to use input features of the entire bird for all concept predictions, as these were the most consistent input features throughout training. To promote the learning of semantically meaningful concept representations, concepts should vary across samples so that the model learns to distinguish each concept individually.

- Accurate concept annotations: Ensuring that concept annotations correctly reflect the presence of concept semantically meaningful input features in the input image is foundational to training a model to learn semantically meaningful mappings. Without precise annotations that reflect visual features in the input, the model will struggle to learn the intended mappings as it will lack a strong training single from the data. This is understood by reminding ourselves these models are only provided with concept annotations for supervision during training. As with concept correlation, class-level concept annotations make accurate concept annotations difficult to achieve, especially when a dataset includes real-world images that cannot ensure the image subject is completely visible.

For a CBM to learn a semantically meaningful mapping from concept predictions to task labels the dataset has limited impact. Overall all of the datasets in our experiments produced an accurate task predictor model part. However, a key consideration to highlight is concept vectors have to be distinct to ensure task labels can be accurately learned. If two task classes have the same concept vector then the task predictor will not be able to distinguish between them.

Overall these factors answer Sub-questions 1. To train a model to learn semantically meaningful mappings from input features to concept predictions the dataset has to configure concepts to minimise excessive concept correlation and ensure present concepts are accompanied with a visual representation. This is easier to achieve with instance-level concepts and could be considered impractical for class-level concepts.

3.7.2 Effective Training Methods

Our analysis of training methods for CBMs found that although all training methods (independent, sequential and joint) successfully trained CBMs, the independent method resulted in a model that most accurately aligned feature attribution from task prediction to ground truth concept values. This training method avoids balancing task and concept accuracy, as with the joint training method, or relies on concept predictions that may be inaccurate. The independent training method provides the task predictor with a clear training signal of which concepts should map to each task label.

Furthermore, we identified the use of a sigmoid function in a model enhances the alignment of feature attribution values with ground truth concepts. This layer in the model has the potential to improve interpretability by ensuring the model uses the presence of concepts to predict task labels.

These findings answer Sub-questions 2. The most effective method to train a CBM is the independent training method. In addition, placing a sigmoid function between the two model parts can confine a CBM to use the presence of concepts in its decision-making process.

3.8 Limitations

In the analysis of CBMs conducted in this chapter we have identified several limitations.

3.8.1 Datasets

We evaluated CBMs using three datasets: CUB, Playing cards, and CheXpert. As discussed in Section 3.5.1, these datasets cover a range of configurations, enabling us to analyse the effect of dataset attributes on model performance. However, we have identified two limitations from the datasets:

Firstly, expanding the evaluation to include additional datasets would allow for a more comprehensive analysis of concept correlation and concept annotation quality. For instance, additional datasets would increase the number of variations of concept correlation in our analysis and thus could provide increased insights into the generalisability of our findings.

Secondly, While our models trained on CheXpert performed inline to those in the literature (Saporta et al., 2022), they showed significantly lower accuracy compared to models trained on CUB and Playing cards. This difference in model accuracy highlights challenges associated with training CBMs on real-world datasets. Evaluating additional real-world datasets could provide further insights into the viability of CBMs beyond synthetic settings.

3.8.2 Methods

The methods used in this chapter are all feature attribution techniques. Despite these revealing which input features are used for concept and task predictions, their local explanation nature means we cannot evaluate how model predictions change with input manipulation or other shifts in images. We are restricted to measuring concept attribution with the samples available in the dataset, which may not cover all situations observed in the real-world.

3.9 Summary

This chapter evaluates CBMs through the application of XAI techniques. Specifically, we analyse both the concept encoder and task predictor, focusing on how feature attribution values are distributed across input features from concept predictions, and concepts from task label predictions.

In our evaluation, we compared how feature attribution aligns with ground truth concept segmentations from the dataset. Beginning with the concept encoder, we investigated how concepts are predicted based on the presence of semantically meaningful input features. We trained CBMs on three datasets: CUB, Playing cards, and CheXpert.

Models trained on datasets with class-level concept annotations (CUB, Class-level Poker cards, and Class-level CheXpert) applied feature attribution values to semantically irrelevant input features, for example, across the entire bird in each CUB sample. We argue this is due to different issues from each dataset: CUB contains concept annotations that often fail to represent the visual content of the images, while concept annotations in Class-level Poker cards and Class-level CheXpert have a high inter-concept correlation, making it difficult for the model to distinguish which input features correspond to individual concepts. In contrast, models trained on datasets with instance-level concept annotations (Random cards, Poker cards, and Instance-level CheXpert) demonstrated strong alignment between concept predictions and semantically meaningful regions within sample images. These datasets provide more accurate annotations that directly reflect what is visually present in each sample. Random cards and Poker cards showed near-perfect alignment, likely due to minimal annotation noise, while alignment in Instance-level CheXpert was slightly lower, reflecting some inaccuracies in the concept annotations, expected in a non-synthetic dataset. These differences between models trained on different datasets were consistent across both individual saliency map results and the aggregated results.

We achieved training CBMs to predict concepts using semantically meaningful input features by ensuring concept annotations were accurate and ensuring a high correlation between concepts in the dataset was avoided. Mapping input features to semantically meaningful concepts does not depend on the use of class or instance-level concepts. Instead, it relies on ensuring that if a concept is annotated as present, it is also visually identifiable in the corresponding image. However, our findings demonstrate that achieving such mappings is more straightforward with instance-level concept annotations.

Predicting concepts using semantically meaningful input features benefits inherent interpretability as we can be sure the model is predicting concepts for the right reasons. If a CBM makes concept predictions using input features with the same meaning, we can argue the model will be easier to build trust with as concept predictions will use the expected input features from a human perspective.

Saliency maps from the predicted task label back to the concept vector show CBMs are capable of applying high feature attribution values to concepts aligned with the ground truth concept annotations. We analysed feature attribution and found CBMs should be trained using the independent training method, or where that is not suitable, use a sigmoid function between the concept encoder and task predictor. By doing so we can maximise the alignment of positive feature attribution values to ground truth concept labels annotated as present. We also demonstrate the ability to calculate proportional concept contributions to task labels.

Chapter 4

Robust Concept Representations

4.1 Introduction

If we can train CBMs to learn semantically meaningful input features by confining the models using training data with adequate concept annotations, we may also assume these models learn concepts to be disentangled such that one concept cannot be used to predict another concept, or the removal of unrelated input features to have minimal impact on the prediction of a concept. Essentially we intend CBMs to learn to predict concepts as independent outputs and resilient to the removal or addition of unrelated input features, or combinations of concepts not seen during training. For instance, if the training data often includes pacemakers alongside heart conditions, we wouldn't want the model to predict a heart condition solely because it detected a pacemaker. Doing so could lead to inaccurate detection of undiagnosed heart issues.

In this chapter we analyse CBMs w.r.t. the information encoded within concepts and how susceptible concepts are to changing input features. In Chapter 3 we primarily use local explanations. These are limited in their capability as they cannot tell us about a DNN as a whole (Arrieta et al., 2019), and reliance solely on feature attribution techniques for evaluation can be misleading (Adebayo et al., 2018; Sixt et al., 2020), or on a single metric instead of getting a consensus from multiple metrics to get a complete picture. We have analysed CBMs with additional evaluation metrics to obtain a comprehensive understanding of how they encode concept representations.

This chapter answers **RQ2** ("*How does the relationship between concepts and* *input features in the training dataset influence the information encoded in learned concepts and the model's reliance on input features for predicting those concepts?*") which can be broken-down into the following sub-questions:

1. How does the configuration of concepts in the training dataset affect information leakage of learned concepts?
2. How does the configuration of concepts in the training dataset affect input feature dependence?

By answering RQ2 we make the following contributions:

- **RC4**: We perform an in-depth evaluation of CBMs revealing CBMs can be trained to minimise the encoding of extraneous information in concept representations, and concepts can be resilient to irrelevant input feature alterations. We demonstrate that CBMs generally learn underlying concept correlations present in the training data.
- **RC5**: We conclude that two factors are critical for CBMs to learn semantically meaningful input features: (i) accuracy of concept annotations and (ii) high variability in the combinations of concepts co-occurring, that is, each concept in a dataset should appear alongside a variety of others to help the model distinguish between them.

This chapter contains work introduced in our paper "*Can we Constrain Concept Bottleneck Models to Learn Semantically Meaningful Input Features?*"

Below is a list of acronyms and their meanings that are predominantly used in this chapter.

CRA Concept Removal Accuracy

DCI Disentanglement, Completeness and Informativeness

MNIST Modified National Institute of Standards and Technology

MSE Mean Squared Error

OIS Oracle Impurity Score

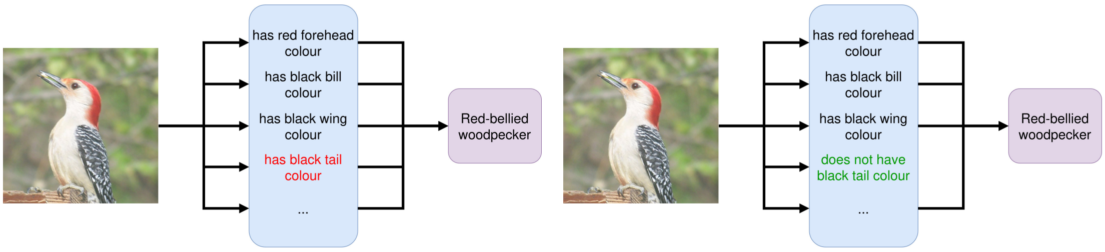
SSIM Structural Similarity Index

4.2 Motivation

The motivation of the work in this chapter is to understand how CBMs encode concept representations they learn from their training data. We approach this from two angles: (1) understanding what information is encoded in concept outputs, and (2) the sensitivity concept predictions are to modified input features.

As introduced in Section 2.1.5, metrics to analyse CBMs without feature attribution measure either *information leakage* (Mahinpei et al., 2021) or *concept feature sensitivity*. We have separated these two measurements by defining information leakage as the degree of which additional information is encoded in concept outputs than is required to predict the concepts themselves, and concept feature sensitivity as how reliant concept predictions are to the presence of input features other than those for the concepts themselves (i.e. are concepts only reliant to the presence of their respective semantically meaningful input features). We expand on these classes of metrics in Section 4.3 and Section 4.4.

Information leakage originates from *disentanglement* metrics (Bengio et al., 2013), where it's generally desired for concepts to only encode information that is required to predict themselves. With CBMs we can hypothesise the worst case scenario would be when all concepts can be predicted from just a single concept, or for the task label can only be predicted if all relevant concepts are present together. Both situations suggest the model has learned that certain concepts always co-occur, resulting in task class predictions with little to no variance in



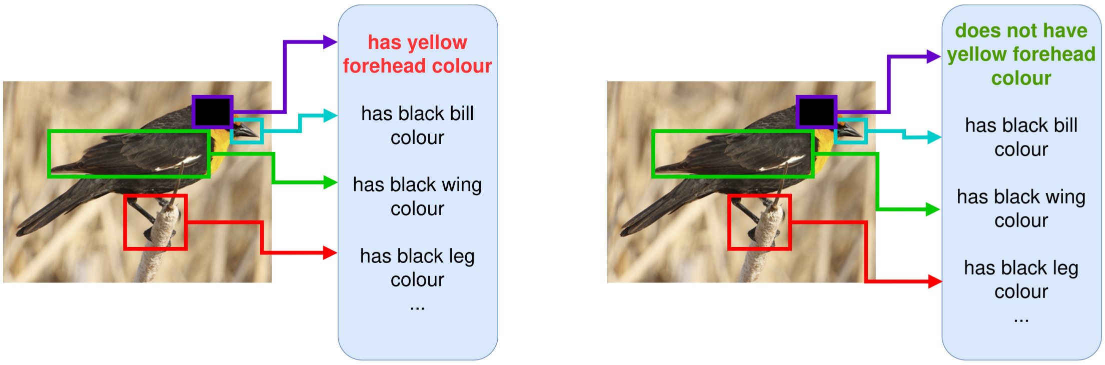
(a) Example of poor information leakage which may result in a model that can only predict a downstream class if all relevant concepts are present.

(b) Example of good information leakage which may result in a model that can predict concept, even if some concepts are missing.

Figure 4.1: Example of good and poor information leakage.

their concept vectors. This is illustrated in Figure 4.1a where the model can only make an accurate task prediction if all concepts are predicted as present, even if some of those predictions are incorrect given the concepts visible in the input image. On the other hand, the best-case scenario would have no extra information encoded into each concept which in turn would mean the decision-making process for task label predictions should rely on a combination of present and not present concepts with the model still able to reason correctly even the set of present concepts is not complete (e.g. a model may be able to predict the type of bird despite not having a present concept prediction for the tail). This ideal case is illustrated in Figure 4.1b. Overall, by analysing information leakage we aim to understand if the concept representations a CBM has learned encodes extra information than is required to accurately predict the concepts themselves (Mahinpei et al., 2021).

Similar to concept predictions from semantically meaningful input features discussed in Chapter 3, another desired property of CBMs is for concept encoders to learn how concepts are *spatially localised* (Raman et al., 2024). This is such that concepts are predicted using semantically meaningful input features and are not modified by the addition or removal of unrelated input features. Depending on how a CBM is deployed the input to the model may have, or may include different



(a) Example of a model with poor concept feature sensitivity. The concept for forehead is predicted as present despite the black mask over the corresponding input features.

(b) Example of a model with good concept feature sensitivity. The concept for forehead is not predicted as present with a mask over the corresponding input features.

Figure 4.2: Example of good and poor concept feature sensitivity.

concept combinations not seen during training. Taking bird identification as an example, an image of a bird may crop out bird parts, the bird may be orientated such that some bird parts are hidden, or the bird may be behind an object. If a CBM is resilient to these kinds of inputs then concepts should still be predicted with high confidence only if they are visible in the input. However, if a CBM is not resilient then some concept predictions may change despite remaining identifiable in the input image. We have illustrated these cases in Figure 4.2 whereas in Figure 4.2a the concept for the forehead is still predicted as present despite a mask over the corresponding input features as an example of poor feature sensitivity. In Figure 4.2b the concept for the forehead is not predicted as present as an example of good feature sensitivity.

Finally, concluding the analysis in Chapter 3, information leakage analysis and concept feature sensitivity analysis, we have identified several best practices to follow for future CBM research and model training. Impotently these best practices identify the conditions and required attributes in a dataset to ensure concepts are predicted with semantically meaningful input features, minimise information leakage, and reduce undesired feature sensitivity.

4.3 Information Leakage

In the literature, the following metrics have been used to analyse CBMs and similar model types:

Mahinpei et al. (2021) evaluates information leakage with CBMs. Their metric looks at task-blind training such as independent and sequential training. In their paper, concepts required for task prediction were removed from the dataset with their metric measuring the task accuracy after training with the missing concepts added back to the dataset. In their experiment they used the Modified National Institute of Standards and Technology (MNIST) dataset where the model predicted whether a digit was odd or even, with two binary concepts “is four” and “is five”. All “fours” and “fives” digits were removed for training which should have made the task accuracy a random guess (50%) but instead the accuracy they measured was (69%) and thus information leakage occurred. A similar experiment was performed by Margeloiu et al. (2021) with the joint training method and gradually removing concepts from the dataset. As with the task-blind training methods (such as CBM independent training), concept leakage was observed with the joint training method.

Mahinpei et al. (2021) also introduced the idea of *concept purity* which measures whether concept predictions can be used to predict the labels of other concepts. They test this with a model with additional unsupervised outputs to capture information that does not fit into the concept space. They find information leakage still occurs with models trained using task-blind training methods.

Marconato et al. (2022) evaluates models according to the metric DCI (Eastwood and Williams, 2018). They train CBMs with varying amounts of concept supervision and measure the resulting accuracy, alignment (how similar the models and humans' semantic representations are to each other) and explicitness scores (how well a linear regressor fits the concepts in the dataset). To be expected, they observed less entanglement as concept supervision increased during training.

Espinosa Zarlenga et al. (2023) introduces OIS and Niche Impurity Score to measure concept purity. OIS builds on the purity measurement by Mahinpei et al. (2021) and measures inter-concept predictability w.r.t. the expected predictive performance of the dataset. Espinosa Zarlenga et al. (2023) argues their metrics are better suited to CBMs than alternative disentanglement metrics, primarily because other disentanglement metrics assume all entanglement is undesired which ignores correlation in the ground truth concept annotations.

The main idea of OIS is to asses whether concept representations can predict one another in line with what would be expected from the ground truth concept annotations. This helps determine if the relationships of learned concepts reflect the actual relationships in the training datasets.

to compute OIS, the divergence of two matrices is measured:

1. An *oracle matrix* that measures how well ground truth concepts can predict each other.
2. A *purity matrix* that measures how well learned concepts can predict ground truth concepts.

These matrices are created by training a series of helper models which receive either ground truth concepts or predicted concepts and minimise the loss of predicting ground truth concepts. If OIS results in a value of 0 then learned concepts do not encode any more or less information than the ground truth concepts. However, a value of 1 means each learned concept can perfectly predict all other concepts.

While OIS focuses on impurities encoded into single concept representations, Niche Impurity Score considers impurities within subsets of concepts, offering an alternative perspective on concept purity.

4.4 Concept Feature Sensitivity

As discussed in Section 2.1.5 concept feature sensitivity evaluates the degree to which concepts are predicted using input features that are correlated to the occurrence of concepts but are unrelated to their prediction. A high feature sensitivity would mean a concept prediction has a high reliance on input features that should not cause any change to the concept prediction. A low feature sensitivity would mean only the input features representing a concept can cause that concept to be predicted as present or not present.

Heidemann et al. (2023) introduced the metric *CRA* which measures the number of samples for which the model's concept prediction changes from present to not present when the input features for an unrelated concept are removed over the number of all true positive concept predictions. Heidemann et al. (2023) only produced qualitative results with their metric and instead used a second metric, *difference in test accuracy*, for quantitive results. This metric measures how concept accuracy changes when trained on one subset of a dataset (e.g. where concept A and B are both present at the same time), while tested where only one concept is present (e.g. just concept A). Heidemann et al. (2023) found that a high correlation of concept annotations in a dataset may lead a model to use one concept as a proxy to predict others.

Raman et al. (2024) also aimed to measure concept feature sensitivity with a similar metric to CRA which they called *locality masking.* Locality masking also measured how concept accuracy changed when input features for concepts were masked, but extended CRA by masking input features both semantically meaningful (referred to as locality-relevant masking) and irrelevant to a concept (referred to as locality-irrelevant masking). They found masking made little difference to concept prediction accuracy which means their model had learned to use input features unrelated to concepts to maintain accurate concept accuracy.

Overall, these results suggest CBMs are incapable of being sensitive to only the semantically meaningful input features. However, as commonly discussed in the literature, these papers do not show results on a range of datasets. Heidemann et al. (2023) only uses the dataset CUB, while Raman et al. (2024) uses the datasets CUB and COCO (Lin et al., 2014) where concepts are the objects in each scene. Both of these datasets are problematic with CUB using class-level concept annotation, while the concepts in COCO have a high potential of being correlated to the rest of the scene (e.g. a bike on a road). It remains unseen how sensitive a CBMs is to concepts trained on a wider range of datasets with different configurations of concept annotations.

4.5 Methods

This chapter answers RQ2: "How does the configuration of concepts in the training dataset affect encoded information in learned concepts, and input feature dependence for concept predictions?" This question builds on RQ1 answered in Chapter 3. RQ2 is focused on the learned representations of concepts by CBMs.

To address RQ2 we used the datasets introduced in Section 3.5.1. Namely we evaluated CBMs trained on the CUB, Playing cards, and CheXpert datasets. This gives us the same variance in dataset configurations to establish how the dataset affects learned concept representations.

To assess information leakage we used the OIS metric, implemented via helper models formed of a two-layer ReLU multi-layer perceptron with 32 activations in the hidden layer. Each helper model was trained on a single concept where the input was the concept value (between 0 and 1) and predicted the values of all other concepts. The helper models are trained to minimise the loss of predicting all other concept values. We repeated the process and averaged individual OIS results to remove any run-to-run inconsistency.

The OIS metric was chosen because it captures inter-concept predictability within the dataset, while also capturing if correlations are unavoidable as the correlations are present in the dataset. OIS also disentangles inter-concept correlations within the dataset from inter-concept correlations observed in concept predictions, which allows us to analyse these factors separately in our conclusions.

For feature sensitivity, we used the CRA metric to measure the impact of masking semantically meaningful, and semantically irrelevant input features associated with a given concept on predictions of other concepts. For each concept(s) that was masked in the input, we measured the change from present to not present predictions for all other concepts.

We used two masking types based on those introduced by Raman et al. (2024): *single-concept masking*, where only input features relevant to a single concept are removed; and *multi-concept masking*, where input features for all but one concept are removed. By using the two making types we can evaluate whether a model is sensitive to both the removal of semantically meaningful input features, and the removal of irrelevant input features.

Before discussing our results, it is important to understand the correlation between concepts within the datasets. To this end, we show the Pearson correlation coefficient for all pairs of concepts in each dataset and the dataset variation used for training concept encoders. The results are visualised as matrices, where the x and y axes correspond to the respective concept pairs.

Starting with CUB in Figure 4.3, the correlation between a concept and itself (diagonal elements) has an average correlation of 1 and off-diagonal elements correlation averaging to 0.014. This appears to show a dataset with minimal correlation between concepts. However, inspecting smaller groups of concept pairs reveals additional smaller diagonals of highly correlated concepts that appear in a

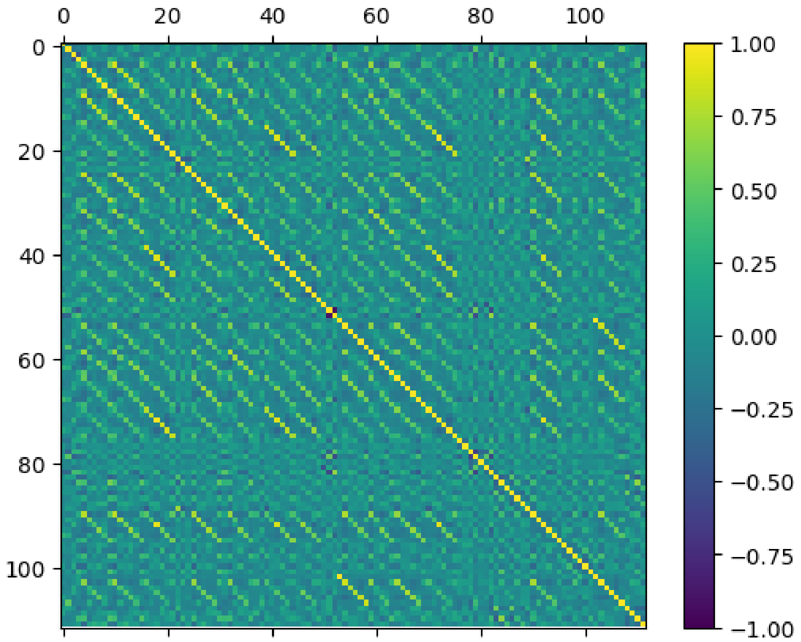
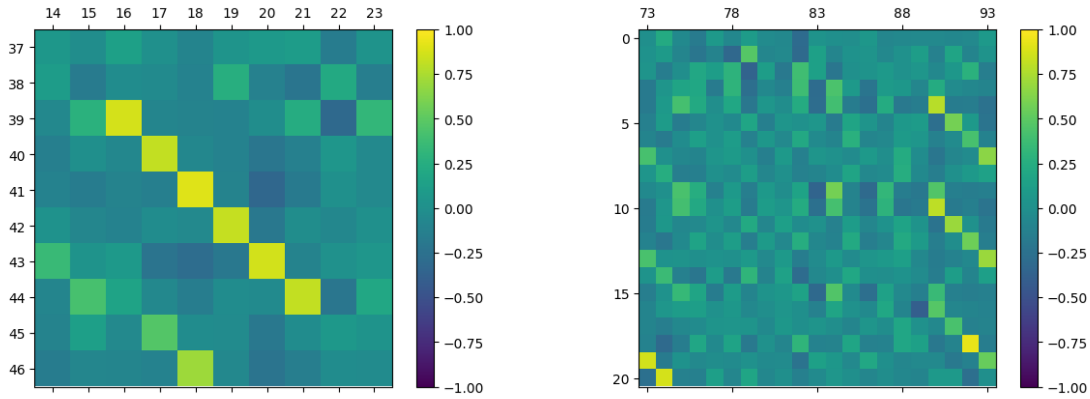


Figure 4.3: Pearson correlation coefficient of the CUB dataset.

regular pattern. In Figure 4.4a we have cropped the matrix to show one of these small diagonals. This diagonal shows a high correlation for concepts representing the bird's underparts colour on the x-axis, and breast colour on the y-axis where both bird parts are brown. As these small diagonals appear at regular intervals, and concepts in the dataset are organised by bird parts, these small diagonals show there is a high correlation between bird parts that have the same colour.

Additionally, there is one horizontal bar, and one vertical bar of low correlation shown in Figure 4.3 from concepts 76 to 89. A section of one of these is shown in Figure 4.4b. These concepts all represent "has_wing_shape" (concepts 76 and 77), "has_size" (concepts 78 to 80), "has_shape" (concepts 81 and 82), "has_back_pattern" (concepts 83 to 85), "has_tail_pattern" (concept 86 to 88) and "has_belly_pattern" (concept 89). The common attribute among these concepts is they represent the size, shape and pattern of a bird or bird part and therefore does not appear more often with one colour or another.

Next, the correlation between pairs of concepts for our Playing cards dataset is

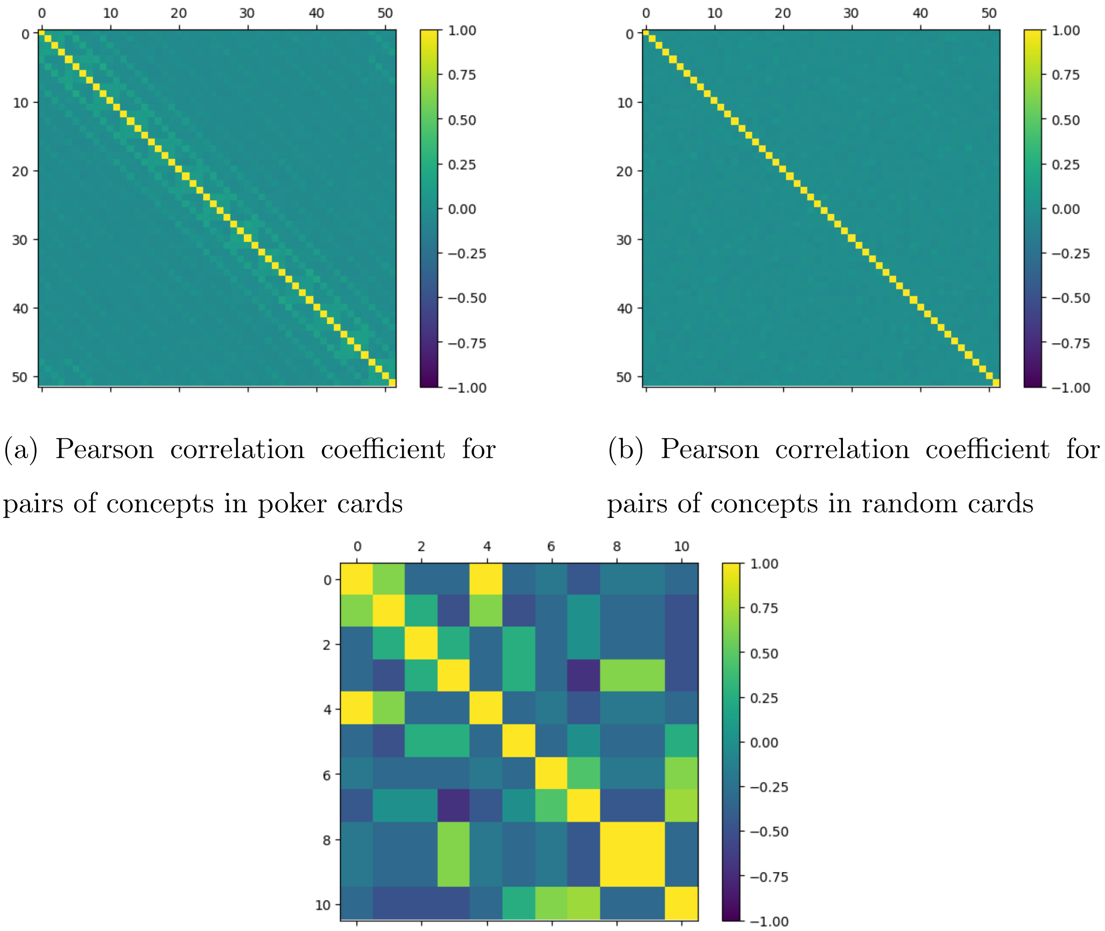


(a) Pearson correlation coefficient of the CUB dataset cropped to show concept pairs with a high correlation

(b) Pearson correlation coefficient of the CUB dataset cropped to show concept pairs with low correlation

Figure 4.4: Some concept pairs show high correlation while others do not.

shown in Figure 4.5. The average diagonal correlation is 1 for all dataset versions and an off-axis correlation of -0.02, -0.02 and -0.097 for Poker cards, Random cards and Class-level poker cards respectively. Starting with Poker cards in Figure 4.5a, apart from the centre diagonal elements we see little correlation between concepts. However, additional diagonal lines are going across the entire matrix showing a low but noticeable correlation. They occur at regular intervals with a spacing of 4 concepts between each diagonal line. The diagonal lines near the centre diagonal line have a higher correlation with diagonal lines showing less correlation as they get further away from the centre. This dataset was constructed to balance the task classes, resulting in some card combinations appearing more often than others. The combinations that were repeated the most were card hands for the task label "straight flush" where 48 card combinations were repeated around 37 times each. This task label means there are three suited cards in sequence. Concepts in the dataset are ordered first by card rank and then by card suit. This means every 4th concept represents a new card rank (e.g. the first 4 concepts are for the card rank "2" and then the next 4 concepts are for the card rank "3"). This is not the only



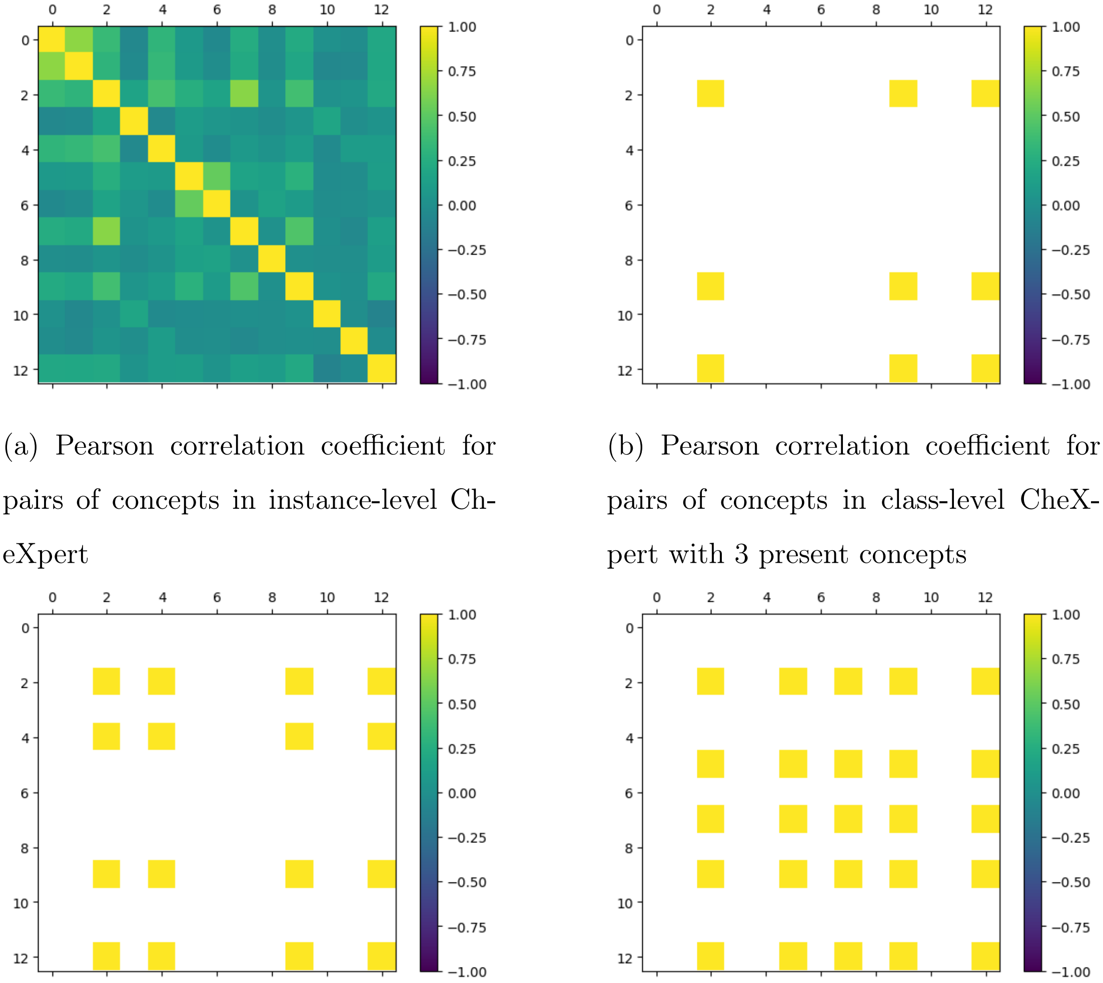
(a) Pearson correlation coefficient for pairs of concepts in poker cards

(b) Pearson correlation coefficient for pairs of concepts in random cards

(c) Pearson correlation coefficient for pairs of concepts in class-level poker cards

Figure 4.5: Pearson correlation coefficient for pairs of concepts in the Playing cards dataset.

task label that requires all concepts to have the same suit or rank, but these have fewer card combination repeats. As the number of card combinations increases, the correlation between concepts decreases. This explains why the correlation is highest close to the centre diagonal. Essentially the task label “straight flush” will have to include concepts close together whereas “straight” does not have this constraint. Figure 4.5a shows the concept correlation has picked up the repeated hand ranks that are a by-product of balancing task labels.



(a) Pearson correlation coefficient for pairs of concepts in instance-level CheXpert

(b) Pearson correlation coefficient for pairs of concepts in class-level CheXpert with 3 present concepts

(c) Pearson correlation coefficient for pairs of concepts in class-level CheXpert with 4 present concepts

(d) Pearson correlation coefficient for pairs of concepts in class-level CheXpert with 5 present concepts

Figure 4.6: Pearson correlation coefficient for pairs of concepts for the CheXpert dataset. White represents concept pairs where one or more concepts were not present for all samples.

As Random cards do not balance task labels there is no pattern observed in the concept correlation matrix in Figure 4.5b. Apart from the centre diagonal elements with high correlation, all off-diagonal elements have almost no correlation.

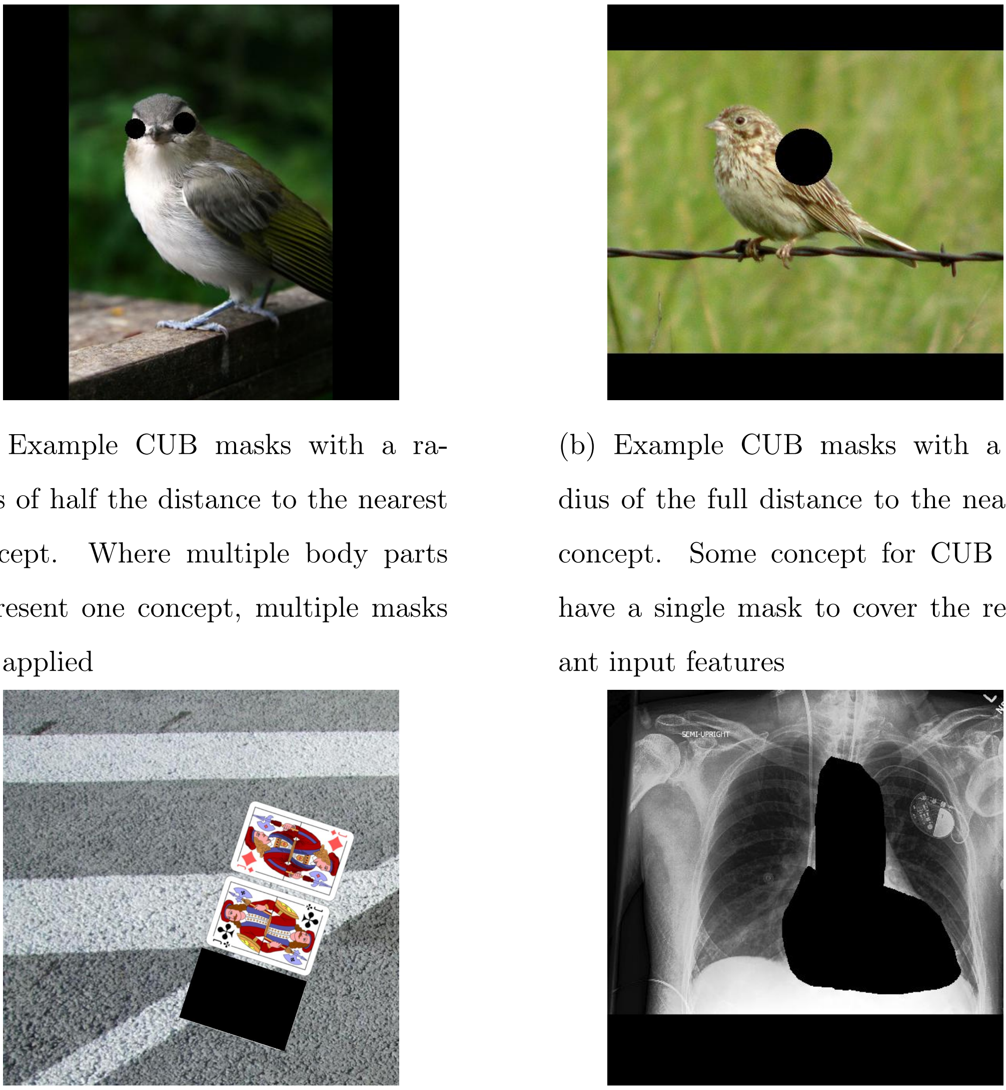
Class-level poker cards show several concept pairs with high correlation. All of the concept pairs with high correlation align with the concept occurrences in the dataset as seen in Table 3.2. Any concept pairs that only occur together e.g. “6 of Diamonds” and “9 of Diamonds” (concepts 8 and 9) have a concept correlation of 1, while concept pairs that do not exclusively occur together have a lower correlation in line with how many other concepts they occur with. Concept pairs that never occur together have a negative correlation.

Finally, for CheXpert we have displayed the concept correlation for the 4 versions of the dataset in Figure 4.6. Starting with Instance-level CheXpert in Figure 4.6a, most concept pairs have a little correlation. As all concepts relate to observations in the chest this is to be expected some concepts are more likely to occur together e.g. “enlarged_cardiomediastium” (concept 0) and “cardiomegaly” (concept 1) are both enlargements of the heart. Two concepts, “Pleural_other” (concept 10) and “fracture” (concept 11) have distinctly lower correlations than other concepts. “Pleural_other” occurs infrequently in the dataset. In the test dataset split that we used to create these matrices it occurs 8 times. “fracture” also occurs infrequently (6 times in the test dataset split) and is the only concept that is not related to an organ and thus we would not expect it to appear more often with an observation found with an organ. Moving to our Class-level CheXpert datasets we observe a concept correlation of 1 for all concepts that are present. We could not compute the correlation for any other concepts as they are not present in every sample. These concept pairs are shown as white in the matrices.

4.6 Experiment Set-up

In this chapter, we used the same datasets and models as described in Chapter 3. Namely, we use the datasets CUB, Playing cards, and CheXpert, including the same dataset variations. The VGG model architecture for the CUB and Playing card models concept encoders, and Densenet121 model architecture for CheXpert models.

As detailed in Section 4.5, our OIS results are averaged over multiple runs of



(a) Example CUB masks with a radius of half the distance to the nearest concept. Where multiple body parts represent one concept, multiple masks are applied

(b) Example CUB masks with a radius of the full distance to the nearest concept. Some concept for CUB will have a single mask to cover the relevant input features

(c) Example Playing cards mask. Each mask covers an entire playing card

(d) Example CheXpert mask. Each mask fills a human segmentations

Figure 4.7: CRA single concept masking covers the input features for one concept.

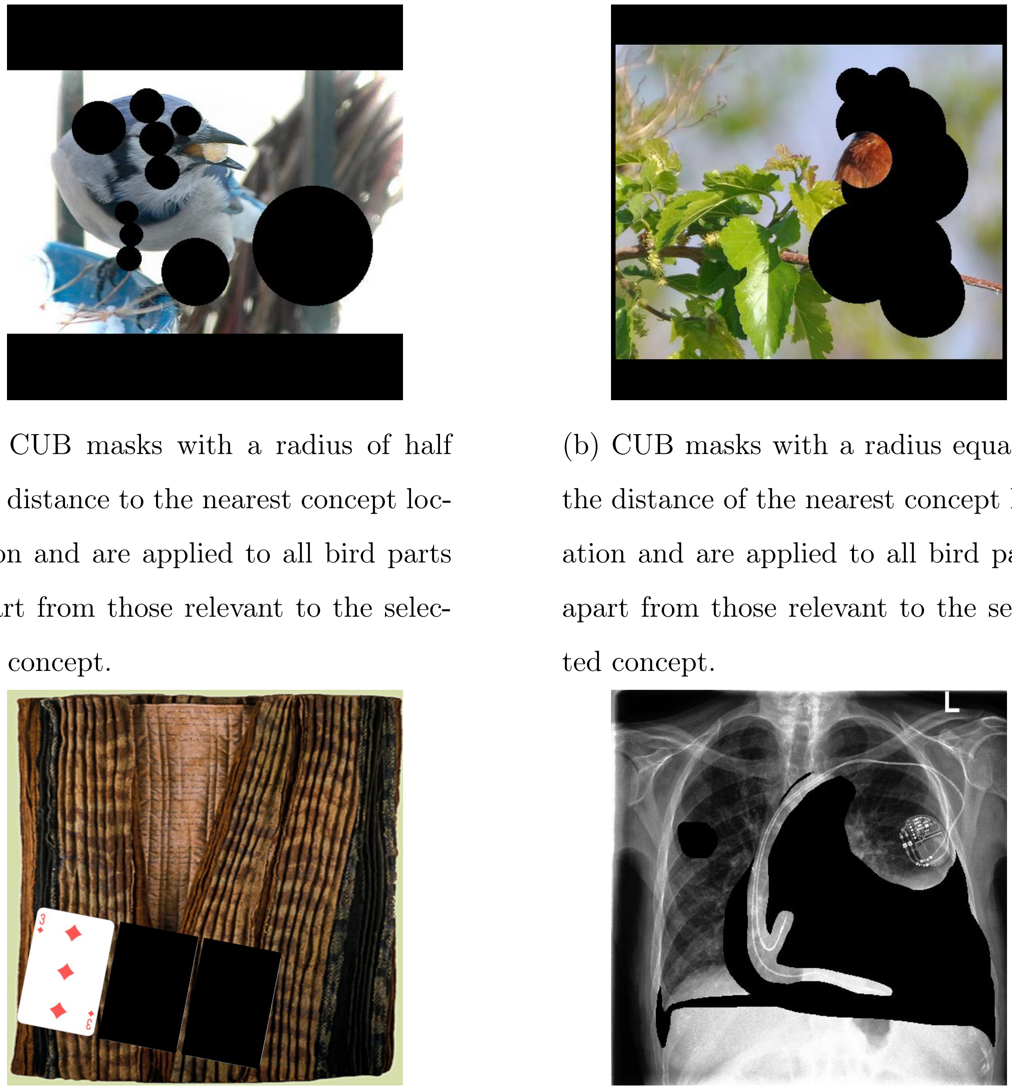
results. Specifically, we generate results using the training repeats we created for each dataset, with each model weight used to generate an OIS result 3 times for a total of 15 OIS runs for Playing cards and CheXpert, and 9 for CUB.

CRA masks concepts by changing each input feature value that's semantically meaningful to a concept. Formally, each concept j can be predicted by a set of semantically meaningful input features $a_j \subseteq x$, where $x \subseteq \mathbb{R}^d$ represents the set of all input features (e.g., pixels in the input image). The set a is the smallest subset of x that is sufficient to predict concept j. We apply zero masks, meaning that for each input feature $\eta \in a_j$, we set $\eta = 0$.

For the Playing cards and CheXpert datasets, we have ground truth pixel masks for each concept, whereas, for CUB, only the x and y coordinates of bird parts are available. To address this, we applied circular masks where the radius either extends to the nearest bird part or is set to half the distance between parts. Our method differs from (Raman et al., 2024), which based mask sizes on the image dimensions. Instead, by calculating the radius from the relative position of bird parts, we better accommodate variations in bird size, shape and image perspective. Concepts in CUB may require multiple bird parts to be masked (e.g. eye colour will apply a mask to both eyes if visible).

Single-concept masks cover input features for a single concept, while multi-concept masks cover input features for all concepts before revealing those relevant to a single concept. Multi-concept masks are applied in this way as some concept semantically meaningful input features overlap. An example of single-concept masks for all datasets is shown in Figure 4.7, and multi-concept masks in Figure 4.8.

We used single-concept masks to evaluate the model's sensitivity to removing specific concept input features. In contrast, multi-concept masks were used to assess the model's sensitivity when only the input features associated with a single concept were retained. Together, these two masking strategies offer complementary insights: single-concept masks reveal the importance of individual concept input features, while multi-concept masks assess the model's dependence on input features for a single concept and its sensitivity to the absence of input features associated with other concepts.



(a) CUB masks with a radius of half the distance to the nearest concept location and are applied to all bird parts apart from those relevant to the selected concept.

(b) CUB masks with a radius equal to the distance of the nearest concept location and are applied to all bird parts apart from those relevant to the selected concept.

(c) Two out of the three playing cards are covered when multi-concept masks are applied.

(d) All human segmentations are masked before removing the mask for a selected concept segmentation.

Figure 4.8: Example multi-concept masks for each dataset. All concept input features are masked apart from the features for a single concept.

Dataset	OIS
CUB	0.125 (±0.004)
Random cards	0.206 (±0.031)
Poker cards	0.193 (±0.019)
Class-level poker cards	0.261 (±0.018)
Instance level CheXpert	0.506 (±0.016)

Table 4.1: OIS is a metric for measuring additional or lacking information in learned concepts compared to the ground truth concepts. Lower OIS values indicate that the information in the model's learned concepts are more aligned with the ground truth concepts. Models trained on CUB and Poker cards encode the least difference in information from learned concepts compared to ground truth concepts which is closely followed by Random cards. Instance-level CheXpert encode the most difference from learned concepts compared to ground truth concepts. We show the standard deviation for OIS in brackets.

4.7 Results

4.7.1 Concept Purity

To assess inter-concept impurity across different datasets and models, we measured the OIS for CUB, Poker cards, Random cards, Class-level Poker cards, and Instance-level CheXpert. Table 4.1 summarises the results. Among these, CUB models had the lowest impurity, while the Instance-level CheXpert models showed the highest OIS with a value close to 0.5. This suggests that the inter-concept predictability of the learned concepts in CheXpert differs significantly from the dataset's concept annotations.

For Playing card models, those trained on Poker cards slightly outperformed models trained on Random cards, indicating models trained on Poker cards can better align concept outputs to the ground truth concept annotation in the dataset. Within the three variations of Playing card models, the Class-level Poker card models had the highest impurity.

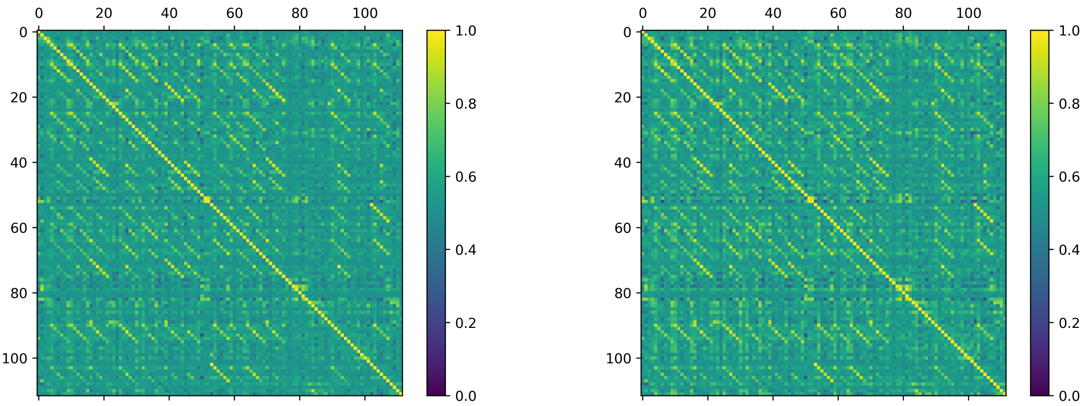
Since OIS is based on the expected inter-concept predictability from the concept annotations in each dataset, if several ground-truth concepts are highly correlated and the model's concept outputs reflect the same correlation, the OIS will show low impurity. However, if the concept representations capture different inter-concept predictability between concept pairs, the OIS will indicate higher impurity. As we have shown, Random cards have a higher OIS than Poker cards. We can infer that this difference is because the Random cards' concept encoder is trained on the Random cards version of the dataset, but tested on Poker cards. This means the concept representation Random card models have learnt may have avoided capturing the imbalance of co-occurring concepts as seen previously in Figure 4.5a.

Our findings on synthetic datasets align with prior research (Espinosa Zarlenga et al., 2023), where the authors achieved an OIS of approximately 0.2 on their datasets.

To understand OIS values more thoroughly, we need to break down the metric into its components, for which we present the oracle and purity matrices that make up the OIS for each model. These matrices organise concepts by their index along the x and y axes, with each element representing the AUC value of the concept on the y-axis predicting the concept on the x-axis. Centre diagonal elements correspond to a concept predicting itself.

4.7.1.1 CUB

Our CUB models, as shown in the oracle and purity matrices in Figure 4.9, show high AUC values for diagonal elements from concept 0 to concept 112, indicating that each concept contains sufficient information to accurately predict themselves. Additionally, we observe smaller diagonals at regular intervals throughout the



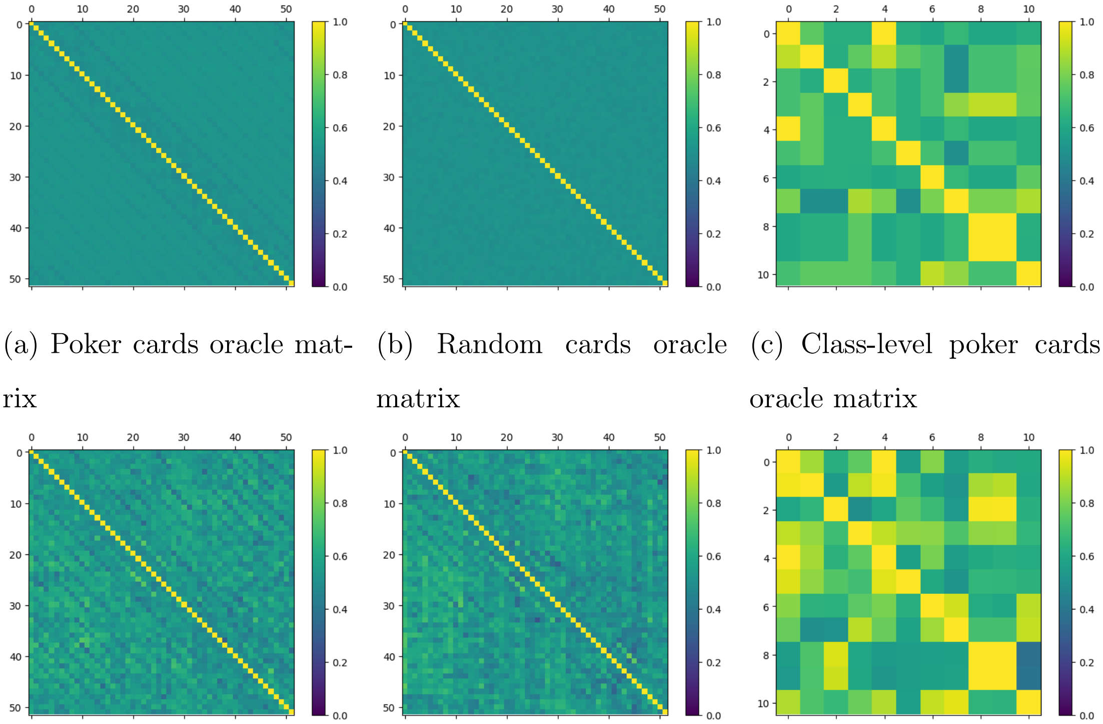
(a) Instance-level CUB oracle matrix

(b) Instance-level CUB purity matrix

Figure 4.9: The oracle and purity matrices for CUB are similar to one another and also has a strong resemblance to the correlation of concepts in the dataset. Each concept has encoded the required information to predict itself and other similar concepts such as those for other bird parts of the same colour.

matrices, showing periodic, but regular, clusters of inter-concept predictability. In addition, between concepts 76 and 89 there is a noticeable reduction in inter-concept predictability. When compared to the inter-concept correlations in the dataset (Figure 4.3), the regions of high inter-concept predictability closely align, which suggests the CBM concept encoder has learned the correlations of concepts in the training data.

Comparing the oracle and purity matrices, we observe minimal differences between them. This alignment directly contributes to the low OIS and shows the model's concept predictions are nearly as accurate in predicting ground truth concepts as the ground truth concepts themselves. In other words, the model effectively mirrors the predictive relationships found in the ground truth data.



(a) Poker cards oracle matrix

(b) Random cards oracle matrix

(c) Class-level poker cards oracle matrix

(d) Poker cards purity matrix

(e) Random cards purity matrix

(f) Class-level poker cards purity matrix

Figure 4.10: Oracle and purity matrices for Random cards, Poker cards and Class-level poker cards. The oracle matrices show inter-concept predictability of ground truth concepts while the purity matrices show inter-concept predictability of learned concepts w.r.t. ground truth concepts. Poker cards show numerous diagonals of non-random inter-concept predictability. Class-level poker cards show a high level of inter-concept predictability for a few concepts within learned concept representations that did not exist in the oracle matrix.

4.7.1.2 Playing Cards

The matrices for our Playing card models are shown in Figure 4.10. All oracle matrices show a strong predictability for each concept predicting themself by the centre diagonal elements all being close to 1. In the oracle matrix for Poker cards (Figure 4.10a), we observe additional diagonals of elements with AUC values below 0.5. This shows the presence of non-random relationships between different concepts. In contrast, the oracle matrix for Random cards only shows the centre diagonal has high inter-concept predictability, with all off-diagonal elements showing random inter-concept predictability.

The off-diagonal elements in the purity matrix for Poker card models (Figure 4.10d) show the relationships between certain concepts are non-random. Each of these concept pairs also appears in Figure 4.10a. Hence, it is clear Poker card models have learned the same inter-concept predictability that exists in the training data. In contrast, these off-diagonal elements of non-random correlation are significantly reduced in the Random cards models purity matrix (Figure 4.10e), reflecting the absence of inter-concept predictability that exists in the Random cards dataset. The purity matrices for Poker cards and Random cards align with the higher OIS value for Random cards in Table 4.1. The higher OIS indicates a lack of expected information for inter-concept predictability for models trained on Random cards compared to the structured patterns seen in Poker cards ground truth concepts.

As for Class-level poker cards, the oracle matrix (Figure 4.10c) displays most ground truth concept pairs having a low inter-concept predictability. In particular, these represent concept pairs where the concepts never co-occur in any sample. These concept pairs in the dataset have a negative correlation, and as such, the low inter-concept predictability of concepts aligns with the negative correlation of ground truth concepts. Concept pairs in Figure 4.10c that have high inter-concept predictability, e.g. "Four of Hearts" (index 4) and "Two of Hearts" (concept 0), align to concepts that exclusivity co-occur. Concepts that co-occur with multiple sets of concepts have a slightly lower inter-concept predictability.

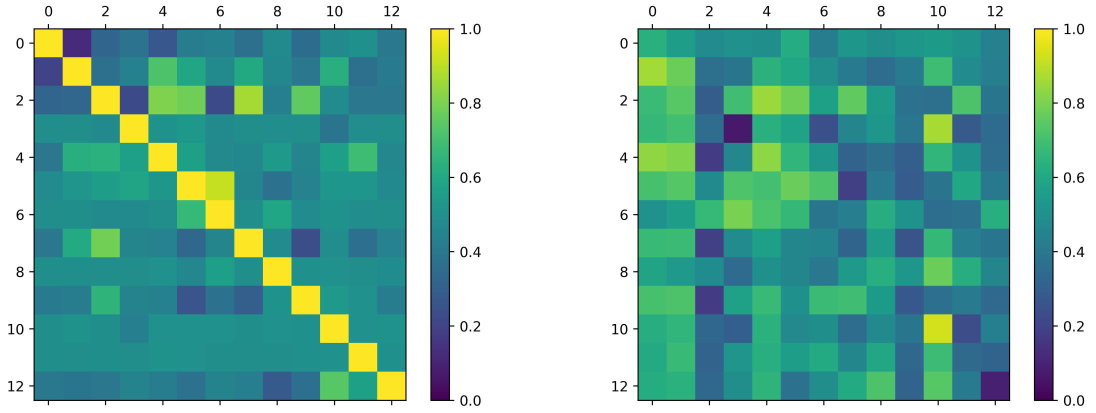
The purity matrix (Figure 4.10f) shows higher AUC values overall, indicating that the learned concepts encode more information than the ground truth concept labels. As the ground truth concepts are either 0 or 1, but the predicted concepts can have a value between 0 and 1, the predicted concepts allow for more information to be encoded. For example, the concept "Four of Clubs" (index 2) can predict the presence of the “Six of Diamonds” (index 8) and “Nine of Diamonds” (index 9), despite no explicit task-related connection.

4.7.1.3 CheXpert

Finally, the oracle and purity matrices for Instance-level CheXpert are shown in Figure 4.11. As expected, the oracle matrix shows a high OIS values for the centre diagonal elements, indicating that each ground truth concept has sufficient information to predict itself. Additionally, the inter-concept predictability of the non-centre diagonal element aligns reasonably well with the correlations observed in the dataset in Figure 4.6.

In contrast, the purity matrix shows a notably large difference from the oracle matrix, starting with the absence of a centre diagonal with high (or low) AUC values. This shows the information the model has learned for individual concepts is not enough to accurately predict themselves. Instead, there is a mixture of AUC values across the matrix. A few concepts, such as “edema” (concept 4), “atelectasis” (concept 7), and “pleural_effusion” (concept 9) on the y-axis, and “lung_opacity” (concept 2) on the x-axis show some alignment with the dataset’s inter-concept predictability, but overall, the purity matrix suggests that the model has not learned the inter-concept predictability that exists in the ground truth concept values.

The models used for the purity matrix do not exhibit particularly high accuracy, with the sequential model achieving an average concept accuracy of 75.765%. This explains the greater variability observed in the purity matrix compared to the oracle matrix. Certain concepts, such as “support_devices” (concept 12), may be easier to predict due to their distinct visual differences, while other concepts may appear together more often, or share similar parts on the input image.



(a) Instance-level CheXpert oracle matrix

(b) Instance-level CheXpert purity matrix

Figure 4.11: The oracle matrix shows high OIS values along the center diagonal which are absent from the purity matrix. The purity matrix shows mixed AUC values suggesting the model struggles to learn individual or inter-concept predictability. The difference in the purity and oracle matrices reflects the model's average concept accuracy of 75.77%.

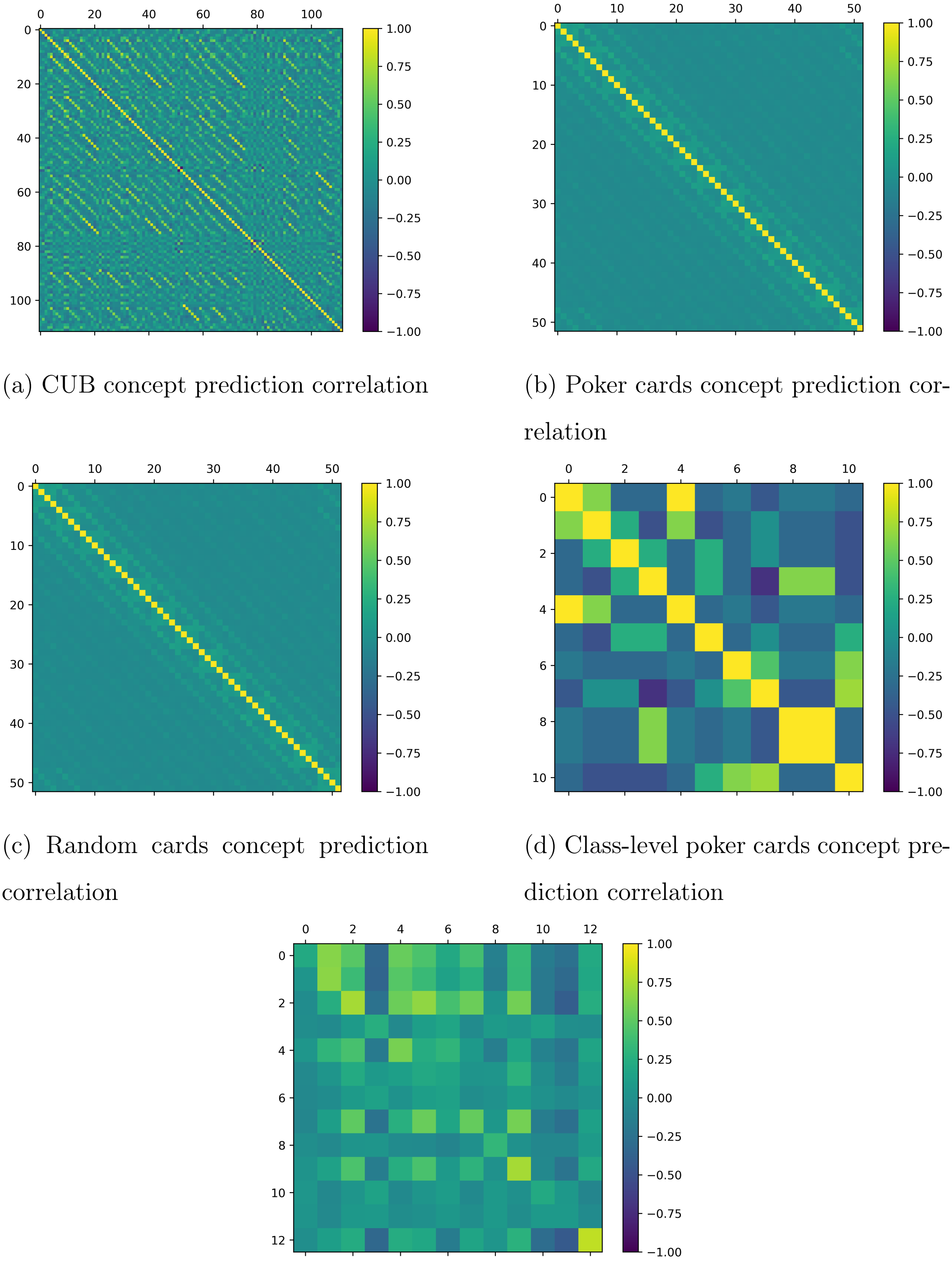
4.7.1.4 CBMs Learns The Inter-concept Predictability From Their Training Data

RQ2, and sub-question 1 in particular, asks how the dataset affects information leakage. The OIS results demonstrates all of our models suffer from at least minor information leakage. In our case this is evidence from the inter-concept predictability of concept outputs. Our models trained on CUB, Random cards, and Poker cards, resulted in a low OIS indicate that the models effectively learn relationships between related concepts observed in the dataset. However, a low OIS also indicates a model has learned the inter-concept predictability of the data they were trained on. Examples of this occurring with our models include high inter-concept predictability in CUB, such as bird parts sharing similar colours, or Poker cards models accurately capturing the concepts that have a greater frequency of co-occurring. Both instance and class-level concepts can learn the underlying inter-concept predictability in a dataset.

CBMs predict concepts with continuous concept outputs (values between 0 and 1). This allows concept outputs to encode richer inter-concept relationships compared to binary outputs from the dataset. This is particularly evident for the models trained on the Class-level Poker cards dataset, where continuous outputs enable indirect inter-concept predictability between concepts. For example, the model links the concept “Six of Diamonds” (concept 8) indirectly to the concept “Four of Clubs” (concept 2) by utilising their shared co-occurrence with the concept “Four of Diamonds”. Specifically, “Four of Clubs” may encode the presence of “Six of Diamonds” through varying concept values: 0.6–0.7 when “Four of Diamonds” is present (indicating “Six of Diamonds” is absent) and 0.8–1.0 when “Four of Diamonds” is absent (indicating “Six of Diamonds” is present).

This behaviour is not observed for models trained on CUB, likely due to the higher number of concepts and the resulting complexity encoding additional concepts in each concept value. In datasets with many concepts or significant crossover in concept co-occurrence, the capacity to encode inter-concept information into learned concept representations is reduced, as the additional concepts create ambiguity over what concept values mean unless there is a direct co-occurrence link.

The key takeaway from the OIS results is that CBMs learn similar inter-concept predictability to the training data. This means that if concepts in a dataset facilitated inter-concept predictability then a CBM will learn this and the trained model will exhibit an equivalent degree of information leakage. While this may not cause harmful effects when the training data accurately reflects real-world scenarios, it may lead to bias in trained models if real-world scenarios have a different frequency of concepts to the training data. In such situations, CBMs may struggle to represent concepts appropriately, making them unsuitable for their intended applications.



(a) CUB concept prediction correlation

(b) Poker cards concept prediction correlation

(c) Random cards concept prediction correlation

(d) Class-level poker cards concept prediction correlation

(e) Instance-level CheXpert concept prediction correlation

Figure 4.12: The correlation of predicted concepts closely mirroring the dataset's concept correlations.

The results from the Random cards dataset demonstrate that it is possible to mitigate some of these drawbacks. By separating the concept dataset and task dataset, the resulting model did not learn the same representations of concepts that was preset in the task dataset (Poker cards). This avoided some of the bias Poker cards models learned which allowed them to predict the co-occurrence of concepts that appeared more often in the dataset.

Expanding on the purity matrices, in Figure 4.12 we evaluate the Pearson correlation coefficients between predicted concepts and ground truth concept annotations across the datasets. Most models achieve strong alignment, with predicted concept correlations closely mirroring the dataset's concept correlations. This includes models trained on CheXpert where the predicted concept correlations for the most part reflect the dataset's underlying structure, except the centre diagonal. This was to be expected considering the model's lower concept accuracy.

To quantitatively evaluate alignment, we use the Structural Similarity Index (SSIM) metric (Wang et al., 2004), which compares both structural similarity and the values of concept pairs. We used SSIM instead of Mean Squared Error (MSE) as MSE only captures element-wise differences and does not account for structural patterns in the data. SSIM, by contrast, considers local spatial correlations, making it more appropriate for structured matrices. For example, even if the individual values in the oracle and purity matrices did not match, SSIM can detect the similarity of structural patterns, such as the diagonals we have observed with Porker cards matrices. We show the results in Figure 4.13 where we performed two comparisons: (1) oracle and purity matrices alignment, and (2) the alignment between predicted concepts correlations and dataset concept correlations.

For oracle and purity matrices, models trained on datasets with class-level concept annotations have a higher SSIM scores than instance-level models. The higher SSIM scores for class-level concepts reflect how these models were able to capture both the concept pairs and AUC values in the learned concept representations.

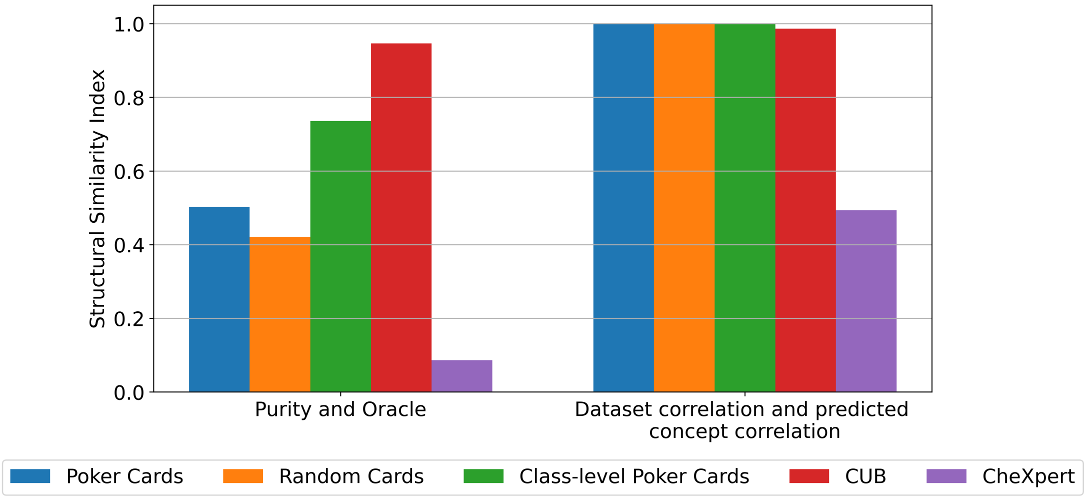


Figure 4.13: Purity and oracle matrices show a closer alignment for models trained on datasets with class-level concept annotations than models trained on datasets with instance-level concept annotations where the models capture much of the structure but show increased noise in purity matrices. Dataset concept and predicted concept correlation SSIM shows high alignment for all models, apart from those trained on CheXpert.

The instance-level models trained on Playing cards were able to align much of the structure between the two matrices, but the purity matrices introduced a greater amount of noise.

For predicted concept correlation vs. dataset concept correlations, most models achieve high SSIM, confirming strong alignment with dataset concept structures. However, CheXpert achieves an SSIM of 0.5, indicating moderate alignment. While this value suggests CheXpert captures some similarities between predicted concept pairs and ground truth concept pairs, it also highlights there are key differences. As previously discussed, this is due to the model concept accuracy. Importantly, an SSIM of 0.5 means predicted concept correlations are not reversed (SSIM equal to -1) or void of all similarities (SSIM equal to 0).

4.7.2 Concept Removal Accuracy

Sub-question 2 from RQ2 asks how training data affects input feature dependence. To answer this we evaluated how changes in input features affect concept accuracy using the CRA metric (Heidemann et al., 2023) and locality masking techniques (Raman et al., 2024). By using this metric we assess whether concepts that are predicted as present change to not present when relevant input features are removed, and whether concept predictions remain predicted as present when irrelevant input features are masked.

As previously mentioned, we used two types of masking based on the masking options defined in (Raman et al., 2024): single-concept masking and multi-concept masking. Single concept masking is similar to Raman et al. (2024)'s concept-relevant masking, where we apply a mask to cover a single concept with semantically meaningful input features. In this case, we expect only the accuracy of the masked concept to decline, which demonstrates a model has learned to predict concepts using semantically meaningful input features. Multi-concept masking is based on Raman et al. (2024)'s concept irrelevant masking, where we mask all but one concept's input features, and the expected output would see the unmasked concept accuracy to maintain high, while the masked concepts accuracies to be low.

Our use of CRA is mathematically defined in Equation 4.1. The CRA value, denoted as $R_{i|j}$, measures the change in accuracy of concept i when the input features corresponding to the concept(s) j are masked. Specifically, we measure the fraction of true positive concept predictions for i (TP_i) where the concept predictions $\hat{c}_i \in \mathrm{TP}_i$ changes from present to not present ($\hat{c}_i$ drops below 0.5) when input features for concept j are masked. This is calculated over the total number of true positive predictions for concept i.

$$R_{i|j} = \frac{|\{\hat{c}_i \in \mathrm{TP}_i \mid \hat{c}_{i|j} >= 0.5 \wedge \hat{c}_{i|\neg j} < 0.5\}|}{|\mathrm{TP}_i|} \tag{4.1}$$

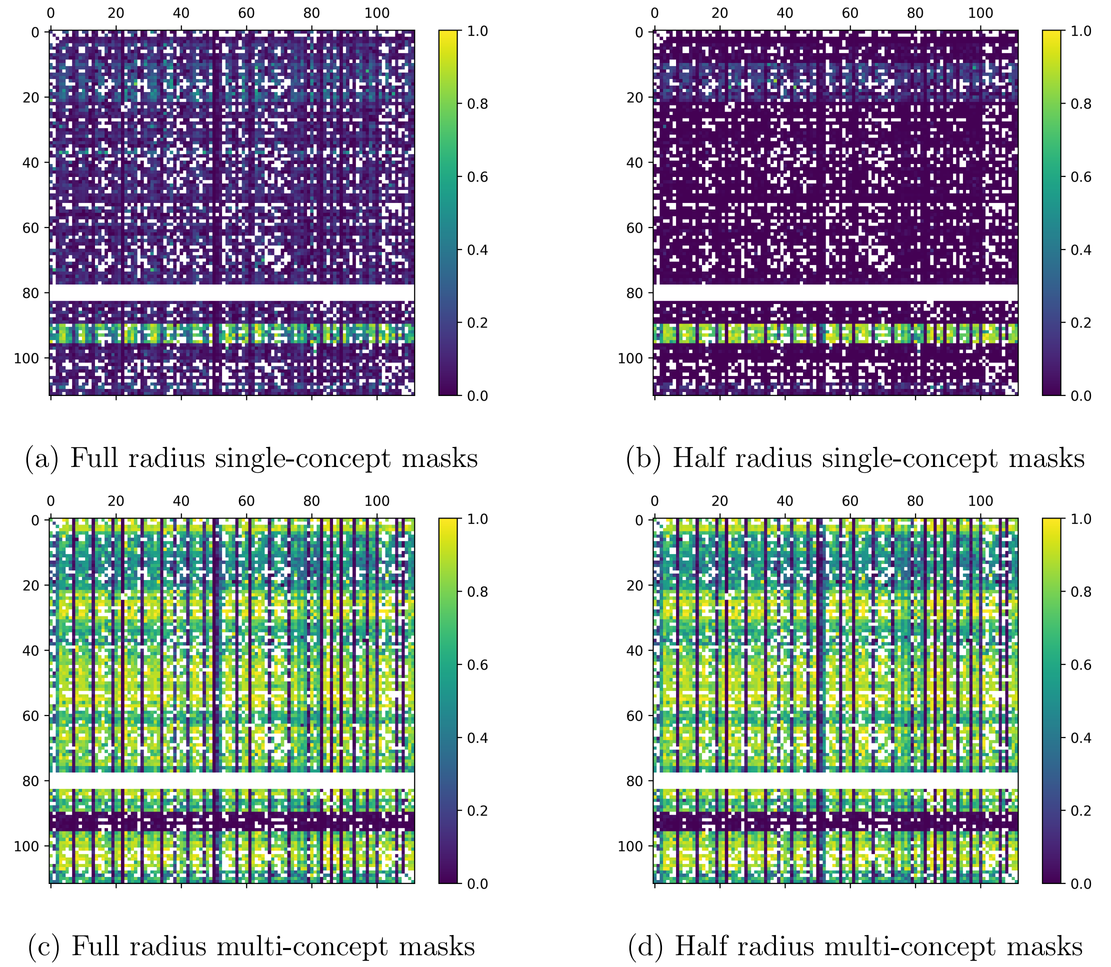
Single-concept masking will see i set to the concept being measured and j the single concept masked. When i and j are the same the masks are equivalent to concept-relevant masks. Multi-concept masking will see i set to the concept being measured and j set to all concepts that are masked. We always exclude one concept from j. If i and j are the same then multi-concept masking is equivalent to concept irrelevant masking.

Our results are presented in a matrix where the y-axis represent the concepts whose input features are either masked or retained, and the x-axis correspond to target concepts being evaluated. Our results are averaged over the repeated training runs for each dataset. The matrix elements show the average CRA for each concept pair, with white squares representing concept pairs that are not present in the dataset. A perfect CRA result for single concept masking would show a CRA of 1 for all concepts along the centre diagonal elements, while all other concepts would have a CRA of 0. We'd expect to see the opposite for multi-concept masking.

4.7.2.1 CUB

Starting with models trained of the CUB dataset, we produced the CRA by placing circular masks over concept locations using two different radius sizes: *full radius* masks where the radius of each circle extends to the next nearest concept location, and *half radius* masks where the radius is set to half the distance to the next nearest concept location. We used two different mask sizes as the ground truth input features for each concept are not explicitly provided and thus by using two mask sizes we can analyse the difference in change of concept accuracy between the two.

As shown in Figure 4.14, both half and full radius masks for single-concept masking have minimal impact on the accuracy of predicting the corresponding concept. Our results show CUB does not match a perfect CRA. All concepts change from



(a) Full radius single-concept masks

(b) Half radius single-concept masks

(c) Full radius multi-concept masks

(d) Half radius multi-concept masks

Figure 4.14: Concepts for models trained on CUB are not sensitive to semantically meaningful input features being masked in the input, while sensitive to all but the semantically meaningful input features being masked.

a present prediction to a not-present prediction at a similar rate, irrespective of which concept is masked. In total only 13.1% of concept predictions change from present to not present with full radius masks (Figure 4.14a), and 5.8% with half radius masks (Figure 4.14b). This makes sense as full-radius masks will cover a larger amount of the bird in each image than half-radius masks and thus there will be more information left in the input for half-radius masks that the model can use for concept predictions.

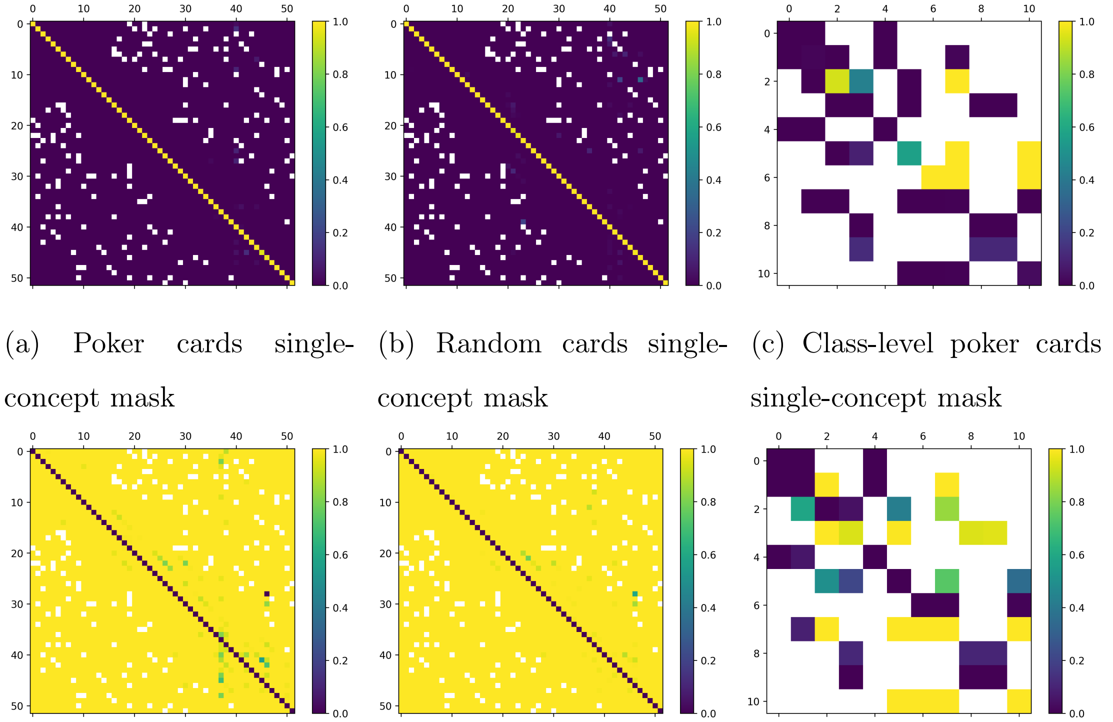
When we remove concepts 90 through 95, which correspond to the primary bird colour, we observe a significant drop in CRA across most concepts, with nearly all concept predictions changing from present to not present. This is due to these concepts placing masks over most of the bird's visual representation. Thus, masking most input features for the model is required to accurately predict any concept.

Concepts 78 to 82 are blank as these concepts are for bird sizes and shapes and would require a mask placed over all concept input features. Since these concepts do not correspond to any specific bird part, they were excluded from CRA.

Multi-concept concept masks, for the most part, show the reverse of the single-concept masks. Figure 4.14c shows the CRA results with full-radius masks, and Figure 4.14d shows the results with half-radius masks. In most cases, the model's prediction changes from identifying most concepts as present to absent. Concepts 90 through 95 show almost no reduction in concept accuracy, as the majority of each bird remains visible in each image.

In addition, with multi-concept masks, we see concepts that represent bird parts that are black, such as concept 19 ("underparts color black") and concept 34 ("upper tail color black"), do not change their concept prediction no matter which concept input features are masked. As the masks are black in colour, they blend into the parts of the input image they are supposed to mask, allowing the model to use the masked regions to maintain accuracy.

With circular masks, we expect some overlap between masked input features and those irrelevant to the target concept. Despite this, we do not observe any discernible noise in the results. With single-concept masks, we might expect additional concepts to lose present predictions if the mask covers input features relevant to other concepts. On the other hand, with multi-concept masks, we might expect some concepts to retain present predictions if their input features are only partially masked. In both cases, we do not observe these effects with



(a) Poker cards single-concept mask

(b) Random cards single-concept mask

(c) Class-level poker cards single-concept mask

(d) Poker cards multi-concept mask

(e) Random cards multi-concept mask

(f) Class-level poker cards multi-concept mask

Figure 4.15: Concepts for model trained on Poker cards and Random cards are sensitive to the removal of semantically meaningful input features. Concepts for models trained on Class-level poker cards are mostly not sensitive to semantically meaningful input features.

either half-radius or full-radius masks.

4.7.2.2 Playing Cards

For Playing cards CRA (Figure 4.15), we have first separated the results between the models trained on datasets with instance-level concept annotations and class-level concept annotations. In both Poker cards and Random cards, we observe that when a single concept is masked, only the corresponding concept shows a reduction in accuracy. Other concept pairs show little to no change in their predictions, which suggests that removing semantically meaningful input features exclusively impacts the corresponding concept predictions.

Similarly, for the multi-concept mask, we observe that only the remaining concept is still predicted as present. However, there are some exceptions. In both Poker cards and Random cards, certain concepts do not follow this pattern, as shown by the green diagonal in Figure 4.15d and Figure 4.15e. These diagonals are spaced four concepts away from the centre, and thus we can conclude they relate to the downstream class “Three of a kind” appearing more often than other classes in the dataset. In any case, the presence of the model continuing to predict a masked concept is lower than how often it does not predict the presence of a masked concept. We also observe a vertical line with Poker cards for multi-concept masks. Similarly, this is not a general case for all concept predictions and only affects two of our five models trained on this dataset version, so is likely caused by a run-to-run variation.

Class-level poker cards largely confirm prior observations: the input features used to predict each concept are not always semantically meaningful. Figure 4.15c shows that for most concepts, predictions remain unchanged when semantically meaningful input features are masked. The exceptions to this are the concepts “Four of Clubs” (concept index 2), “Four of Spades” (concept index 5), and “Five of Clubs” (concept index 6). These same concepts also show high proportions of positive feature attribution values in Figure 3.13, reinforcing the idea that the model is capable of learning semantically meaningful input features to predict concepts with class-level concepts, but this capability is very limited. Even for these concepts “Four of Clubs” and “Four of Spades” do not consistently change from a present prediction to a not-present prediction when semantically meaningful concepts are masked.

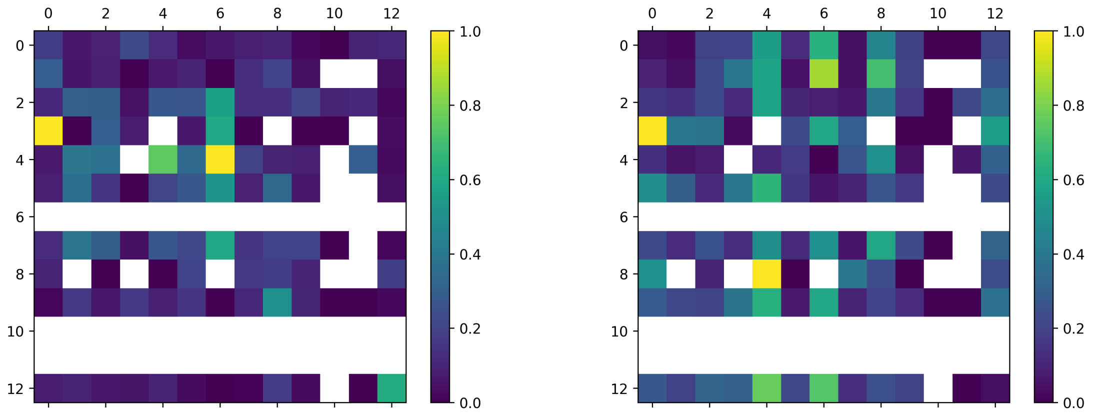
Other concepts show varying behaviour with single-concept masks. Some concept predictions change from present to not present when unrelated input features are masked, demonstrating the model’s overly sensitive to unrelated input features. Alternatively, other concept predictions remain the same regardless of which concept is masked. In this case, the model has learned multiple sets of input features can accurately predict the same concept. For instance, the “Ten of Hearts” (concept index 10) demonstrates prediction changes when unrelated input features are masked, while “Two of Hearts” (concept index 0) never changes its prediction.

Turning our attention to CRA with multi-concept masks, as shown in Figure 4.15f, we observe some concepts CRA are an inverse of the CRA they achieved with single concept masks. For the concepts “Four of Clubs”, “Four of Spades”, and “Five of Clubs” this again confirms the model has limited ability to learn the intended input features for concepts. However, for other concepts, we see no change in CRA (such as the concept “Two of Hearts”), once again demonstrating the models can use multiple sources of input features to predict such concepts. Overall it remains clear models trained on Class-level poker cards do not generally predict the presence or absence of concepts based on semantically meaningful input features, and are sensitive to irrelevant input features.

4.7.2.3 CheXpert

Finally, for models trained on the CheXpert dataset, we present the CRA for both our instance-level and class-level trained models (see Figure 4.16 for instance-level results and Figure 4.17 for class-level results). Starting with the instance-level results, there are several cases where no CRA value is available. These missing values occur when no ground-truth positive predictions exist, or when concept segmentations are not available for the concept being removed/kept. For example, “pleural_other” (concept 10) and “fracture” (concept 11) appear only 8 and 6 times respectively in the test dataset, while “pneumonia” (concept 6) appears 14 times. There is little opportunity for these concepts to appear alongside other concepts.

Single concept masks (Figure 4.16a) show most concepts do not switch from a



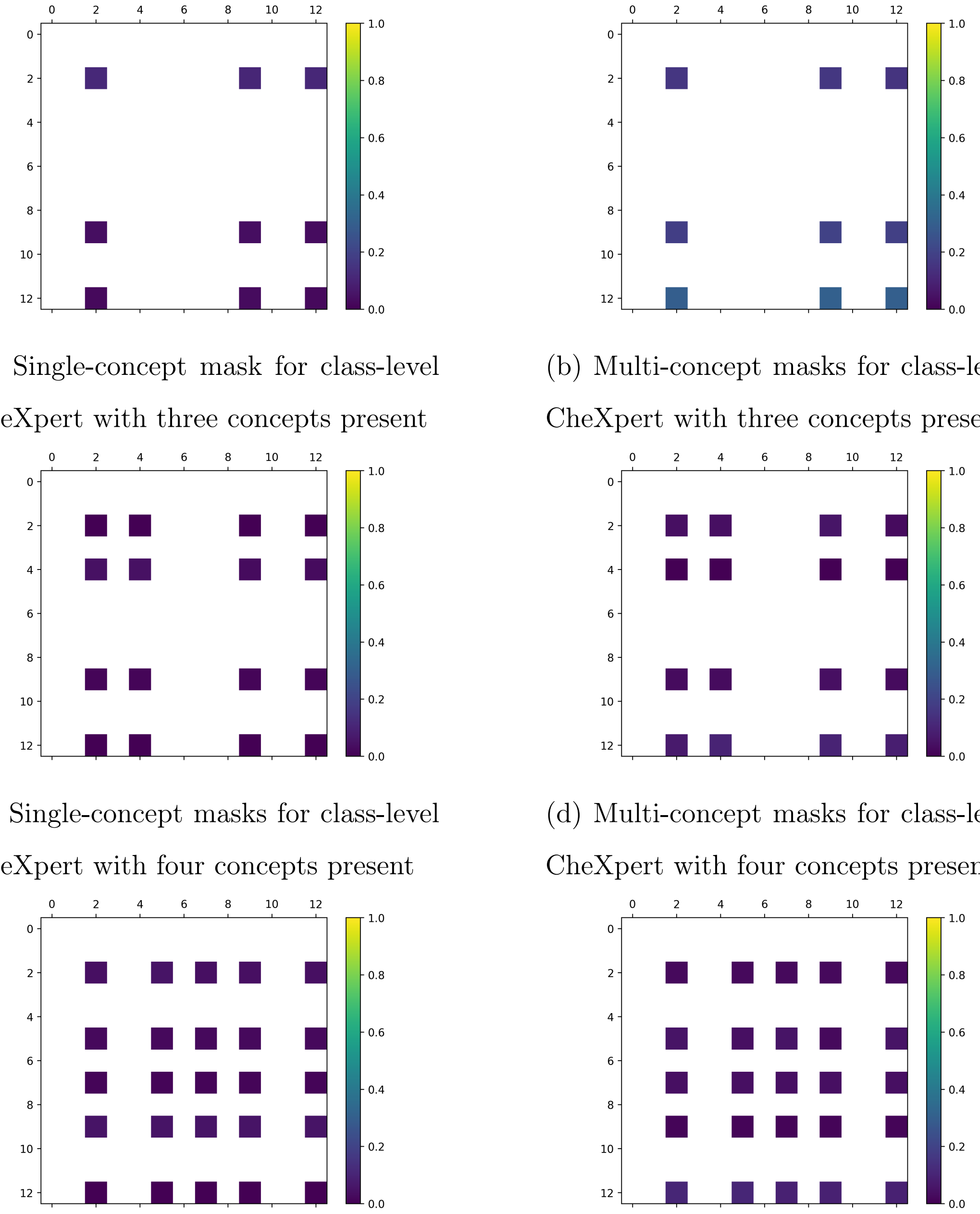
(a) Instance-level CheXpert single concept masks

(b) Instance-level CheXpert multi-concept masks

Figure 4.16: Concepts for models trained on Instance-level CheXpert demonstrate that most predictions do not change from present to not present, regardless of the concepts masked. However, some concepts such as "support_devices" (concept 12) do not follow this trend as removing semantically meaningful input features reduced accuracy, while other concept predictions remain unaffected.

present to not present prediction, both for semantically meaningful concept masks and unrelated concept masks. For semantically meaningful concept masks, this suggests that there is duplicate information in the input images that allows the model to maintain accurate predictions when masks are applied. This is supported by CRA results for concept the concept "support_devices" (concept 12). "Support_devices" is the only non-organic based concept which also has distinctive shapes and features that the models may learn to identify with ease compared to the other concepts. Removing semantically meaningful input features for this concept significantly impacts its accuracy. In addition, other concept predictions remain unchanged when this concept is masked.

Multi-concept concept masks for CheXpert (Figure 4.16b), show that when all but one concept is masked, the model changes concept predictions from present to not present around 50% of the time for less than 25% of concepts. As expec-



(a) Single-concept mask for class-level CheXpert with three concepts present

(b) Multi-concept masks for class-level CheXpert with three concepts present

(c) Single-concept masks for class-level CheXpert with four concepts present

(d) Multi-concept masks for class-level CheXpert with four concepts present

(e) Single-concept masks for class-level CheXpert with five concepts present

(f) Multi-concept masks for class-level CheXpert with five concepts present

Figure 4.17: Concepts for models trained on Class-level CheXpert are not sensitive to the removal of semantically meaningful input features, while also maintaining concept accuracy when all other input features are masked. These models highlight the models have identified multiple sources of input features can be used to predict each concept.

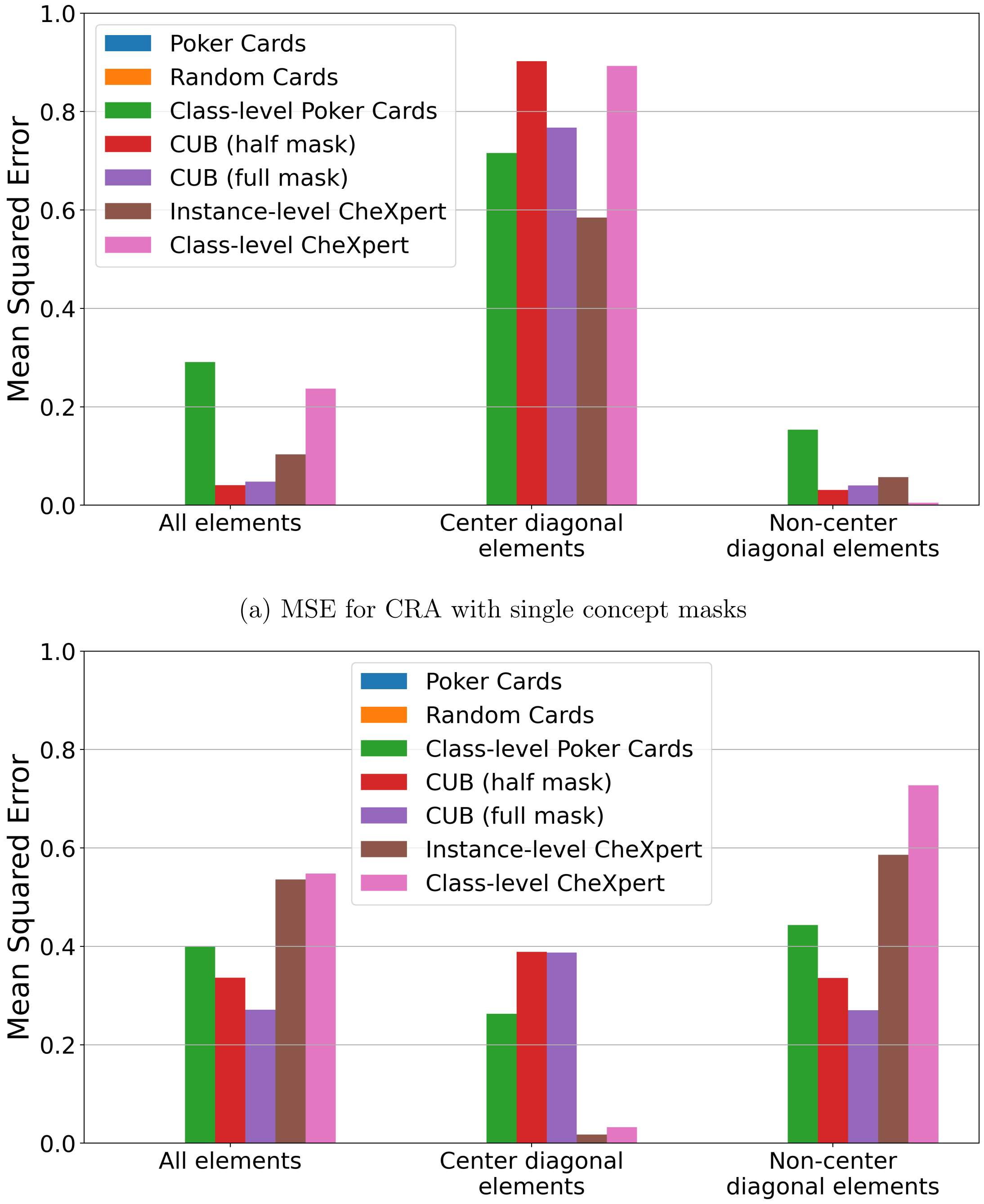
ted, this shows the models are less accurate at predicting more concepts than we observed with single-concept masks. As most concept input features are masked there is an increased likelihood of masking all relevant features for each concept being measured. The centre diagonal of concepts is also showing concepts remaining almost unchanged. This demonstrates that the models have learned the importance of specific input features for concepts. However, when we consider both masking types, it is clear the models have learned to use multiple sources of input features to maintain accurate concept predictions.

4.7.2.4 Concept Bottleneck Models can respect concept locality

To quantitatively measure a model's sensitivity to concept input features, and thus answer RQ2 Sub-question 2, we compared single-concept mask matrices to an identity matrix, and multi-concept masks to a complement identity matrix. A high MSE between these matrices and their respective identity matrices indicates that the model has not learned to predict concepts using semantically meaningful input features. Since MSE is sensitive to the balance between diagonal and off-diagonal elements, there are more off-diagonal elements than diagonal ones, we separated these groups of elements to evaluate two key behaviours:

1. How much does the accuracy of concepts fall when their corresponding input features are removed.
2. How resilient concept predictions are when unrelated input features are masked.

Figure 4.18a shows the MSE for single-concept masks. The Random and Poker card models achieve near-zero MSE across all elements. In contrast, the Class-level Poker card model showed significantly higher MSE, with values over 0.7 for diagonal elements and just under 0.2 for off-diagonal elements. These results quantify the misalignment for concept prediction sensitivity being tied to



(a) MSE for CRA with single concept masks

(b) MSE for CRA with multi-concept masks

Figure 4.18: MSE for CRA shows models trained on instance-level consistently achieve lower MSE compared to models trained on class-level concepts.

semantically meaningful input features. Unrelated input features do not show as large of a misalignment. This shows the models are somewhat resilient to the removal of unrelated input features.

For CUB models and CheXpert models, the MSE values indicate a similar misalignment, with semantically meaningful features often ignored for their respective concept predictions, while unrelated input features resulting in a low MSE indicate concepts are sensitive to the removal from unrelated input features. As Instance-level CheXpert models MSE value is between 0.5 and 0.6, these models show concepts are sensitive to the masking of semantically meaningful input features just over 40% of the time. The same is not observed with CUB models where we observed only 10% to just over 20% of concepts changing from present to not present prediction after semantically meaningful input features were masked.

The MSE for multi-concept masks is shown in Figure 4.18b. As with single-concept masks, the Random and Poker card models achieve very low MSE, demonstrating robust alignment with semantically meaningful input features and robustness against unrelated input features. Class-level Poker card models show a significantly higher MSE compared to the Instance-level Playing card models for both centre diagonal elements and non-centre diagonal elements, further confirming a lack of alignment in concept feature sensitivity.

Equally, CUB models also reflect misalignment between semantically meaningful input features and concept predictions, and robustness against unrelated input features. In the case of CheXpert models, class-level models show a high MSE for off-diagonal elements. In comparison, Instance-level CheXpert models show reduced diagonal MSE, highlighting they are more sensitive to masking of relevant input features compared to models trained on the class-level dataset.

The MSE for centre diagonal elements with multi-concept masks is consistently lower than that for single-concept masks. This indicates that models can rely on semantically meaningful input features to make accurate concept predictions when all other concept regions are masked. However, the higher MSE observed in single-concept masks continues to demonstrate that these same models are also using unrelated input features to predict concepts. In other words, models with high MSE under single-concept masking but low MSE under multi-concept masking have learned redundancy for which input features are required for concept predictions. For instance, a model predicting the concept of a bird wing might use semantically meaningful input features of the wing when available (contributing to a low MSE for multi-concept masks) but also rely on input features for the bird's body or head in the absence of wing pixels (contributing to a high MSE for single concept masks).

Overall we have observed CBMs are sensitive to the removal of semantically meaningful input features and are resilient to the removal of irrelevant input features when the training data includes a consistent mapping between input semantically meaningful input features and concept annotations. This follows the same dataset attribution that allows a model to learn to predict concepts using semantically meaningful input features, as identified in Chapter 3. We found that models trained on instance-level concepts consistently achieved a lower MSE compared to models trained on class-level concepts. Secondly, concept sensitivity to input features is not significantly affected by the addition of limited inter-concept correlation in the dataset as evidenced by the performance of the Random and Poker card models, both of which achieve a low MSE. Finally, In line with prior research (Raman et al., 2024), we find that high concept accuracy does not always correlate with models using semantically meaningful input features for concept predictions, as evidenced by both models trained on Class-level poker cards and CUB.

4.8 Discussion

This chapter addresses RQ2: "How does the configuration of concepts in the training dataset affect encoded information in learned concepts and input feature dependence for concept predictions?". Overall, as with predicting concepts using semantically, meaningful input features in Chapter 3, we find the dataset used for CBM training affects how the model learns concept representations. CBMs can be trained such that additional information encoded into individual concepts is minimised, and concept accuracy cannot be manipulated with the addition or removal of irrelevant input features. Expanding RQ2, we introduced two sub-questions:

1. How does the configuration of concepts in the training dataset affect information leakage of learned concepts?
2. How does the configuration of concepts in the training dataset affect input feature dependence?

4.8.1 How Does the Configuration of Concepts in the Training Dataset Affect Information Leakage of Learned Concepts?

Using OIS we revealed the purity of concept predictions made using CBMs trained on our three datasets. By analysing these results we revealed the capability for CBMs to learn the underlying correlation of concepts in their training data. If the dataset includes correlation between concepts then a CBM is capable of learning this correlation in their concept predictions. This means CBMs will suffer from information leakage if the training dataset does not restrict it.

For a CBM to be trained such that information leakage is minimised concepts in the dataset should be balanced both in occurrence and in co-occurrence with other concepts. We showed CBM learned concepts were able to predict ground truth concepts similar to the Pearson correlation coefficient of the concepts in the training data. A simple method to minimise unbalanced co-occurrence of concepts in a dataset would be to balance dataset samples according to the Pearson correlation coefficient. Ensuring a balanced dataset is more important when there are fewer concepts in a dataset, as demonstrated with Class-level poker cards compared to CUB, as it allows for indirect relationships between concepts to be learned by the model.

4.8.2 How Does the Configuration of Concepts in the Training Dataset Affect Input Feature Dependence?

Our analysis of input feature sensitivity using CRA showed the correlation of concepts in a CBM's training dataset had a limited impact. Small correlations between concepts, such as Poker cards and Random cards, did not lead to a significant difference between the trained models CRA. On the other hand, a high correlation between concepts does have an impact on input feature dependence. For our models trained on datasets with class-level concept annotations, concepts were both insensitive to the removal of relevant input features and sensitive to the removal of irrelevant input features.

4.8.3 Concept Bottleneck Model Training Best Practices

Following the results shown both in this chapter and in Chapter 3 we have provided some best practices to train a CBM which predicts concepts using semantically meaningful input features, minimises concept leakage, and where concepts are not sensitive to erroneous input features:

- Concept correlation: Concept annotations should not have a correlation between concepts unless that correlation is intended or required. As shown by the oracle and purity matrices, unintended concept correlation can lead to unrelated concepts accurately predicting each other. In extreme cases, it can obscure which input features are semantically meaningful, as seen in Chapter 3 with saliency maps for models trained with class-level concepts. However, some correlation between concepts will not significantly degrade model performance, as evidenced by the results comparing Random cards and Poker cards. Since our evaluation did not cover an extensive range of concept correlations, we leave a more thorough investigation to future work.

- Ensuring concept annotations and visualisations are consistent: Unlike previous studies, we restricted some of our datasets so we could ensure concepts' visual representations were present in sample images. Following this requirement helps to ensure there is a clear training signal for CBMs to learn semantically meaningful concept representations.

We also recommend the use of instance-level concept annotations over class-level concept annotations. Although we show training a CBM to map input features to concepts semantically is possible with class-level concept annotations, such as the concepts “Four of Spades” and “Four of Clubs”, we demonstrate it's far easier to achieve semantically meaningful concept mappings with instance-level concept annotations.

4.9 Limitations

4.9.1 Dataset

This chapter uses the same datasets as we used in Chapter 3. While these datasets were sufficient for our analyses, their constraints limited the generalisability of our findings. For instance, the datasets used feature fixed levels of correlation between concepts, which restricted our ability to analyse how fine-grained variations of concept correlation affected a model's ability to reduce feature sensitivity to irrelevant concepts. Incorporating a dataset with fine-grained concept correlation adjustments would have enabled this additional analysis. This may be achieved with synthetic datasets such as Playing cards by systematically limiting which cards may appear together. However, achieving this with real-world datasets poses a large challenge as it adds further constraints to data collection in addition to those already imposed by the need for concept annotations.

Furthermore, the inclusion of an additional real-world dataset with instance-level concepts could have provided analysis to enhance the robustness of our findings with models trained on CheXpert.

4.9.2 Methods

We analysed information leakage with the OIS metric which is a measurement of concept purity. A core part of this metric is using the AUC value. This metric presented limitations when applied to the Class-level CheXpert dataset. Specifically, the AUC calculation requires that every concept in the dataset be present at least once. Due to this constraint, we were unable to measure information leakage for this dataset variation.

4.10 Summary

In this chapter, we extend our analysis of CBMs w.r.t. information encoded into concept representations, and reliance on semantically meaningful input features to make concept predictions. We start by looking at the correlation of concept occurrences in the datasets before using metrics that measure information leakage and concept correlation in our trained CBMs. We continue to use three datasets: CUB, Playing cards, and CheXpert and as such have measured how different dataset concept configurations can change how the models learn to represent concepts.

As in Chapter 3, the introduction of our Playing cards dataset shows the capability of CBMs if the dataset only has concepts represented by semantically meaningful input features. If a CBM makes predictions using semantically mapped input features, we can argue the model will be easier to build trust with as concept predictions will use the input features that match human expectations.

This chapter answers RQ2 ("*How does the relationship between concepts and input features in the training dataset influence the information encoded in learned concepts and the model's reliance on input features for predicting those concepts?*"). CBMs are reliant to the configuration of concepts in the dataset. We show CBMs are capable of learning to encode concepts inline to the inter concept correlation of concept in their training data, and instance-level concept annotations are required for predictable input feature dependence.

We show how learned concept representations can encode correlations between concepts from the dataset which may lead to concepts being predicted based on the presence or absence of other concepts. If this is to be avoided the correlation of concepts in the dataset should be considered and can be mitigated by splitting the training of the two model parts to learn concepts and the downstream task independently, allowing the dataset to train the concept encoder can be balanced separately than the dataset used to train the task predictor.

Finally, we also analyse how concept accuracy changes when concepts are masked. We show instance-level concepts are vital for ensuring concept accuracy changes in a predictable manner i.e. only the semantically meaningful concepts change from a present prediction to a not-present prediction when masked. We also show the correlation of concepts in a model's training data plays a minimal role in ensuring concepts are predicted using semantically meaningful input features.

Chapter 5

The Impact of Concept Explanations and Interventions on Human-machine Collaboration

5.1 Introduction

After a CBM makes a prediction a human collaborating with the model will be able to inspect the concept explanations to help understand the model's decision-making process, making these models inherently interpretable. In domains such as healthcare this may be used to answer why a downstream task was predicted. The concept explanations also introduce the capability for a human collaborator to intervene in the concept predictions and inspect how these change the downstream task prediction. A human collaborator can correct mistakes the model made when predicting concepts, or otherwise ask what-if questions in regards to the model downstream task prediction if the model had a different set of concept predictions.

In Chapters 3 and 4 we used XAI and disentanglement metrics to show what configurations of concept annotations are required for datasets to confine CBMs to learn concepts such that concepts are predicted solely using semantic meaningful input features and with minimal information leakage. By using these findings to train CBMs we have argued these models are capable of meeting their original promise of interpretability. In the paper introducing CBMs, the authors made claims of improved human collaboration (Koh et al., 2020), but human studies to show this are limited and instead compare CBMs to other model architectures

(Jeyakumar et al., 2023; Dubey et al., 2022), or complete tasks such as selecting the concepts participants believe the model detected Wang et al. (2023a). Only a few studies analyse the class of CMs with collaborative tasks (Mysore et al., 2023; Nguyen et al., 2024).

This chapter presents two human studies where we analyse CBMs in a collaborative setting. We answer **RQ3** ("*Do Concept Models improve task accuracy and model interpretability in a human-machine setting?*"). We have broken this questions down into the following sub questions:

1. Do test-time interventions improve human task and concept accuracy?
2. Do interventions increase the interpretability of CBMs?
3. Are CBMs trusted?

By answering RQ3 we make the contributions RC6 and RC7:

- **RC6**: We perform the first human studies using CBMs in a joint human-machine task setting which analyses the interaction between humans and the CBM. We find participants who performed interventions increased trust in a model, but this trust was sometimes misplaced. Additionally, the CBM decision-making process is not aligned to that of the humans.
- **RC7**: We show providing concept explanations to humans increases both model interpretability and task accuracy. In addition, interventions can be used to reveal model bias. This upholds the model's promise of increasing interpretability from high-level concepts.

Our first study is in the domain of Dermatology with experts in that field. Our second study asked participants to play games of blackjack and involved participants who were not necessarily experts at playing the game. In both studies the CBM is acting in an advisory role. This chapter contains work in our paper "*The Impact of Concept Explanations and Interventions on Human-machine collaboration*".

Below is a list of acronyms and their meanings that are predominantly used in this chapter.

Acc Accurate model

CExp Concept Explanation

CExp+Int Concept Explanation with Interventions

CExp+Int+SMap Concept Explanation with Interventions and Saliency maps

DDI Diverse Dermatology Images

Inacc Inaccurate model

NoExp No Explanations

NoInt No Interventions

SUS System Usability Scale

WithInt With Interventions

5.2 Motivation

In this chapter, we look at the research gap analysing the original promises of interventions improving model task accuracy and CBM interpretability made in (Koh et al., 2020). We achieve this by running two human studies, both of which use a CBM as an AI assistant to suggest the action a human participant should take. Running human studies means we can evaluate human interaction which cannot be automated with non-human metrics (Yadav et al., 2019; Rong et al., 2024; Schmutz et al., 2024).

CBMs are positioned as being inherently interpretable, but as discussed in Chapter 2, this primarily comes down to two aspects of the model architecture and capabilities. Firstly, the models predict high-level concepts which are argued as aligning the model with a human understanding of the same task. This allows a human to understand what contributing components a model detects for a task label prediction. Secondly, interventions will enable humans to ask "what if these concepts were predicted instead?" This counterfactual explanation capability is argued to reveal the model's decision-making process. Linking back to (Miller, 2019), interventions can be considered as contrastive explanations as humans can modify concept values to understand why one task label was predicted instead of another. From the CBM capabilities mentioned here, if automated metrics are to be believed, we may expect improved model accuracy as incorrect concept predictions are replaced with their correct values from interventions. Further, if the model task accuracy improved from intervention, trust in the models may increase. In addition, as bias can be identified from model predictions (Adebayo et al., 2020), CBMs may help humans identify model bias by inspecting updated task label predictions.

From the literature discussed in Chapter 2, we identified two papers, (Mysore et al., 2023; Nguyen et al., 2024), that asked participants to interact with a CM to complete a task, and no papers that looked at evaluating the interpretability or capabilities of CBMs. This leaves human interaction and CBM capabilities unexplored apart from automated metrics that, as just discussed, may not be a good representation of human interaction. Further, in a different human study, it was found that the concepts a model uses for task predictions may not be the same as those humans would use (Barker et al., 2023). It remains unknown if this applies in a collaborative setting and if it impacts human-machine team accuracy.

Although our findings from Chapter 3 and Chapter 4 suggest CBMs can make

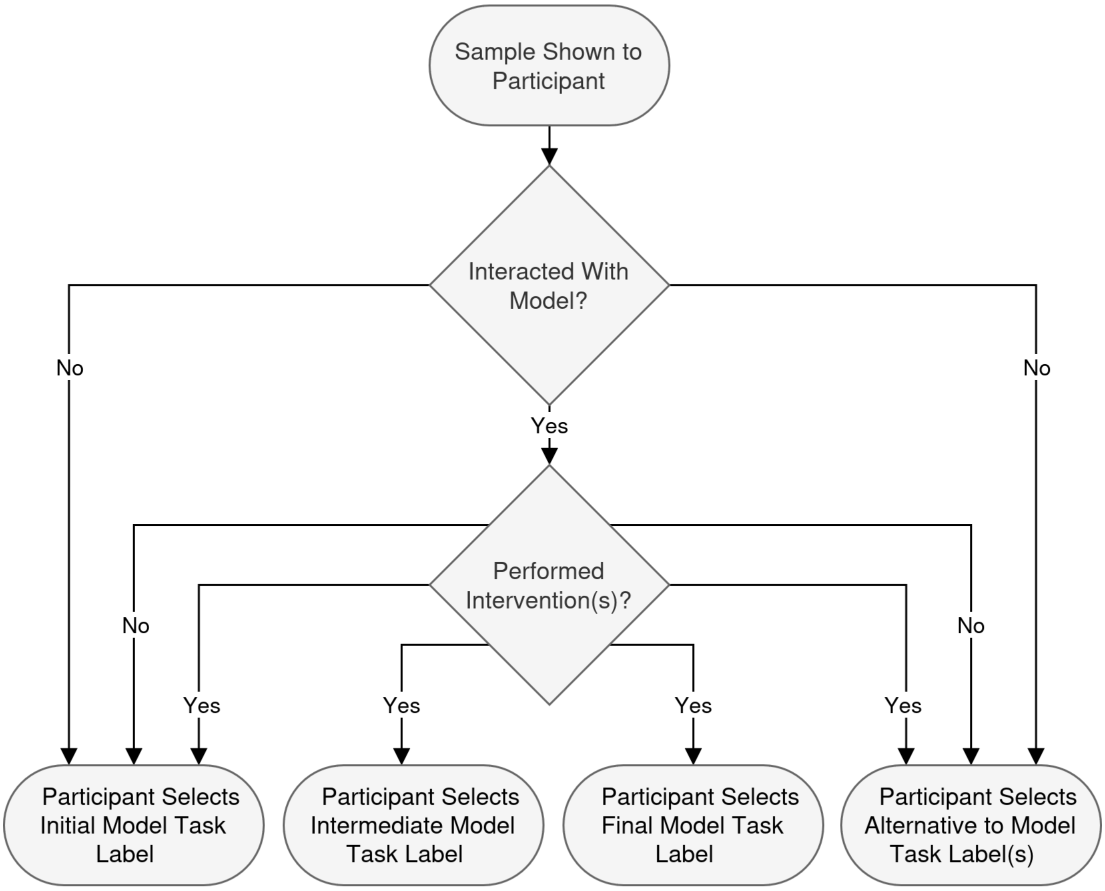

Sample Shown to Participant
Interacted With Model?
No
No
Yes
Performed Intervention(s)?
No
No
Yes
Yes
Yes
Yes
Participant Selects Initial Model Task Label
Participant Selects Intermediate Model Task Label
Participant Selects Final Model Task Label
Participant Selects Alternative to Model Task Label(s)

Figure 5.1: Flow chart to show the interaction participants in our studies can have with a model.

predictions aligned with human decision-making, we need to evaluate the models with real humans to verify they are indeed interpretable. Therefore, this chapter is focused on exploring this research gap. Further, we have the question about how humans interact with CBMs in general. This includes how their capabilities compare to standard DNNs, and how humans perform interventions.

We created a flow chart to show the interaction participants can have with a model before labelling samples in Figure 5.1. Participants have the option of interacting with the model, performing interventions (participants can either update predicted concept values or leave them unchanged), and labelling the sample. When labelling the sample they may select the initial model task label before interventions, the final model task label prediction, an intermediate task label prediction, or a task label the model did not predict. This flow chart helps to answer Sub-question 2 and Sub-question 3 as it can highlight if human selected task labels are aligned with the model because of model interaction, or just by chance.

Although we used CBMs, all of our findings also apply to other CMs with intervention capabilities and that predict task labels in the same feed-forward fashion from input to concepts, to task label. Namely these are Concept Embedding Models Zarlenga et al. (2024), Sidecar CBMs Lockhart et al. (2022), and hybrid CBMs Mahinpei et al. (2021).

5.3 Related Work

In Section 2.3.3 we discussed human studies evaluating CMs. Here we have reintroduced these studies and expand on the literature to inform the direction and design of our studies. Several previous studies have analysed CBMs and similar model architectures with human participants. These can be placed into several categories; human concept preference (Barker et al., 2023; Ramaswamy et al., 2023), concept explanations (Jeyakumar et al., 2023, 2022; Wang et al., 2023a; Sixt et al., 2022; Dubey et al., 2022), human-in-the-loop (Mysore et al., 2023; Nguyen et al., 2024) and bias discovery (Yuksekgonul et al., 2023; Midavaine et al., 2024). A summary of studies is provided in Table 5.1.

5.3.1 Human Concept Preference

Barker et al. (2023) explores human concept selection with the dataset CUB. In their experiment they asked participants to select concepts they thought were relevant to the downstream task classification, finding a large variance in the number of concepts selected. These concepts also performed worse when used by a model predicting the downstream task than concepts the model selected.

Paper	Category	Number of participants	Study aim	Key findings
Barker et al., 2023	human concept preference	30	Identify concepts participants select as relevant	Participants selected concepts varied and were less effective
Ramaswamy et al., 2023	human concept preference	125	Explore concept reasoning capacity	Participants are slower and less accurate with more concepts
Jeyakumar et al., 2023	concept explanations	75	Explore explanation format preference	With time-series-data concept explanations ranked lower than other formats
Jeyakumar et al., 2022	concept explanations	-	Explore explanation format preference	With video data concept explanations were preferred
Wang et al., 2023a	concept explanations	50	Test if participants understand model-learned concepts	Participants understood learned concepts
Sixt et al., 2022	concept explanations	240	Identify important features via explanations	Concept explanations performed poorly
Dubey et al., 2022	concept explanations	150	Predict model decisions from explanations	CBMs performed 5% worse than their custom model
Mysore et al., 2023	human-in-the-loop	20	Evaluate recommendation quality post-intervention	Interventions improved recommendations (20–47%)
Nguyen et al., 2024	human-in-the-loop	150	Compare static vs dynamic explanations	Little difference in performance
Yuksekgonul et al., 2023, Midavaine et al., 2024	bias discovery	30	Test participant ability to fix CBM bias	participants detected and corrected model bias

Table 5.1: Summary of human studies. Number of participants is provided where available.

Barker et al. (2023) believed this was because the concepts humans selected were generic, and were not informative to the downstream task.

Ramaswamy et al. (2023) evaluates concept explanation complexity, the number of concepts a human can reason with. In their study, they asked participants to (1) select concepts they thought were present in an input, and (2) select the concept explanations they prefer when the explanations have varying numbers of concepts. Ramaswamy et al. (2023) made two key findings relevant to our studies. Firstly, by showing more concepts, participants took longer to recognise them and were generally worse at doing so. The time increase is expected as there are more concepts to make a judgement for, but the decreased performance in recognising concepts present in a sample shows humans are better at making fewer judgements. The author's second finding was that the preferred number of concept explanations was 32 or less. As many datasets have more than 32 concepts, CUB, to name one, we need to ensure either fewer concepts are used in our models, or an optimal selection technique is used to maximise the usefulness of concept explanations while remaining below this threshold.

5.3.2 Concept Explanations

Explanation preference is a common theme for CBM human studies (Jeyakumar et al., 2023, 2022; Sixt et al., 2022; Dubey et al., 2022). These compare CBM concept explanations with other explanation types and model architectures. Jeyakumar et al. (2022) ran a study asking participants about their preferred explanation format for activity recognition from videos. The authors found the top two explanation methods were concept explanation with and without attention, where attention is how important a concept is to a downstream task prediction. This study shows promise for concept-based explanations.

Sixt et al. (2022) conducted a human study to evaluate explanation techniques for bias discovery. They found that concept explanations performed poorly, achieving accuracy below a random guess. However, since the model used was not a CBM and the concepts were discovered automatically, this finding may not translate over to concepts in CBMs. Similarly, Dubey et al. (2022) compared CBM concept explanations with posthoc explanations and their own method, in a study participants guessed the model's downstream task prediction based on the explanations provided. Despite a training phase designed to help participants build a mental model of the artificial agent, CBMs performed approximately 5% worse than the authors' method. However, the dataset used (CUB) is known to be unsuitable for CBM training, as discussed in Chapter 3. Lastly, Jeyakumar et al. (2023) examined human preferences for explanation formats in a model trained on input sensors. Participants consistently ranked CBM explanations as the least preferred, though this finding may not generalise to other modalities due to the author's use of non-image, time-series data.

Besides concept preference, one study asked participants to identify what concepts a model had learned (Wang et al., 2023a). Although the authors did not use a CBM, their model architecture is similar. They found their model was able to learn human identifiable concepts, and with saliency maps annotations, their model was close to the performance of manual human annotated samples.

5.3.3 Human-in-the-Loop

Out of all the studies we found, only two asked participants to interact with a model to help perform a task (Mysore et al., 2023; Nguyen et al., 2024). In (Mysore et al., 2023) the authors introduce a recommender system inspired by CBMs and trained on text. A human provides an input of documents similar to what they are looking to be recommended. The model then computes concepts from the provided documents, in addition to human-provided concepts, before recommending similar documents. The human can interact with the concepts the model uses to adjust the recommendations, similar to interventions with CBMs.

In their human study Mysore et al. (2023) found their artificial agent to be effective at improving the recommendations participants received by 20-47% after tuning by the artificial agent compared to initial concept values. This was better than a distance-based recommendation agent that they compared to, which improved recommendations by 18-40% from the initial recommendations. The initial recommendations from the author's artificial agent were also better than the comparison agent.

Nguyen et al. (2024) examined the effectiveness of explanations generated by the CHM-Coor model architecture (Taesiri et al., 2022), a visual correspondence-based classifier that divides input images into patches before predicting a task class. Participants were shown static or dynamic explanations alongside the model's predictions and asked to accept or reject them. Static explanations supported samples with patch annotations, while dynamic explanations allowed users to adjust the model's focus by selecting patches for re-analysis by the model. The study found little difference in accuracy between the dynamic group (73.57%) and the static group (72.68%), both far below perfect accuracy. Additionally, participants often agreed with the model's predictions regardless of if the model was correct or incorrect.

5.3.4 Bias Discovery

Using a model architecture similar to a CBM, the authors in (Yuksekgonul et al., 2023) ran a human study, which was repeated by Midavaine et al. (2024), where they asked participants to improve a model performance when the task data has been shifted from the training data. Participants were shown scenarios where a class had been shifted and had to select a subset of concepts to prune from the model given input and concept predictions.

Improvements in accuracy from human pruning showed to be better than random pruning and only mildly worse than fine-tuning and greedy pruning. Considering human pruning does not require knowledge or access to the training data, this shows the potential of using humans-in-the-loop for situations similar to this.

5.4 Methods

In this chapter we answer RQ3: "How do the claimed improvements to task accuracy and counterfactual explanations through the use of CBM interventions translate to a human-machine settings where the model is used as an assistance to a human operator?" This is broken down into three sub questions that we introduced in Section 5.1.

In our studies, we used CBMs although, as the studies only reveal the model outputs and capabilities, our findings are also applicable to any CM architecture that supports task predictions from concepts and interventions.

We ran two human studies: (1) An expert study where participants had extensive knowledge about the task domain (skin disease diagnosis) where the model acted as a second opinion. (2) A lay-person study with a general task (Playing games of Blackjack), involving participants with experience levels ranging from novices to skilled individuals, but none being professionals. The model also acted as a second opinion, but could also serve as a guide for participants with less experience.

By running two studies we were able to compare results in these two similar, but distinct settings. Both of the studies require participants to complete a task with an AI agent they can use to assist them. Following the taxonomy by (Doshi-Velez and Kim, 2017), our lay-person study is human-grounded as we do not use expert participants and use a simulated task, while the expert study is application grounded as we use both expert participants and a real-world task.

For our studies, we split participants into several groups which are detailed in Table 5.2. Due to the smaller number of participants, the expert study had two groups, whereas the lay-person study had eight groups. Returning to our

Expert study	Lay-person study	
Participant groups	Participant groups	
	Accurate model (Acc)	Inaccurate model (Inacc)
CExp+Int	NoExp	NoExp
CExp+Int+SMap	CExp	CExp
	CExp+Int	CExp+Int
	CExp+Int+SMap	CExp+Int+SMap

Table 5.2: Participants in the expert study were split into two groups, both with access to the same model. Participants in the lay-person study were split into eight groups where the model used and explanations provided were varied.

sub-questions, these require us to analyse the use of interventions and the interpretability and trust of CBMs. Therefore, both groups for the expert study included participant access to interventions and instead we varied the model's explanation completeness (Kulesza et al., 2013). As we did not have the same limitation on the number of participants in the lay-person study we also included groups that had access to a different model and included groups with no model explanations and just concept explanations.

We use the following acronyms to separate each participant group:

Acc Accurate model (lay-person study only).

Inacc Inaccurate model (lay-person study only).

NoExp No explanations (lay-person study only).

CExp Predicted task label and concept explanations (lay-person study only).

CExp+Int Concept Explanation (CExp) plus Intervention capability.

CExp+Int+SMap CExp+Int plus Saliency maps.

WithInt Participants who performed interventions or samples where interventions were performed.

NoInt Participants who did not perform interventions or samples where no interventions were performed.

Acc and Inacc are placed before the model output and feature capabilities. With Interventions (WithInt) and No Interventions (NoInt) are placed after model output and feature capabilities. For example, participants using the accurate Blackjack model with concept explanations and interventions, and who performed interventions would be referred to as Acc-Concept Explanation with Interventions (CExp+Int)-WithInt.

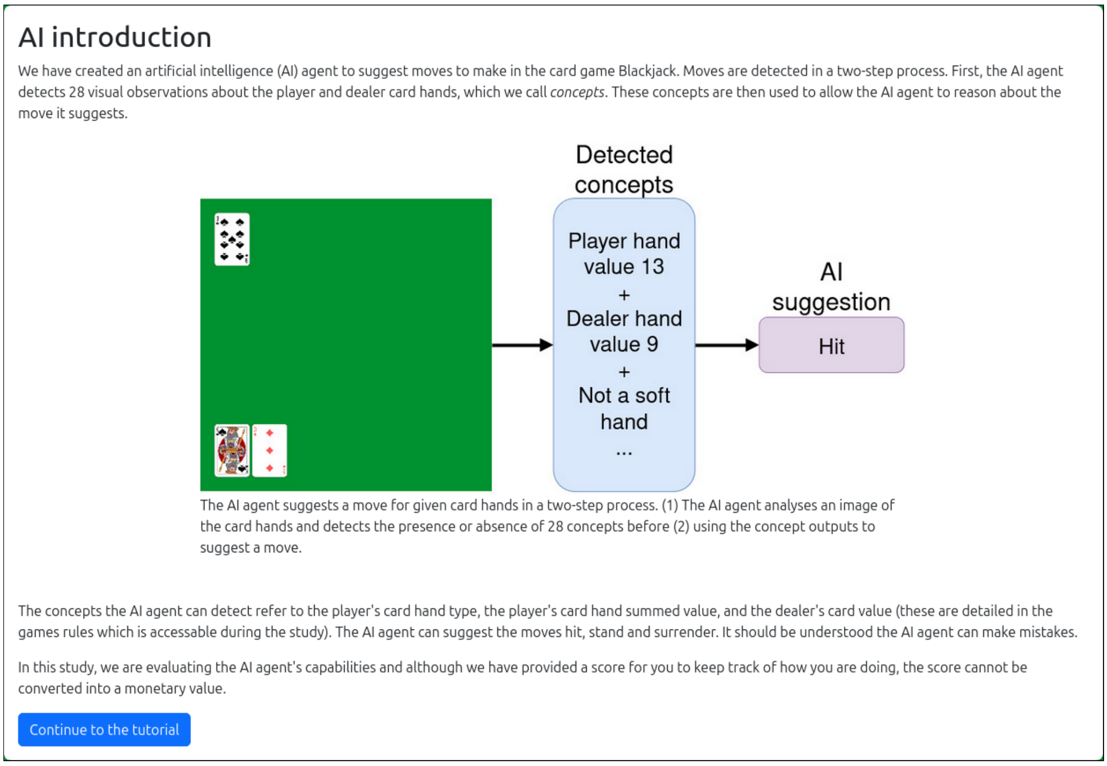
5.5 Experiment Set-up

Our first human study looked at participants with expert knowledge in dermatology. This allowed us to evaluate a CBM in a setting where the human will have the required expertise to complete the task without model input, but the CBM will act as a second opinion.

The second study did not need participants to have any specialist knowledge and as such some participants will have minimal knowledge about Blackjack while others will have good knowledge of the game. The model was made available to all participants but did not require participants to interact with it.

The studies[1] share a lot of similarities including participants are not required to have any knowledge about computer science, AI, or XAI. The studies also share a similar interface, which includes reducing the number of concepts initially shown to participants as recommended by Ramaswamy et al. (2023). We also ordered concepts by prediction value and hid additional concepts in a scrollable list.

[1] Additional details about the study designs and ethics are included in the Appendix in Section B.3



AI introduction

We have created an artificial intelligence (AI) agent to suggest moves to make in the card game Blackjack. Moves are detected in a two-step process. First, the AI agent detects 28 visual observations about the player and dealer card hands, which we call *concepts*. These concepts are then used to allow the AI agent to reason about the move it suggests.

The AI agent suggests a move for given card hands in a two-step process. (1) The AI agent analyses an image of the card hands and detects the presence or absence of 28 concepts before (2) using the concept outputs to suggest a move.

The concepts the AI agent can detect refer to the player's card hand type, the player's card hand summed value, and the dealer's card value (these are detailed in the games rules which is accessable during the study). The AI agent can suggest the moves hit, stand and surrender. It should be understood the AI agent can make mistakes.

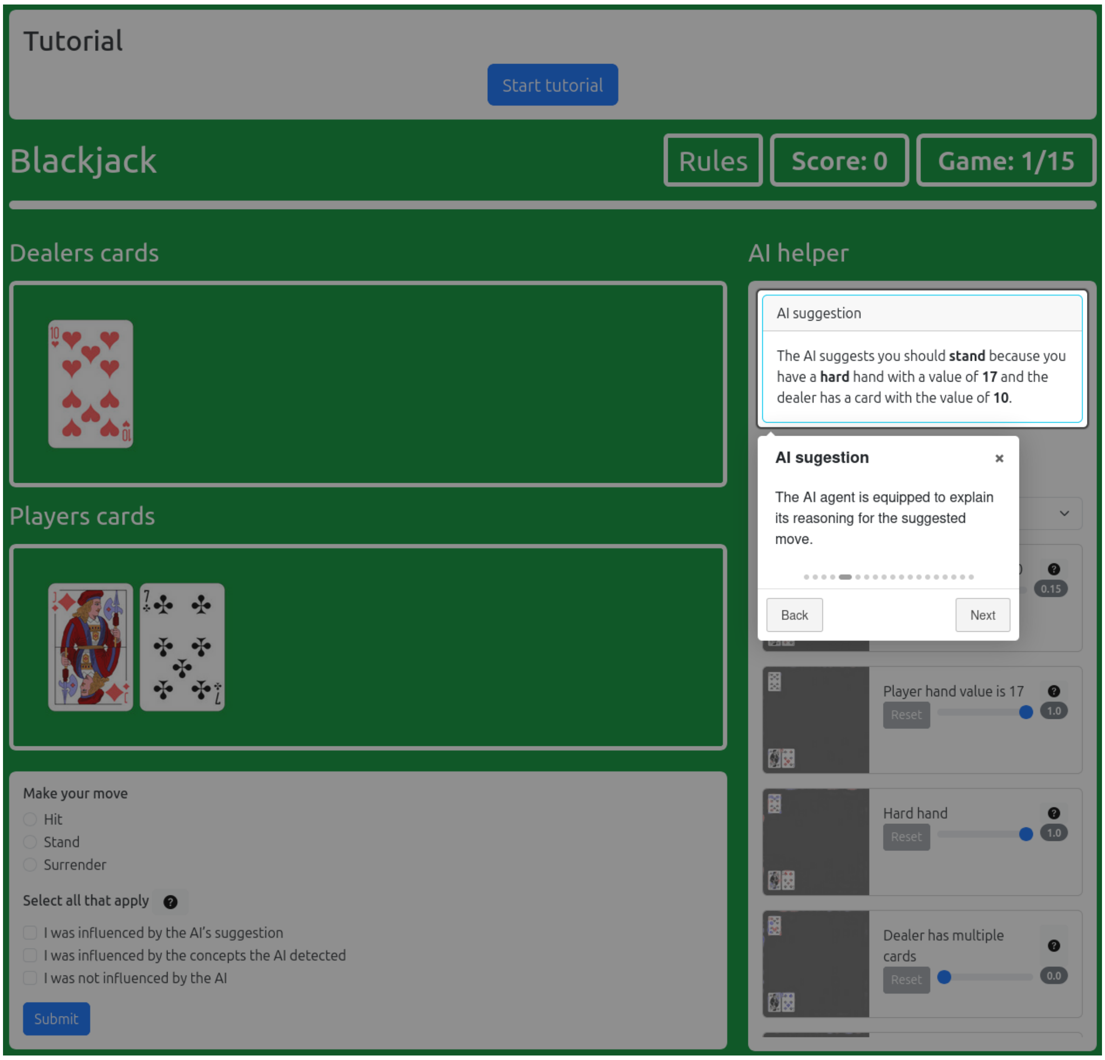
In this study, we are evaluating the AI agent's capabilities and although we have provided a score for you to keep track of how you are doing, the score cannot be converted into a monetary value.

Figure 5.2: Example AI brief show to participants.

Both studies start with a short demographic survey asking participants' for their age, gender, computer science experience and skin disease identification/blackjack experience. Computer science experience and skin disease identification/blackjack experience are recorded using a Likert scale (Likert, 1932). Following the demographic survey participants were briefed on how the model works (see Figure 5.2 for an example from the lay-person study) and followed a tutorial so they know how to participate in the study and interact with the model (see Figure 5.3 for an example from the lay-person study). Following this participants completed the study task, and finally completed a closing survey.

5.5.1 Expert Study

Throughout the design and development of the expert study we consulted with a dermatologist from the University Hospital of Wales. Their feedback enabled



AI suggestion

The AI suggests you should **stand** because you have a **hard** hand with a value of **17** and the dealer has a card with the value of **10**.

AI sugestion ×

The AI agent is equipped to explain its reasoning for the suggested move.

Make your move

○ Hit
○ Stand
○ Surrender

Select all that apply

□ I was influenced by the AI's suggestion
□ I was influenced by the concepts the AI detected
□ I was not influenced by the AI

Submit

Figure 5.3: Example study tutorial shown to participants.

us to refine the study and find participants. Regarding the study design, their feedback lead to two key developments:

1. **Study task**: We initially designed a classification task, but this was changed to a diagnosis task. This change is inline with the typical objectives doctors will perform in their job.

2. **Sample image selection**: The initial sample images we selected from the dataset had varying resolutions, some were out of focus, and they contained a mixture of skin diseases. With the guidance from the dermatologist we

selected images that were of higher quality, in focus, and had a limited focus of skin diseases to include.

5.5.1.1 Dataset

Skincon (Daneshjou et al., 2022b) is a real-world image dataset with 48 clinical concepts, of which we have kept 22 that have 50 or more occurrences in the dataset. Concepts were selected by two dermatologists using standard descriptive terms such as “plaque” and “scale”. We have provided an example sample with concept annotation in Figure 5.4. Skincon was created by combining two datasets: Fitzpatrick 17k (Groh et al., 2021a) and Diverse Dermatology Images (DDI) (Daneshjou et al., 2022a), with Skincon adding concept annotations. Skincon has instance-level concepts and contains 3886 images, 3230 from (Groh et al., 2021a) and 656 from (Daneshjou et al., 2022a). For downstream task labels, we use the malignant label which is provided for both of the original datasets. We split the dataset into train, validate and test splits. Fitzpatrick 17k was randomly split so 80% of the samples were used for training and 20% for validation. All samples from the DDI dataset were used for testing. We also removed 10 images from the Fitzpatrick 17k samples before splitting the data which were used in the human study. These images had the attribute “seborrheic keratosis” or “malignant melanoma”. In total, we had 2574 training samples, 644 validation samples and 656 test samples. During training, samples were randomly rotated by up to 15 degrees, translated by up to 5% of the overall image width and scaled by up to 5%. All samples were resized to 512 by 512 pixels.

5.5.1.2 Model

Skincon models use a Densenet121 architecture (Huang et al., 2017) for the concept encoder which was initialised with pre-trained weights from ImageNet, and two linear layers with a ReLU activation function for the task predictor which

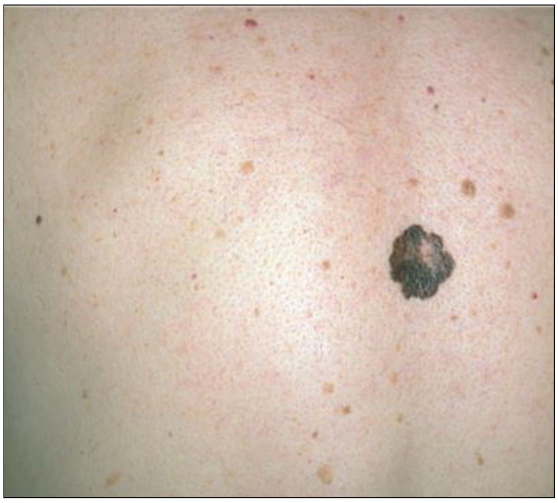
Input	Concepts
	Papule: not present Plaque: **present** Pustule: not present Bulla: not present Patch: not present Nodule: not present Ulcer: not present Crust: not present Erosion: not present Atrophy: not present Exudate: not present Telangiectasia: not present Scale: not present Scar: not present Friable: not present Dome-shaped: not present Brown: **present** White: not present Purple: not present Yellow: not present Black: not present Erythema: not present

Figure 5.4: Example sample from the Skincon dataset with concept annotations.

was not pre-trained. The concept encoder was trained to maximise the AUC of concept predictions, while the task predictor was trained to minimise task loss. To find optimal hyper-parameters for training we used Weights and Biases Sweeps (Biewald, 2020) configured with a Bayesian search method. The paramet-

Training method	LR	Optimizer	Batch size	LR patience	Epochs
Independent concept encoder	0.00053	SGD	32	10	100
Independent task predictor	0.0593	Adam	32	10	100

Table 5.3: Skincon training hyperparameters.

Training method	Concept accuracy (%)	Task accuracy (%)
Independent	91.235	88.474

Table 5.4: Skincon model accuracy. All values are rounded to 3 decimal places.

ers optimised in the sweeps were starting LR (between 0.1 and 0.0001), optimizer (between Adam (Kingma and Ba, 2014) and SGD), LR patience (between 3, 5, 10 and 15 epochs of no improvement in loss). The sweep ran until we stopped seeing improvements in the model accuracy, about 30 iterations. The final hyperparameters are summarised in Table 5.3 and the accuracy of the model we used in the study in Table 5.4.

5.5.1.3 Human Study Design

For the expert study 12 participants were asked to label images from the Skincon dataset (Daneshjou et al., 2022b). We selected 10 images that originated from Fitzpatrick 17k (Groh et al., 2021a) as we found these were of higher quality than images that originated from DDI (Daneshjou et al., 2022a). We excluded images that were out of focus and limited images to those with the label "malignant melanoma" and "seborrhoeic keratosis" as a dermatologist would typically look to diagnose a patient. The task labels "malignant" and "benign" in the dataset are sometimes used as a benchmark for machine accuracy (Daneshjou et al., 2022b; Groh et al., 2021b), but these are unnatural for a medical expert to use. For instance, it would be very odd to go to a doctor's appointment and only be

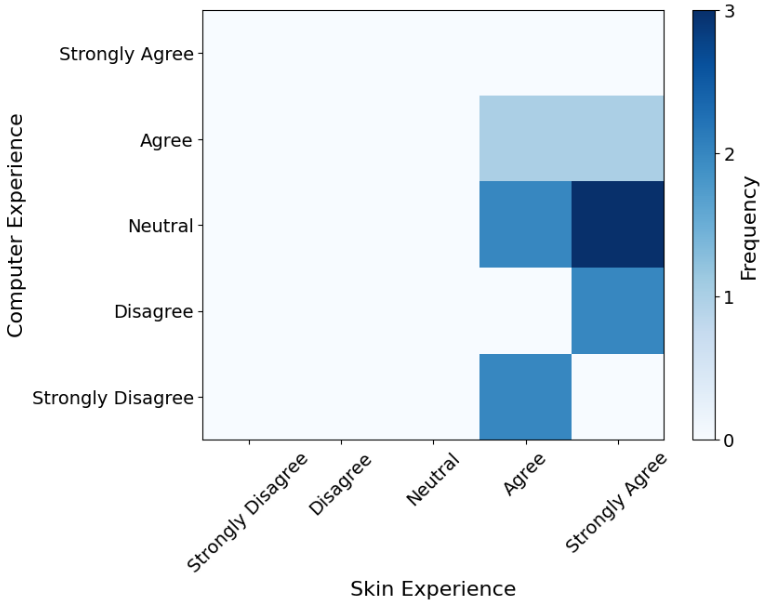


Figure 5.5: Expert study participant demographic.

told if your condition was okay or bad. We trained our model on "malignant" and "benign", but these were labelled as "malignant melanoma" and "seborrhoeic keratosis" for the study as by selecting the 10 images we used from subgroups of the dataset we can guarantee all "malignant" represent "malignant melanoma", and all "benign" samples represent "seborrhoeic keratosis".

All participants were either doctors, consultants or trainees with expertise in dermatology. Participants were located across Wales as we recruited participants via our contact in the Welsh National Health Service. The breakdown of demographics in Figure 5.5 shows that, unsurprisingly all participants have either said they agree or strongly agree they can diagnose skin diseases from images. Computer experience has a greater variance of experience but most participants either say they are neutral regarding having a good experience with computing or disagreeing with the statement.

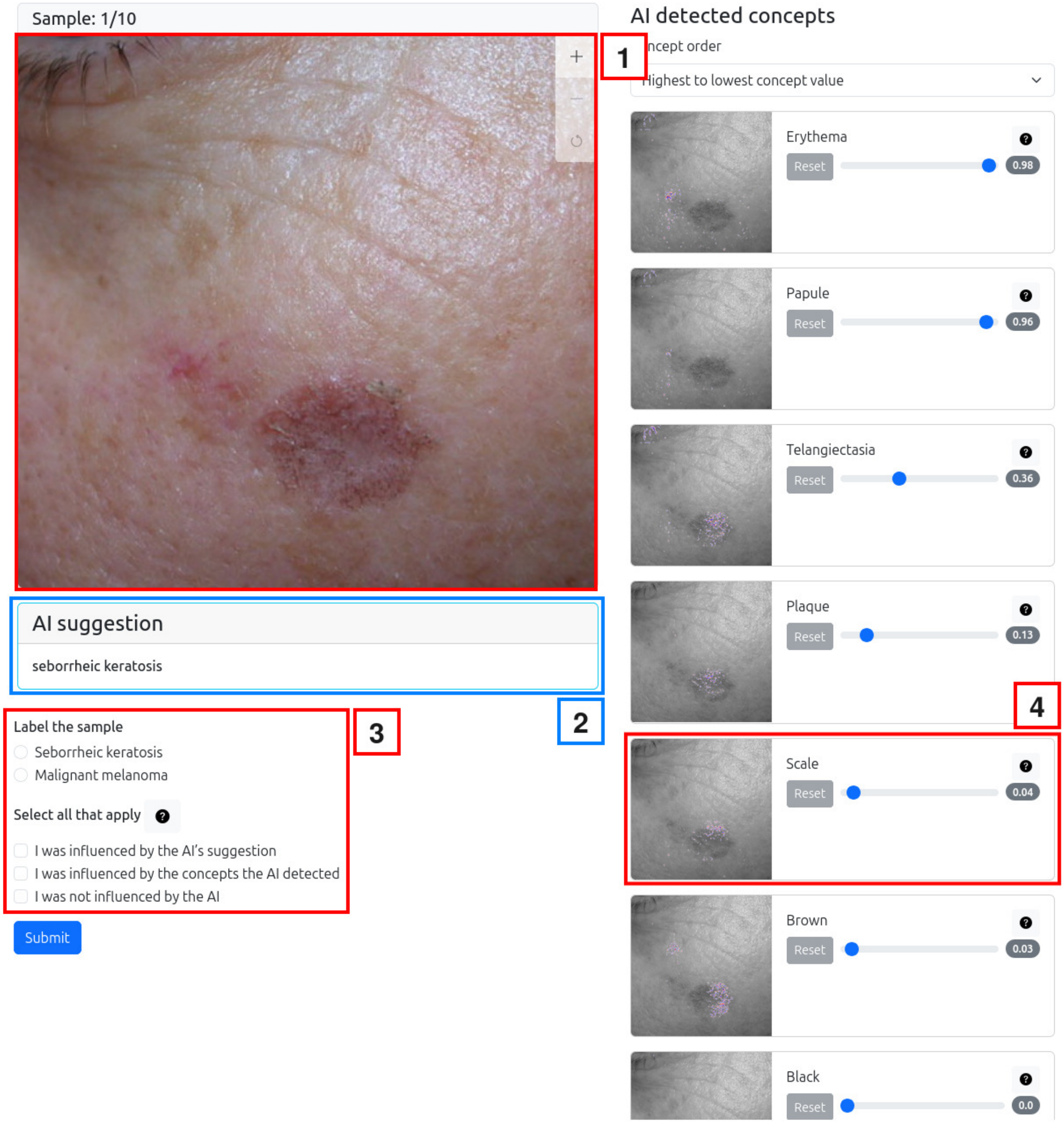
Task: 10 images were shown to participants in a random order. Participants were asked to diagnose the skin condition in each image in addition to selecting how they used the AI assistant with the following options:

- I was influenced by the AI's suggestion
- I was influenced by the concepts the AI detected
- I was not influenced by the AI

The AI use options were designed to capture whether participants selected labels based on the model's outputs or disregarded them. Alongside the images, participants were provided with the model's outputs, which included a predicted task label, predicted concepts, and an intervention slider for each concept. Adjusting any intervention slider automatically updated the model's predicted task label to reflect the changes. An example of the interface is shown in Figure 5.6.

At the end of the study, all participants completed a closing survey. The first part of the closing survey asked participants to complete SCS (Holzinger et al., 2020) questions. These questions are used as a measurement of quality for AI explanations, and use a similar question design to System Usability Scale (SUS) (Brooke, 1995). These questions use a Likert scale. The second part of the survey asked participants to type any other feedback regarding the AI and explanations. We specifically prompt them that this could include how accurate they perceived the AI agent was, and how useful the concept explanations and interventions were.

Variations: As previously detailed, we split participants into two groups. One with the default CBM model outputs and access to interventions (CExp+Int), while the second group also had access to saliency maps (Concept Explanation with Interventions and Saliency maps (CExp+Int+SMap)). These saliency maps were created using Guided Grad-CAM (Selvaraju et al., 2017) to highlight pixels contributed for and against each concept prediction. The two concept explanations outputs are shown in Figure 5.7.



Sample: 1/10

AI suggestion

seborrheic keratosis

Label the sample

○ Seborrheic keratosis
○ Malignant melanoma

Select all that apply

□ I was influenced by the AI's suggestion
□ I was influenced by the concepts the AI detected
□ I was not influenced by the AI

AI detected concepts

ncept order: Highest to lowest concept value

Concept	Value
Erythema	0.98
Papule	0.96
Telangiectasia	0.36
Plaque	0.13
Scale	0.04
Brown	0.03
Black	0.0

Key

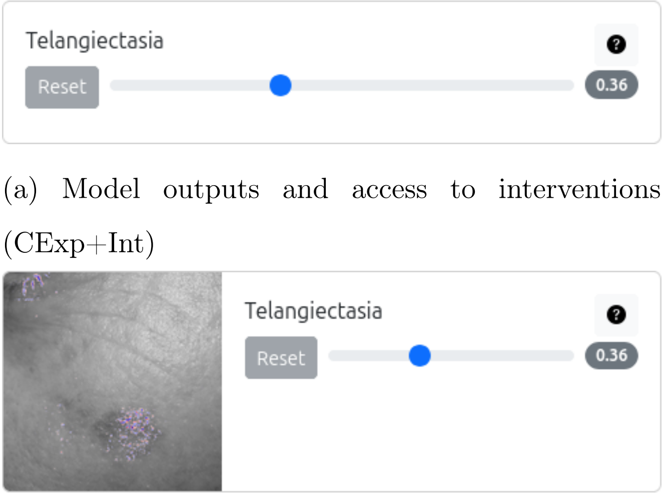
1: Sample image

2: AI agent output class label

3: Sample labels and AI use options for the participant to select

4: Concept outputs, salience map and intervention slider

Figure 5.6: Expert study platform interface with key components labelled.



(a) Model outputs and access to interventions (CExp+Int)

(b) Model outputs, access to interventions, and saliency maps (CExp+Int+SMap)

Figure 5.7: Expert study concept output variations.

5.5.2 Lay-person Study

The lay-person study was created to test CMs with a greater number of participants allowing us to include additional variables for model correctness and completeness. The task, playing games of Blackjack, required no specialist expertise, enabling us to recruit participants without the domain-specific skills that were necessary in the expert study.

5.5.2.1 Dataset

Blackjack is a dataset we introduced similar to the Playing cards dataset introduced in Chapter 3. Concepts in Blackjack represent the sum of card values in the player's card hand, whether the player has an “Ace” card which can have the value 11, the dealer's first card, and if the dealer has multiple cards. The task labels in Blackjack represent the best move available to the player according to the single deck strategy guide for the Blackjack card game (Shackleford, 2023). These labels are *hit* (the player gets another card), *stand* (the player ends the game with the cards they currently have), *surrender* (the player forfeits their hand but also half the loss they would get if they lost to the dealer), and *bust* (the player's cards sums to more than 21). We balanced the task labels which means the occurrences of concept labels are not equal. As observed with Poker cards, this should have minimal effect on the concept representations a CBM learns. We order samples such that they represent full games in Blackjack when placed in sequential order. As such we are able to extract samples to simulate full games during our study.

We created two variations of Blackjack: *standard Blackjack*, and *mixed Blackjack*. Standard Blackjack has cards drawn from a single deck of playing cards, whereas mixed Blackjack draws most cards from a single deck of playing cards, apart from all "Ace" and "Seven" cards which are drawn from a deck of cards with a different appearance. This allowed us to artificially reduce the accuracy of a model trained on the mixed Blackjack version when tested on standard Blackjack. Each dataset variation has 10,000 samples which are split into training samples and test samples with a 70%-30% split respectively. Both variations have instance-level concepts. Example samples can be seen in Figure 5.8.

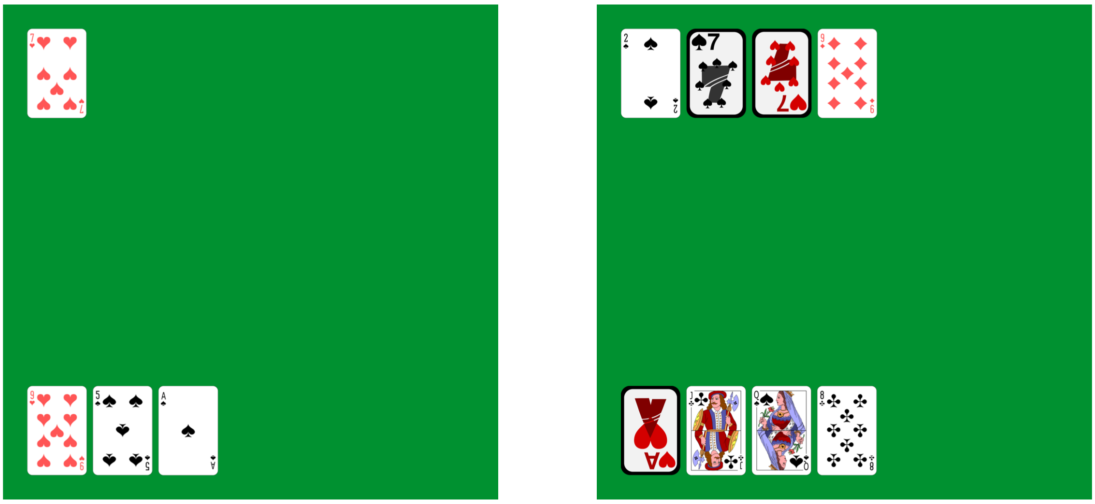
We transformed training samples with a random flip (both horizontal and vertical), applied a colour jitter to the brightness, contrast, saturation and hue, and randomly converted samples to grey scale. Samples are scaled to 299 by 299 pixels. The dataset is publicly available[2] along with the code to generate the dataset[3].

5.5.2.2 Models

Blackjack models used a VGG-11 architecture with batch normalisation (Simonyan and Zisserman, 2015) for the concept encoder and two linear layers with

[2] Blackjack dataset: `https://huggingface.co/datasets/JackFurby/blackjack`

[3] Blackjack dataset generator: `https://github.com/JackFurby/blackjack-dataset-generator`



(a) Standard Blackjack

(b) Mixed Blackjack

Figure 5.8: Example samples from the Blackjack dataset.

a ReLU activation function for the task predictor. Blackjack models were trained to minimise the concept and task loss. Due to the similarities between the Blackjack and Playing cards datasets, we reused the training parameters from Poker cards, which we have summarised in Table 5.5. Model accuracies are shown in Table 5.6. We trained two models: one on Standard Blackjack to create a model with high concept and task accuracy, and one on Mixed Blackjack with low concept accuracy when "Ace" and "Seven" cards are present. These models are the Acc model and Inacc model respectively.

Training method	LR	Optimizer	Batch size	LR patience	Epochs
Independent concept encoder	0.02	SGD	32	15	200
Independent task predictor	0.01	Adam	32	5	200

Table 5.5: Blackjack training hyperparameters.

Training method	Training dataset	Concept accuracy (%)	Task accuracy (%)
Independent	Standard Blackjack	99.818	98.874
Independent	Mixed Blackjack	96.434	81.306

Table 5.6: Blackjack models accuracies. All models were tested on Standard Blackjack and values are rounded to 3 decimal places.

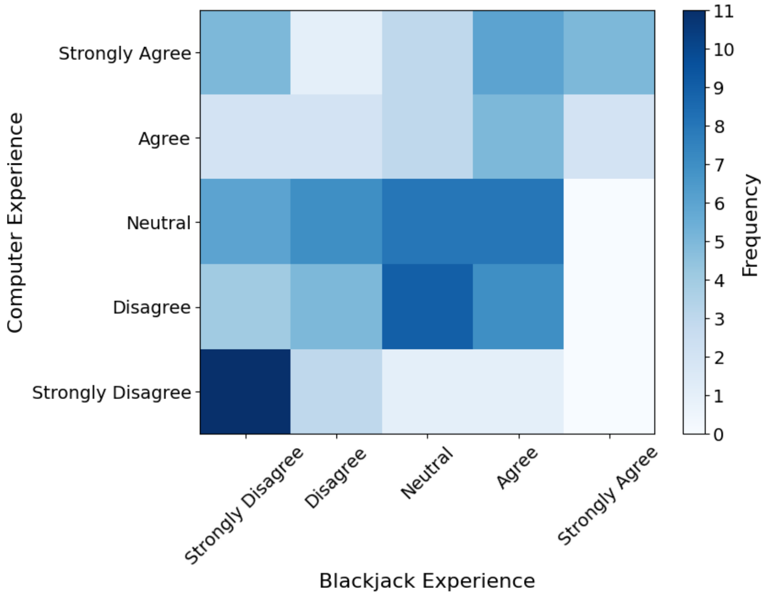
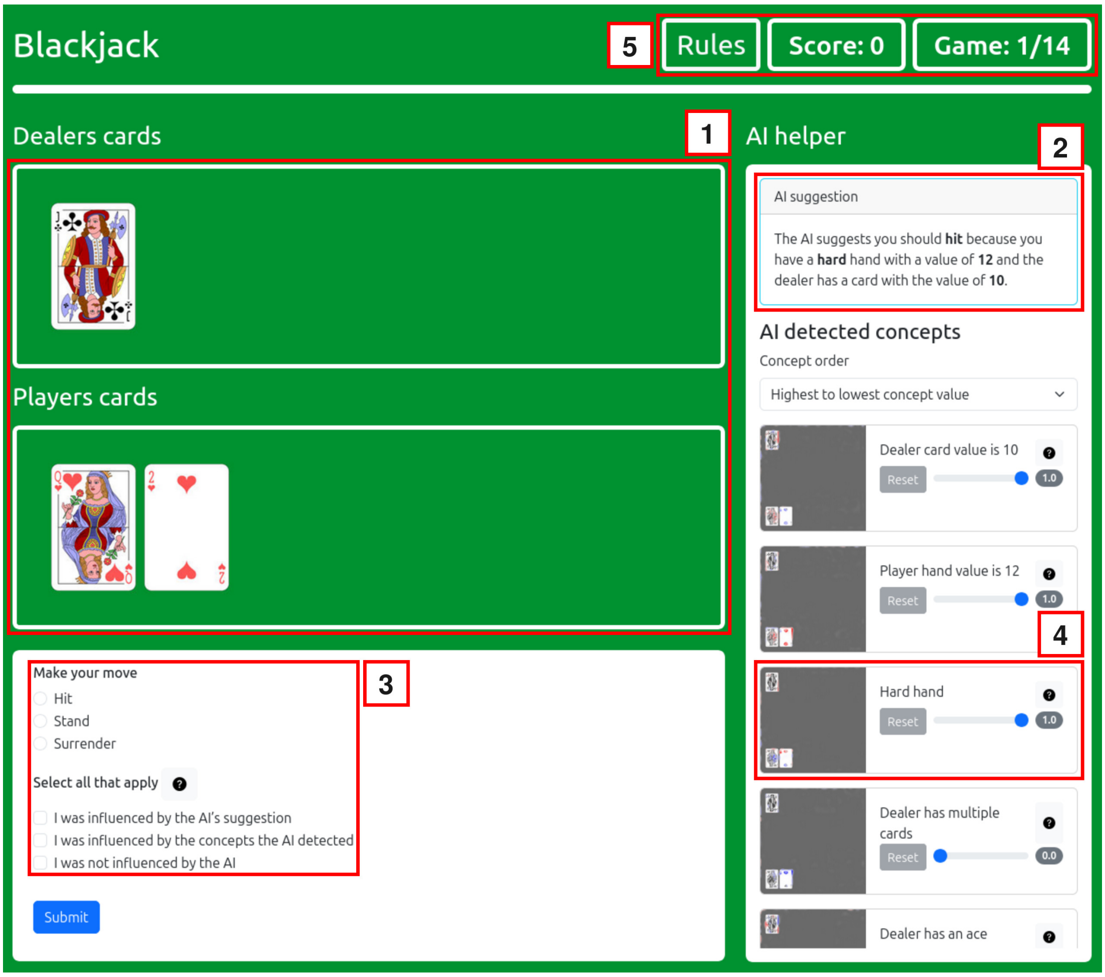


Figure 5.9: Lay-person study participant demographic.

5.5.2.3 Human Study Design

We recruited 104 participants, primarily by posting on university forums, noticeboards, mailing lists, and social media. Participants had to be 18 years old or older as detailed in the studies ethics approval. Although most of our participants were recruited from university forums, we have a mixture of students, members of staff and people without association with the university. We recorded participants' computing ability and blackjack ability using a Likert scale. A full breakdown of demographics is shown in Figure 5.9.



The AI suggests you should **hit** because you have a **hard** hand with a value of **12** and the dealer has a card with the value of **10**.

Make your move

○ Hit
○ Stand
○ Surrender

Select all that apply

□ I was influenced by the AI's suggestion
□ I was influenced by the concepts the AI detected
□ I was not influenced by the AI

Key

1: Concept outputs, salience map and intervention slider

2: Game rules, participant score and game counter

3: Sample image

4: AI agent output class and concept labels

5: Move selection and AI use options for the participant to select

Figure 5.10: Lay-person study platform interface with key components labelled.

Task: Each participant played 15 games of Blackjack, 1 of which was without the model enabled while the other 14 allowed participants to interact with the model.

Like with the expert study, the AI assistant suggested actions for participants to take, in this case, what moves the participant should play.

During the study, games of Blackjack were randomly selected from 100 games that were pre-generated. Each game had between 1 and 7 moves. Each game had cards drawn from a single standard deck of cards, and cards were replaced in the deck at the start of each game. We made some minor modifications to the game including the removal of betting, although participants still had a score which increased or decreased based on the number of wins and losses. Participants could select one of three moves during each game: (1) hit which gives them another card, (2) stand which ends the game with their current cards, and (3) surrender which ends the current game and loses half the number of points a lost game would lose. In addition to labelling samples, participants also selected how they used the AI assistant with the following options:

- I was influenced by the AI's suggestion
- I was influenced by the concepts the AI detected
- I was not influenced by the AI

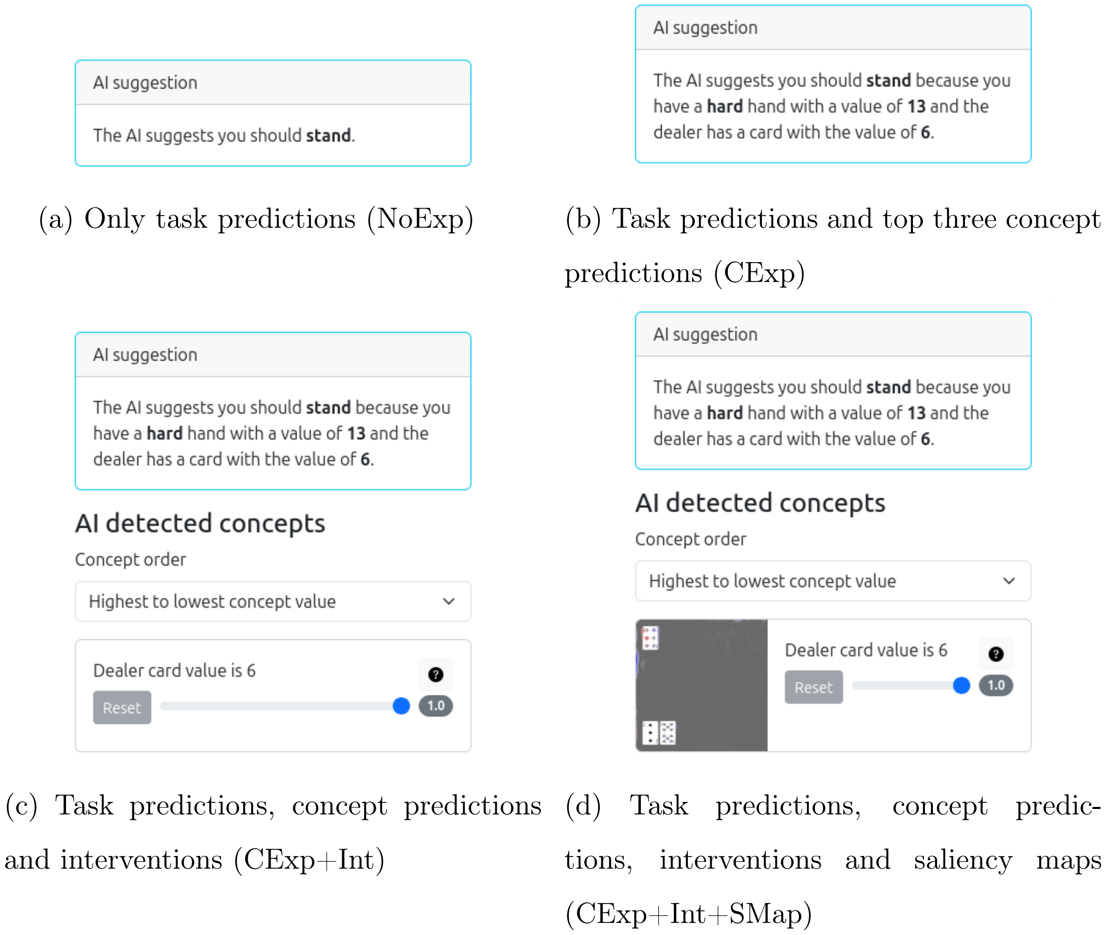
As with the expert study, the AI use options allowed us to capture how participants used the model. An example of the study interface is shown in Figure 5.10.

Finally, all participants completed a closing survey. The first part of the survey included the SUS (Brooke, 1995) questions followed by the SCS (Holzinger et al., 2020) questions. These questions used a Likert scale. As with the expert study, we also include a text box for participants to add any other comments about the model and how they interacted with it.

Variations: All participants had full access to their own card hands and the dealer's first card. Participants were split into eight groups with four versions of the model output and capabilities. An example of the four interface variations that are shown to participants is shown in Figure 5.11. The eight participant groups are:

- Acc-No Explanations (NoExp): Accurate model with only predicted task label
- Acc-CExp: Accurate model with predicted task label and three concepts
- Acc-CExp+Int: Accurate model with predicted task label, all concepts and the ability to perform interventions
- Acc-CExp+Int+SMap: Accurate model with predicted task label, all concepts, the ability to perform interventions, and each concept saliency map
- Inacc-NoExp: Inaccurate model with only predicted task label
- Inacc-CExp: Inaccurate model with predicted task label and three concepts
- Inacc-CExp+Int: Inaccurate model with predicted task label, all concepts and the ability to perform interventions
- Inacc-CExp+Int+SMap: Inaccurate model with predicted task label, all concepts, the ability to perform interventions, and each concept saliency map

As the study only used samples from standard blackjack, the inaccurate model would often predict hands with the cards "Ace" or "seven" incorrectly. Participants were evenly split between the accurate and inaccurate models as one of the motivations for this study, in addition to Sub-question 1, was to see what class labels would be predicted after intervening on concept predictions (Koh et al., 2020). If the model was always accurate then there would be less motivation to perform interventions.



AI suggestion

The AI suggests you should **stand**.

(a) Only task predictions (NoExp)

AI suggestion

The AI suggests you should **stand** because you have a **hard** hand with a value of **13** and the dealer has a card with the value of **6**.

(b) Task predictions and top three concept predictions (CExp)

AI suggestion

The AI suggests you should **stand** because you have a **hard** hand with a value of **13** and the dealer has a card with the value of **6**.

AI detected concepts

Concept order: Highest to lowest concept value

Dealer card value is 6 — Reset — 1.0

(c) Task predictions, concept predictions and interventions (CExp+Int)

AI suggestion

The AI suggests you should **stand** because you have a **hard** hand with a value of **13** and the dealer has a card with the value of **6**.

AI detected concepts

Concept order: Highest to lowest concept value

Dealer card value is 6 — Reset — 1.0

(d) Task predictions, concept predictions, interventions and saliency maps (CExp+Int+SMap)

Figure 5.11: Lay-person study model output variations.

5.5.3 Evaluation Methodology

We evaluated our human studies with a mixture of objective and subjective metrics to analyse interventions, trust, interpretability, and human-machine performance. Starting with interventions we first need to understand how and when these are made. We have classified interventions into two categories: *error correction* and *feature adjustment*. Error correction interventions are made to concepts that participants see as incorrect and thus we define this intervention type as concepts that are intervened a maximum of once per sample where the intervened concept value $\bar{c}$ is in the range $0 \leq \bar{c} \leq 0.1$ or $0.9 \leq \bar{c} \leq 1$. The thresholds 0.1 and 0.9 were selected as they are close to the maximum and minimum limits to reflect decisiveness, but accommodate for minor inaccurate interactions (e.g. clicking on the wrong part of the intervention slider). Feature adjustments are all other interventions which include concepts that are intervened more than once in a given sample, or where the intervened concept value is in the range $0.1 < \bar{c} < 0.9$. Feature adjustment interventions are when the participant is not certain the model has incorrectly predicted the presence of a concept, or where they are inspecting changes to task label predictions.

We have also assigned the following labels to understand how concept values change with interventions:

- Binary change: Measure whether a concept changes from present to not present, or from not present to present.
- Changed model task label: Measure whether an intervention changes the predicted task label.
- Magnitude: Measure how much each intervention changes the concept value by.
- Cumulative Change: Measure the total change from the model predicted concept value to the final intervened concept value.
- Reversal: Measure whether the final intervened concept value is close to its initial value when a concept has had multiple interventions for a given sample.

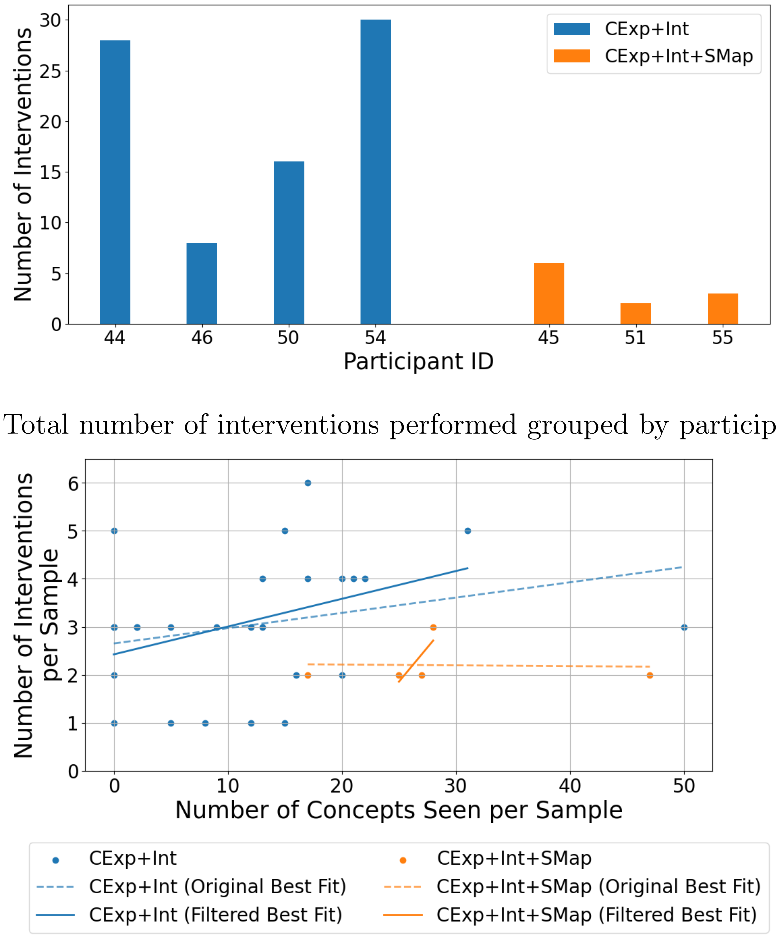
Our objective metrics primarily used intervention data and which concepts were seen by participants. We tracked interventions by participants, and interventions over time in order to evaluate if the rate of interventions increased or decreased over time. For trust, we evaluated participant and model concepts and task label alignment. If the model and human have the same concepts and task labels for a large proportion of samples we can argue the human participants trust the model which we can label as justified and unjustified based on if the model was correct or not (Wang and Yin, 2021; Lai and Tan, 2019). We also analysed the label the participants submit in comparison to ground truth labels from the dataset.

In the paper introducing CBMs (Koh et al., 2020) they made two statements about CBM capabilities. Firstly, the models are interpretable because the models predict high-level concepts which are in tern interveneable. This enables humans to obtain counterfactual explanations. Secondly, the authors evaluated interventions with a metric they called test-time intervention. This metric looked at the change in task error as concept values were updated and replaced with their ground truth values. We looked at both of these by first evaluating interventions as discussed previously in this section. In addition, we repeated test-time intervention with participant data which allowed us to evaluate the intervention's ability to improve task and concept accuracy in a real-world setting.

5.6 Results

5.6.1 Expert Study

Out of our 12 participants, 120 samples were labelled with 2,051 concepts recorded as seen and 93 interventions were made. 7 of the participants performed interventions with the other 5 not using the model intervention capability. The count of interventions by participants can be seen in Figure 5.12a where there is a clear difference in the number of interventions between CExp and CExp+Int+SMap participants. Out of the participants who performed interventions, CExp+Int participants performed an average of 22.5 interventions across 10 samples with a standard deviation of 10.38, while CExp+Int+SMap participants performed an average of 3.67 interventions across 10 samples with a standard deviation of 2.08. Unsurprisingly, the number of interventions increased as more concepts were seen, as shown in Figure 5.12b.



(a) Total number of interventions performed grouped by participant.

(b) Number of concepts seen compared to the number of interventions per sample.

Figure 5.12: Most interventions are preformed by participants without access to saliency maps, and the number of interventions performed increase with the number of concepts seen.

A summary of interventions is displayed in Table 5.7. CExp+Int Participants performed 58.5% interventions to correct what they believed were errors in the predicted concepts with the other 41.5% of interventions made to explore the concept space. Out of the feature adjustment interventions, 17.6% of interventions were reversed with 64.5% of feature adjustment interventions kept. Out of samples with at least one intervention, 2.78 concepts were intervened per sample. Half of all interventions change a concept prediction from present to not present, or the other way around. Interventions changed a concept value on average by around

Data subset	Total interventions	Error correction	Feature adjustment	Interventions per sample	Concept intervened per sample	Binary	Changed model task label	Reversal	Mean intervention magnitude	Mean cumulative magnitude
All	93	48	45	2.91	2.50	44	14	11	0.48	0.5
CExp+Int	82	48	34	3.04	2.78	41	12	6	0.49	0.52
CExp+Int+SMap	11	0	11	2.20	1	3	2	5	0.39	0.11

Table 5.7: Breakdown of interventions performed in the expert study.

0.5. 14.6% of interventions changed the model's task prediction which could suggest the model is not very sensitive to the concepts participants intervened on.

CExp+Int+SMap participants made far fewer interventions with all interventions classified as feature adjustments. Almost all interventions were made to a single concept per sample with that intervention reversed back to the model-predicted concept value. Compared to CExp+Int participants, all interventions appear to be exploring the model sensitivity to changes in concept values, but one concept at a time. As the only difference between the two groups was the addition of saliency maps, the change in how interventions are performed suggests that saliency maps helped participants analyse concepts more efficiently, leading to fewer interventions that were focused on exploring model concept sensitivity.

5.6.1.1 Human-machine Task Alignment

An objective metric of trust is the alignment between the humans and model task outputs (Wang and Yin, 2021; Lai and Tan, 2019). By measuring if participants

Data subset	Overall (%)	Initial model task prediction alignment (%)	Intermediate model task prediction alignment (%)	Final model task prediction alignment (%)
All	80.8 (±3.6)	77.5 (±3.8)	77.8 (±8.2)	65.6 (±8.5)
CExp+Int	**81.7 (±5.0)**	76.7 (±5.5)	81.8 (±8.4)	70.4 (±9.0)
CExp+Int+SMap	80.0 (±5.2)	78.3 (±5.4)	60.0 (±24.5)	40.0 (±24.5)
NoInt	**81.8 (±4.1)**	81.8 (±4.1)	-	-
WithInt	78.1 (±7.4)	65.6 (±8.5)	77.8 (±8.2)	65.6 (±8.5)
CExp+Int-NoInt	**81.8 (±6.8)**	81.8 (±6.8)	-	-
CExp+Int-WithInt	81.5 (±7.6)	70.4 (±9.0)	81.8 (±8.4)	70.4 (±9.0)
CExp+Int+SMap-NoInt	**81.8 (±5.2)**	81.8 (±5.2)	-	-
CExp+Int+SMap-WithInt	60.0 (±24.5)	40.0 (±24.5)	60.0 (±24.5)	40.0 (±24.5)

Table 5.8: Expert study human-machine task alignment.

are labelling samples the same as the CBM we can argue if they trust the model or not. To further reinforce if alignment is helping the human-machine team, we can also compare participants accuracy where higher alignment and accuracy can highlight trust being justified.

Alignment is calculated as the average number of samples where the participant's selected label matches the model's predicted label, or labels if a participant used interventions. Alignment is separated into four groups: (1) Overall alignment which reflects alignment with any model prediction, i.e. participant selected labels matches at least one model predicted label for a given sample. (2) Initial alignment which considers only the model's task label prediction before intervention, (3) final alignment which considers the model's task label predictions after all interventions have occurred, and (4) intermediate alignment which captures alignment with any intermediate label when a sample has been intervened upon two or more times. Alignment results are shown in Table 5.8.

Overall human-machine alignment ranges from 60% to 81.8%. We see a drop from 81.7% to 80% for CExp+Int+SMap participants and CExp+Int participants respectively, and 81.8% to 78.1% for participants who did not perform interventions compared to those who did. Splitting with and without interventions and the explanation versions shows that saliency maps and interventions have a small drop in mean alignment, while saliency maps and interventions have a far larger fall in interventions (but also a far larger standard error).

Human-machine alignment does not increase from the initial model label to the final model label. Where we see an increase in human-machine alignment is between the initial model label and intermediate labels. As the standard error often overlaps the data subsets these results are not conclusive. Therefore, we will extend our discussion with the lay-person study.

These results show interventions reduce the frequency of instances where participants agree with the CBM's labels. However, interventions clearly influence participants' labelling decisions, as alignment with the initial model labels is consistently lower than the overall alignment. In fact, the initial model alignment with interventions (65.6%) is close to the true accuracy of the model without interventions (70%). These findings indicate that interventions help participants calibrate their trust in the model towards the appropriate amount to align with the model's true accuracy. Without interventions, participants appear to be over-trusting the model. A one-tailed t-test reveals statistical significance with a p-value of 0.03, below the 0.05 threshold, confirming that the absence of interventions led to increased alignment in this study. However, the difference in alignment between CExp+Int and CExp+Int+SMap participants was not statistically significant (p-value of 0.59), suggesting that the addition of saliency maps had no meaningful impact on alignment than providing concept explanations alone. However, a larger study may confirm otherwise.

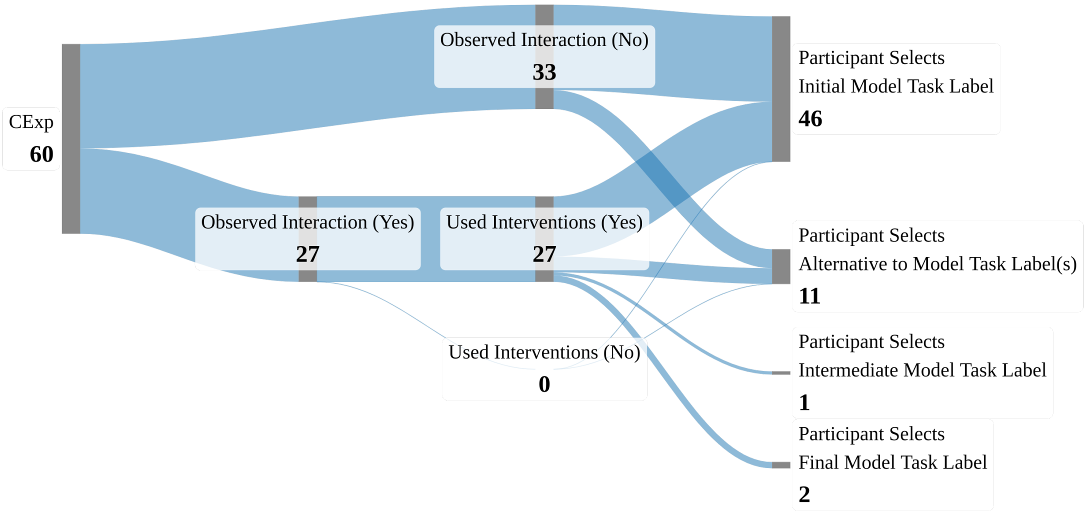


Figure 5.13: CExp+Int participants mostly selected the task label aligned to the initial model task label. If interventions were preformed on a sample then the proportion of samples where the participant label aligned with the initial model task prediction falls slightly.

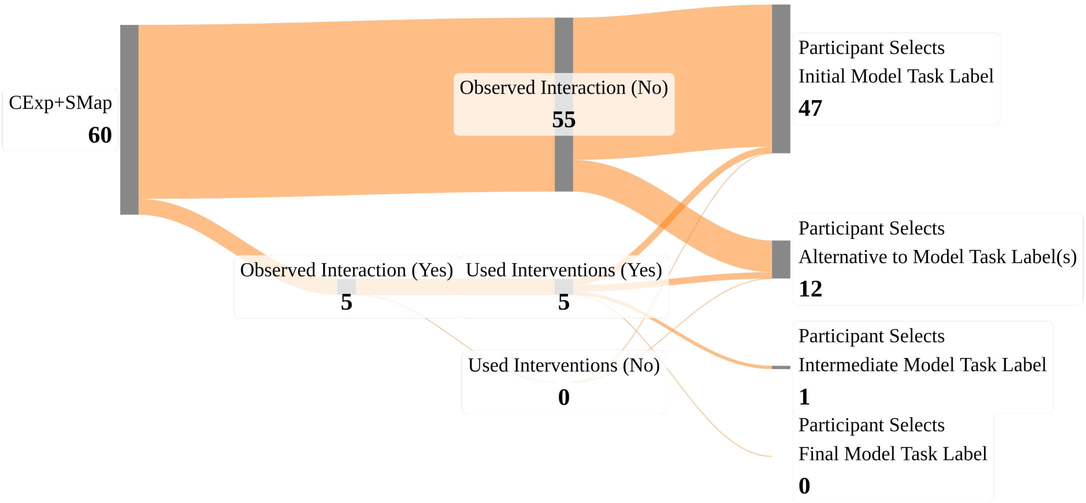


Figure 5.14: CExp+Int+SMap participants mostly sided with the model initial task label prediction. A few samples were labelled with a task label the model did not predict, and no samples were labelled with the final task label predicted.

Data Subset	Overall Accuracy (%)	Malignant Melanoma (%)	Seborrheic Keratosis (%)
All	78.3 (±2.4)	88.3 (±4.6)	68.3 (±3.9)
CExp+Int	75.0 (±3.4)	83.3 (±8.0)	66.7 (±4.2)
CExp+Int+SMap	**81.7 (±3.1)**	93.3 (±4.2)	70.0 (±6.8)
NoInt	78.0 (±3.7)	88.0 (±8.0)	68.0 (±8.0)
WithInt	**78.6 (±3.4)**	88.6 (±5.9)	68.6 (±4.0)
CExp+Int-NoInt	75.0 (±5.0)	80.0 (±20.0)	70.0 (±10.0)
CExp+Int-WithInt	75.0 (±5.0)	85.0 (±9.6)	65.0 (±5.0)
CExp+Int+SMap-NoInt	80.0 (±5.8)	93.3 (±6.7)	66.7 (±13.3)
CExp+Int+SMap-WithInt	**83.3 (±3.3)**	93.3 (±6.7)	73.3 (±6.7)

Table 5.9: Expert study human task

accuracy.

We can visualise how each sample is labelled by following to the flow chart in Figure 5.1. Interaction with the model is labelled as "observed interaction", This includes samples where participants either selected their task label as being influenced by the model or where they performed at least one intervention on that sample. CExp+Int participants (Figure 5.13) nearly equally split samples between interacting with the model and not interacting with it. For most samples, participants' final labels aligned with the initial model predictions. However, when participants performed interventions, they were more likely to select an alternative label or a label predicted by the model in a future iteration. This pattern continued for participants with access to saliency maps (Figure 5.14). This continues to suggest interventions influence a humans decision-making process.

If appropriate trust is given to the model, we should expect human accuracy to

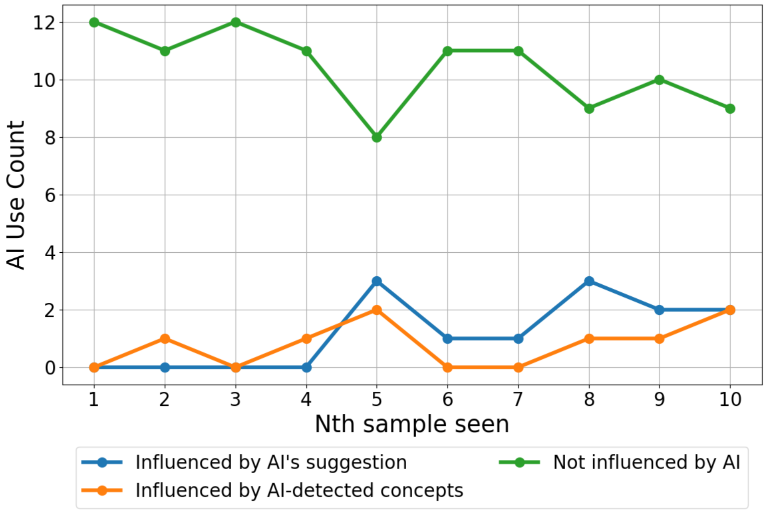


Figure 5.15: AI use shows participants start off not using the model with usage increasing a little over time.

be higher than that of the model, as we may assume that appropriate trust means a participant aligns with the model when it is accurate, and selects a different label when it is inaccurate. We show human accuracy in Table 5.9. Accuracy is averaged by participant, assuming that participants build a mental model of the model over time. Even when participants did not explicitly use the model's prediction for an individual sample, they may still have been influenced by it in their decision-making process. For instance, they may recognise when the model is incorrect without the need to perform interventions.

Overall, CExp+Int participants achieved an accuracy of 75% with CExp+Int+SMap participants achieving an accuracy of 81.7%, indicating that the additional information provided by saliency maps aids decision-making. Evaluating the impact of interventions on task accuracy, participants who did not use interventions achieved an accuracy of 78%, while those who did use interventions achieved a slightly higher accuracy of 78.6%. This suggests that interventions either match or slightly enhance participant performance, contributing to improved participant accuracy.

When comparing CExp+Int+SMap and CExp+Int participants, the former consistently outperformed the latter. The accuracy improvement from saliency maps was greater than the difference between participants who used interventions and those who did not. While we do not have the sample size to show statistical significance (A one-tailed t-test resulted in a p-value of 0.09 for accuracy being higher if participants had saliency maps, and a p-value of 0.46 if participants performed interventions), the observed trends suggest that both saliency maps and interventions provide a distinct advantage for participants completing the study. Additionally, saliency maps and interventions appear to complement each other. Participants with access to both achieved the highest accuracy of any group.

Human-machine alignment with participant stated AI use (whether participants selected they were influenced by the model suggestions, detected concepts, or not influenced by the model) in Figure 5.15 shows that participants initially stated the model suggestions or detected concepts did not influence them. The stated AI use then increased slightly over time as the study progressed. Although we may expect AI use to be high, we are reminded participants are knowledgeable in skin disease identification so the models outputs may not be needed for all samples.

5.6.1.2 Interventions Over Time

If a participant is interacting with a model we may expect the number of interventions to decrease over time. Essentially, when they first start using the CBM they will have no understanding about how sensitive the model is to concept values. Over time we'd expect participants to start to build some understanding about the models and thus their interaction may decrease as they will be exploring concept sensitivity less. In Figure 5.16 we show the average number of interventions per sample with the standard error. For both participant groups the number of interventions starts high before falling. For CExp+Int+SMap participants, this drops to 0, while for CExp+Int participants, they continue to intervene with concept values throughout the 10 samples they see.

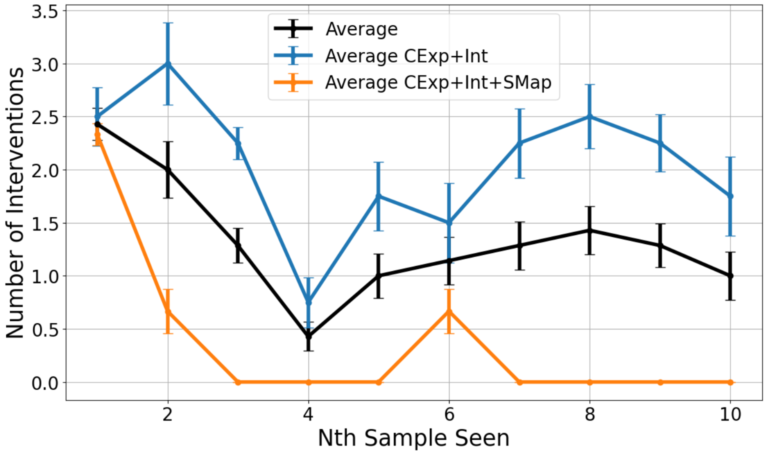


Figure 5.16: Participants in the expert study started by performing 2.5 - 3 interventions per sample. The number of interventions decreases over the next four samples before rising again for participants without saliency maps. Participants with saliency maps, for the most part, do not perform any more interventions.

These results suggest that participants initially have a high engagement with the model, but their engagement declines over time as they build a mental model of its behaviour. The difference between CExp+Int+SMap and CExp+Int participants indicates that saliency map explanations provide additional insights into the model's concept predictions. This additional information may explain the rapid decline in interventions among these participants which does not recover.

In contrast, CExp+Int participants experience a sharp decline in interventions, followed by a gradual increase for later samples. Without access to saliency maps explanations about the model's concept predictions, these participants appear to be incentivised to use interventions as a means of understanding the model's decision-making process.

5.6.1.3 Test-time Intervention and Concept Accuracy

In (Koh et al., 2020) the authors evaluate if interventions can improve task accuracy with a metric they call test-time intervention. Although in their evaluation CBMs show improvements in task performance, they only used ground truth concept values for interventions. Therefore, it remains unknown how interventions improve task performance when interventions are made by humans, and thus interventions may be made to concepts that the model is not sensitive to, or may not result in a concept value being set to 0 or 1.

In the motivation section, we discussed (Barker et al., 2023) where they found CBMs may not apply the same weight to concepts for task labels as humans would. Therefore we hypothesise interventions performed by humans may not see the same improvements in task performance in comparison to the automated metric.

By repeating the test-time intervention metric with participants from our studies we can show if human performed interventions improve the task accuracy of CBMs, and thus confirms if the automated metric results from (Koh et al., 2020) also holds with humans. Alternatively, if our results show no improvement, or a decrease to the model task accuracy we can show there is a misalignment between humans and the model's sensitivity to concepts.

In Figure 5.17 we show test-time intervention results comparing the average task accuracy of our participant groups with interventions to the task accuracy without interventions. In Koh et al. (2020), their test-time intervention results showed task accuracy increased with interventions. So we do not include samples that were not intervened on we only included the same samples between with interventions and without. For example, if participants intervened on concepts for samples 1 and 3 but not 2, we only work out the task accuracy for samples 1 and 3. As participants performed 2 - 3 interventions on average when they did not have access to saliency maps, or 2 with saliency maps, we cannot draw any firm conclusions

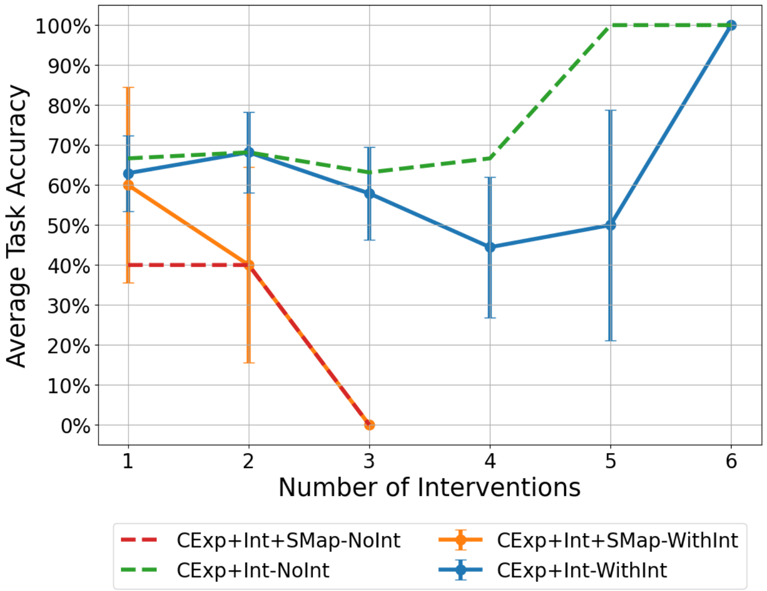
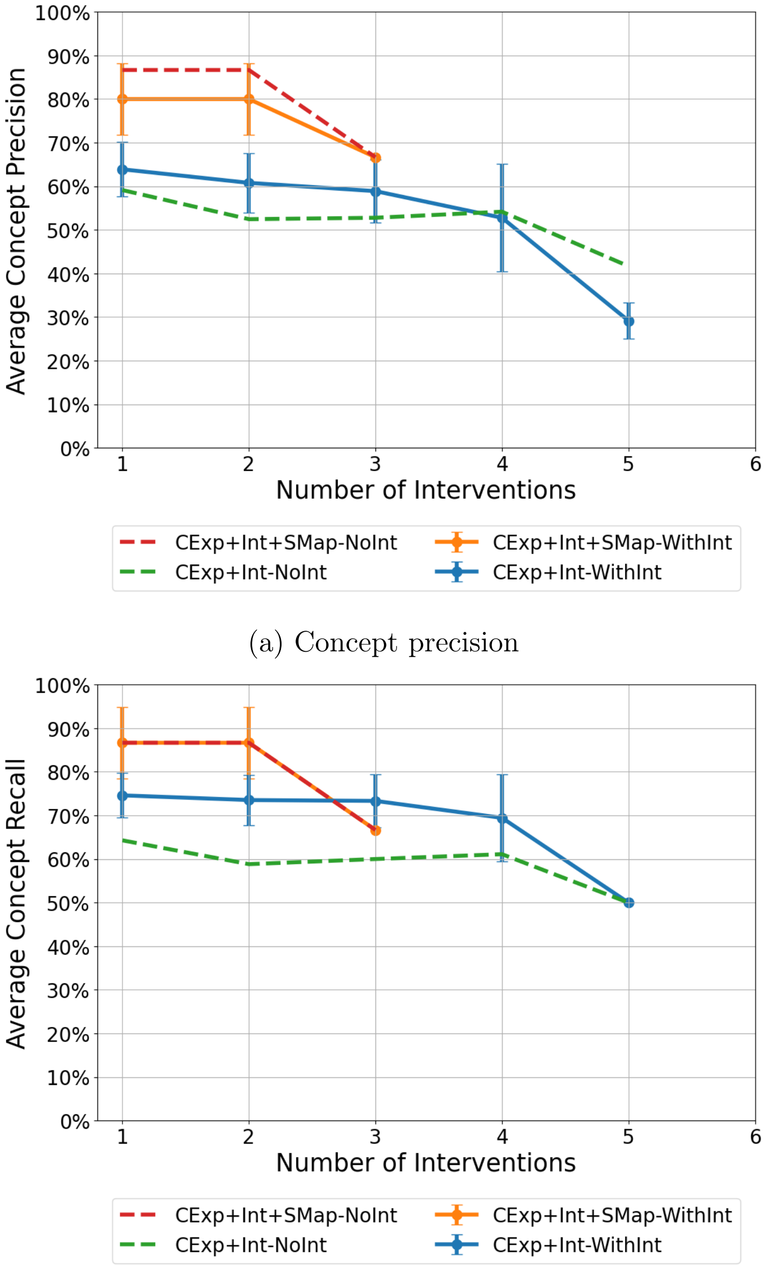


Figure 5.17: Test-time interventions for the expert study on average does not improve the accuracy of the CBM without interventions. The error bars shows the standard error for per participant accuracy. As model accuracy is group per model, these lines do not include error bars.

if interventions improve task accuracy past these intervention counts.

Task accuracy does not improve with interventions compared to no interventions. When 1 - 3 interventions are performed task accuracy is close to matching the accuracy of the CBM with no interventions, and outperforms the model with 1 intervention for CExp+Int+SMap participants. However, as the error bars show the range of task accuracy with interventions ranges from higher to lower than the model alone with no interventions, it is clear these results will require further evidence from the lay-person study. The final insight this result tells us is with more interventions the task accuracy starts to increase for CExp+Int participants, but with few participants, we cannot conclude if this is a significant result.

In addition to measuring changes in task accuracy, we can also measure the change in concept accuracy. Figure 5.18 illustrates the precision and recall for



(a) Concept precision

(b) Concept recall

Figure 5.18: Interventions match or exceed the concept precisions and recall of the model prediction without interventions.

concepts separated by the two participant groups. Notably, interventions lead to an increase in precision for CExp+Int participants. Despite the small sample size, the error bars consistently show that the mean precision with interventions is higher than without them. Interestingly, the same trend is not observed for CExp+Int+SMap participants: their interventions do not improve concept accuracy, but also did not lower the task accuracy as seen in Figure 5.17.

A similar pattern is observed with recall. For both participant groups, recall with interventions either matches or exceeds the recall of the CBM concept predictions alone.

When combining these findings with the test-time intervention results for task accuracy, it becomes clear that the CBM is not aligned to changes in concept values for the concepts participants are adjusting. Although interventions often make concept vectors more accurate, task accuracy does not reflect this improvement. Instead, task accuracy generally remains the same or slightly decreases. This aligns with the findings in (Barker et al., 2023), suggesting that the concepts that CBMs use for task label predictions are not aligned with the concepts humans perform interventions on. Assuming participants intervene on concepts that they expect to be critical for making a task prediction if the concepts used by the model and those targeted by human interventions were aligned, we would expect task accuracy to remain stable. However, we observe a slight reduction in overall accuracy, indicating a misalignment between the two.

5.6.1.4 System Causability Scale and Participant Comments

We used the SCS (Holzinger et al., 2020) to get a subjective rating of explanation suitability (all questions are detained in the Appendix, Table B2). We have provided several subsets of participant responses in Table 5.10. The overall score, computed as the average of participants' summed responses normalised by the maximum possible score, is between 0 and 1 where 0.68 indicates an average response (Holzinger et al., 2020). Individual questions use a Likert scale ranging from 1 for "strongly disagree" to 5 for "strongly agree". Almost all overall scores are either 0.68 or slightly below. The only exception to this is the subset of

Question	All	CExp+Int	CExp+Int+SMap	WithInt	NoInt	Skin Experience Agree	Skin Experience Strongly Agree
Factors in data	3.09	3.00	3.20	3.00	3.20	2.80	**3.33**
Understood	3.73	3.33	4.20	3.67	3.80	3.60	**3.83**
Change detail level	3.18	**3.67**	2.60	3.50	2.80	2.80	3.50
Need support	3.64	3.33	**4.00**	3.50	3.80	**4.00**	3.33
Understanding causality	3.00	3.17	2.80	**3.33**	2.60	2.80	3.17
Use with knowledge	3.45	3.50	3.40	3.50	3.40	3.00	**3.83**
No inconsistencies	3.00	3.17	2.80	3.00	3.00	2.60	**3.33**
Learn to understand	3.55	3.50	3.60	3.50	3.60	3.20	**3.83**
Needs references	3.73	3.67	3.80	3.50	4.00	3.60	**3.83**
Efficient	3.45	**3.67**	3.20	**3.67**	3.20	3.40	3.50
Overall score	0.68	0.68	0.67	0.68	0.67	0.64	**0.71**

Table 5.10: Likert Scores for SCS questions.

participants who selected "strongly agree" as their experience at classifying skin diseases this subset of participants' overall score is 0.71.

Looking into the score for individual questions, we can see most questions averaged to be between high 2 and high 3, which would map to the Likart options "disagree", "neutral", and "agree". Out of the questions, a few stand out. Starting with *change detail level* (*I could change the level of detail on demand*), this was rated higher for CExp+Int participants, those who performed interventions, and participants who self-rated their experience at skin disease identification as "strongly agree". This aligns with our observations of interventions as these were primarily performed by CExp+Int participants. It is clear that if participants perform interventions, they understand this impacts the detail a model provides. Out of participants who rated their experience as "strongly agree", 57% of them used interventions.

Need support (*I did not need support to understand the explanations*) was the highest rated question overall with all participant groups rating 3.33 or higher. The highest ratings for these questions were made by CExp+Int+SMap participants, showing the potential benefit saliency maps provide to help participants interpret the model's concept predictions. The second group of participants to highly rate these questions were participants who answered their skin disease identification experience to be "agree". Out of these participants, 60% (with two of these participants performing almost 60 interventions) used interventions, suggesting their increased interaction with the model improved their understanding of the model.

Finally, *efficient* (*I received the explanations in a timely and efficient manner*) was also consistently rated slightly over 3, which aligned with the "neutral" option. As discussed, with our test-time intervention results, this suggests participants recognised the concepts they intervened with were not aligned with the model decision-making process. Participants who performed interventions answered this question with a slightly higher score than those who did not perform interventions, but overall all participants did not perceive the model to improve explanation efficiently.

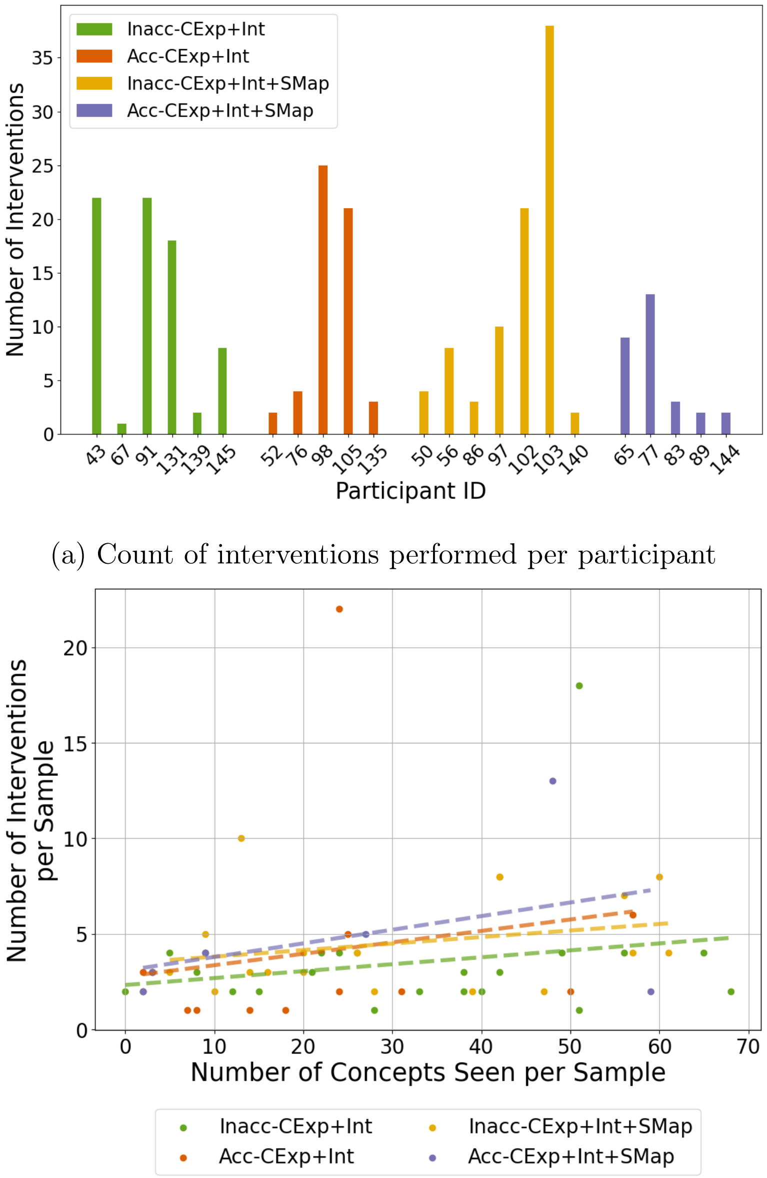
We also asked participants for any other comments, with the suggestion these could be about the CBM accuracy and explanation details. These highlighted participants' understanding of the model and study as a whole. Regarding explanations one participant said "I am not sure what you mean by 'explanations' hence marked neutral for most of them" which is telling of why the SCS scores could be around 0.68. Although concepts were described as explanations in the study, this may not be a typical term for people outside of AI and ML domains. We also had participants saying they "didn't use the concepts at all" while others noted the challenge of labelling images as there was no patient history to go along with the images. Finally, one participant said "AI may be helpful for either non-dermatologists or students" which indicates concepts are most useful when a human is not a domain expert or is otherwise unsure about how to label a sample.

Overall these comments highlight a mixed response between participants. 42% of participants did not comment on the model output or explanations, while 33% questioned the usefulness or meaning of concepts, and 17% were either positive or negative about the model. The negative comment about the model is that model was too confidant or inaccurate. Saliency maps or interventions were not discussed by any participants. As one-third of participants questioned the usefulness or meaning of concepts, more work should be made where both ML practitioners and domain experts (e.g. medical experts) work together to ensure new research aligns with the expectations of humans and target demographics. In the case of CMs, this should focus on the concepts a model uses, and how these concept are used for a model decision-making process.

5.6.2 Lay-person Study

For our lay-person study out of the 104 participants 1,560 games of Blackjack were played with 1,456 of those games including a model to suggest moves to play, while the other 104 games were played without a model output. 11,600 concepts were recorded as seen and 243 interventions were performed by 23 participants out of 52 who had the capability to do so. The count of interventions by participants can be seen in Figure 5.19a. 65.4% of interventions are performed on the inaccurate model, with the remaining 34.6% of interventions performed on the accurate model. This indicates participants are still engaging with the models even if they accurately predict most concepts, and thus it suggests participants are still motivated to use interventions to interpret the model decision-making process. As with the expert study, there is a small correlation between samples seen and interventions performed (Figure 5.19b).

A breakdown of interventions for the lay-person study is shown in Table 5.11.



(a) Count of interventions performed per participant

(b) Number of interventions performed compared to number of concepts seen

Figure 5.19: Most interventions (65.4%) are preformed by participants with the inaccurate model, and the number of interventions performed increase with the number of concepts seen.

Most participant groups perform both error correction and feature adjustment interventions. Participants with the accurate model primarily perform feature adjustment interventions which we expected as the model will rarely predict concepts incorrectly. Therefore these participants are exploring the model's sens-

Data subset	Total interventions	Error correction	Feature adjustment	Interventions per sample	Concept intervened per sample	Binary	Changed model task label	Reversal	Mean intervention magnitude	Mean cumulative magnitude
All	243	83	160	4.05	2.42	152	29	60	0.58	0.53
Acc-CExp+Int	55	11	44	3.93	2.71	38	8	7	0.59	0.57
Inacc-CExp+Int	73	47	26	3.48	2.57	41	8	20	0.56	0.52
Acc-CExp+Int+SMap	29	0	29	4.83	1.50	17	5	8	0.52	0.11
Inacc-CExp+Int+SMap	86	25	61	4.53	2.32	56	8	25	0.62	0.58

Table 5.11: Breakdown of interventions performed in the lay-person study.

itivity to concepts. Participants with the inaccurate model perform almost an equal number of error correction interventions (47.2%) and feature adjustment interventions (52.7%).

The intervention patterns show CExp+Int+SMap participants performed slightly more interventions on average (4.53–4.83) than CExp+Int participants (3.48–3.93). CExp+Int+SMap participants primarily intervened on 1–2 concepts per sample, while CExp+Int participants intervened on 2–3 concepts. Binary interventions were predominantly used with inaccurate models, likely due to the higher number of error correction interventions performed (47 compared to 11 for accurate models).

Interestingly, reversal interventions remained high for inaccurate models, which might suggest that participants needed to perform additional interventions to build a mental model of the model. Additionally, it could indicate a misalignment between the participant and model decision-making processes. E.g. if the participant did not agree with the resulting task label prediction from the model after an intervention was performed. Despite the higher number of binary interventions for inaccurate models, participants in all groups changed the model task label at similar rates (5–8 per group).

5.6.2.1 Human-machine Task Alignment

Measuring if participants are labelling samples the same as the CBMs we show the alignment of task labels in Table 5.12. Alignment with all samples averages 77.3% which perhaps seems a little low considering the model task accuracy is 99.8% and 96.4% for the accurate and inaccurate model respectively. However, as participants are playing a game, they may have strategies that are not aligned with the training data. Splitting the samples between those with interventions and those without, without interventions maintains an average alignment of 77.1% while with interventions increases to 86.7%.

Splitting between the participant groups we can see the accurate model consistently result in higher alignment than the inaccurate model. With interventions, human-machine alignment is consistently higher than without interventions.

We also analysed alignment w.r.t. participants' self-reported Blackjack skill, as shown in Table 5.13. Among participants who selected strongly disagreed, disagreed, or neutral, alignment remained between 74.3% and 79.2% for the accurate model, and 74.5% and 75.4% for the inaccurate model, both with overlapping error bars. Participants with strongly disagree skill aligned with the accurate model more often than participants with other skill levels. This may show these participants are less confident and thus are more likely to rely on the model's predictions. Participants with a skill level of agree or strongly agree both showed a higher alignment with the accurate model compared to lower skill levels, while their alignment with the inaccurate model was consistent with lower skill participants. These results show that although alignment increases for skilled

Data subset	Overall (%)	Initial model task prediction alignment (%)	Intermediate model task prediction alignment (%)	Final model task prediction alignment (%)
All	77.3 (±0.8)	77.1 (±0.8)	81.5 (±5.3)	83.3 (±4.9)
NoInt	77.1 (±0.8)	77.1 (±0.8)	-	-
WithInt	**86.7 (±4.4)**	76.7 (±5.5)	81.5 (±5.3)	83.3 (±4.9)
Acc	**80.1 (±1.1)**	80.0 (±1.1)	93.8 (±6.2)	90.0 (±6.9)
Inacc	74.6 (±1.2)	74.3 (±1.2)	76.3 (±7.0)	80.0 (±6.4)
Acc-NoExp	**79.8 (±2.2)**	79.8 (±2.2)	-	-
Inacc-NoExp	70.4 (±2.6)	70.4 (±2.6)	-	-
Acc-CExp	**84.5 (±2.0)**	84.5 (±2.0)	-	-
Inacc-CExp	73.9 (±2.5)	73.9 (±2.5)	-	-
Acc-CExp+Int-NoInt	78.8 (±2.4)	78.8 (±2.4)	-	-
Acc-CExp+Int -WithInt	**92.9 (±7.1)**	78.6 (±11.4)	90.0 (±10.0)	92.9 (±7.1)
Inacc-CExp+Int-NoInt	74.8 (±2.5)	74.8 (±2.5)	-	-
Inacc-CExp+Int -WithInt	**76.2 (±9.5)**	66.7 (±10.5)	73.7 (±10.4)	71.4 (±10.1)
Acc-CExp+Int+SMap -NoInt	76.0 (±2.5)	76.0 (±2.5)	-	-
Acc-CExp+Int+SMap -WithInt	**100 (±0.0)**	100 (±0.0)	100 (±0.0)	83.3 (±16.7)
Inacc-CExp+Int+SMap -NoInt	78.5 (±2.4)	78.5 (±2.4)	-	-
Inacc-CExp+Int+SMap -WithInt	**89.5 (±7.2)**	78.9 (±9.6)	78.9 (±9.6)	89.5 (±7.2)

Table 5.12: Lay-person study human-machine task alignment.

participants, the overall trends remain close to the average alignment across all participants.

Finally, we observed a general increase in alignment between the initial and final

Data subset	Overall (%)	Initial model task prediction alignment (%)	Intermediate model task prediction alignment (%)	Final model task prediction alignment (%)
Acc-Strongly Disagree	79.2% (±2.5)	79.2% (±2.5)	-	-
Inacc-Strongly Disagree	74.8% (±2.1)	74.3% (±2.1)	78.6% (±11.4)	73.3% (±11.8)
Acc-Disagree	74.3% (±3.4)	74.3% (±3.4)	66.7% (±33.3)	66.7% (±33.3)
Inacc-Disagree	75.4% (±2.7)	75.0% (±2.7)	50.0% (±22.4)	83.3% (±16.7)
Acc-Neutral	77.5% (±2.5)	77.5% (±2.5)	100.0% (±0.0)	100.0% (±0.0)
Inacc-Neutral	74.5% (±2.5)	74.5% (±2.5)	100.0% (±0.0)	100.0% (±0.0)
Acc-Agree	84.3% (±1.8)	83.8% (±1.8)	100.0% (±0.0)	92.9% (±7.1)
Inacc-Agree	73.1% (±2.9)	72.7% (±2.9)	81.2% (±10.1)	82.4% (±9.5)
Acc-Strongly Agree	82.3% (±3.6)	82.3% (±3.6)	100.0% (±0.0)	100.0% (±0.0)
Inacc-Strongly Agree	76.6% (±6.2)	76.6% (±6.2)	-	-

Table 5.13: Lay-person study human-machine task alignment with participant Blackjack skill.

model task predictions for all groups. The only notable exceptions to this were for Acc-CExp+Int+SMap participants, and participants whose skill rating was Strongly disagree and using the accurate model. In both cases, these participants intervened on the accurate model. It is therefore plausible for them to align with the model's initial task prediction, as interventions would only lower the model's task accuracy.

These findings demonstrate that interventions improve human-machine task alignment. We performed a one-tailed t-test with the null hypothesis that interventions do not increase alignment and the alternative hypothesis that they do *increase

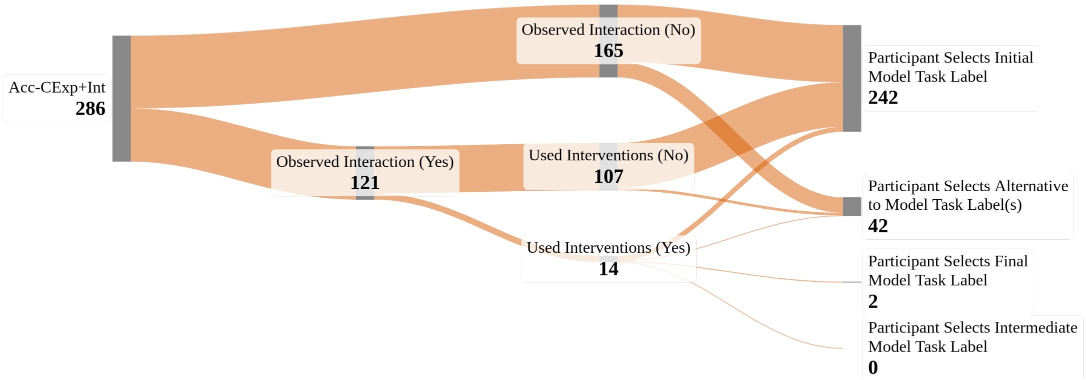


Figure 5.20: Most samples labelled by Acc-CExp+Int participants agreed with the initial model task prediction, irrespective if they interacted with the model or not.

alignment. The test resulted in a p-value of 0.041 when including all participant groups and 0.04 when focusing only on participants capable of performing interventions. Since both values are below our significance threshold of 0.05, we can accept our alternative hypothesis and conclude that interventions improving human-machine alignment is statistically significant. Further, we also perform a t-test evaluating the significance of providing just concept explanations compared to no explanations. Participants in the group CExp had a higher human-machine alignment compared to participants in the NoExp group with a p-value of 0.036. Finally, to assess whether participants' self-reported Blackjack skill influenced alignment, we performed a one-way Analysis of Variance test. This test produced an F-statistic of 1.636 and a p-value of 0.162, showing that participant skill level does not significantly change alignment.

Overall, these results highlight the potential of both interventions and concept explanations to increase human-machine task alignment. In addition, the difference in performance between participants with and without interventions is greater than the difference between the accurate and inaccurate models.

Figures 5.20, 5.21, 5.22, and 5.23 visualise how each sample is labelled for Acc-CExp+Int, Inacc-CExp+Int, Acc-CExp+Int+SMap, and Inacc-CExp+Int+SMap

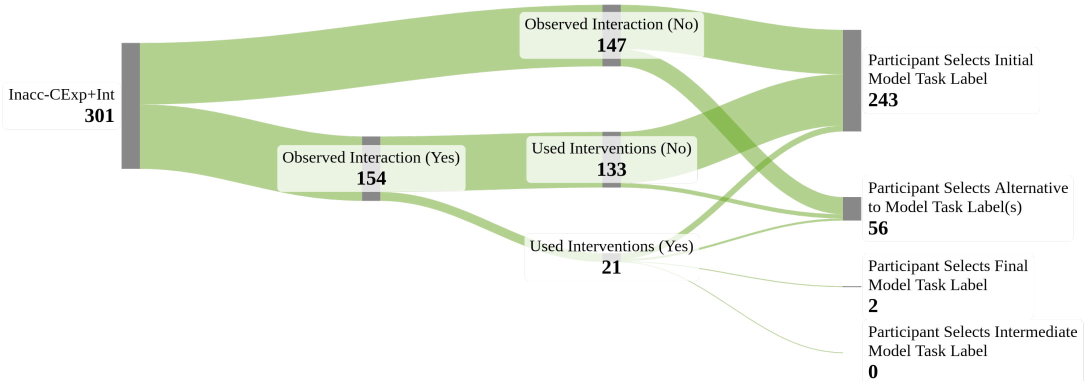


Figure 5.21: Most samples labelled by Inacc-CExp+Int participants agreed with the initial model task prediction, irrespective if they interacted with the model or not. Compared to the accurate model, more samples without observed interaction resulted in an alternative task label to the model prediction selected.

participants, respectively, based on the flowchart in Figure 5.1. Across all participant groups, a similar number of game moves were made with and without the model's ouput and explanations. This suggests that while participants do not always rely on the model, they use it when they find it useful. As some participants playing styles are different from the model's, the largest number of alternative task labels are when participants do not interact with the model.

An unexpected observation was the number of labels selected by participants who (1) had observed model interaction but selected the initial model label without performing interventions, and (2) selected an alternative label without performing interventions. These findings suggest that concept explanations enable participants to evaluate the model's decision-making process without the need for counterfactual explanations, allowing them to either accept or reject the model's task predictions.

Comparing alignment in Table 5.12 to task accuracy in Table 5.14, we observe that participants without access to a model had the lowest task accuracy among all participant subsets. When models were enabled, participants who performed

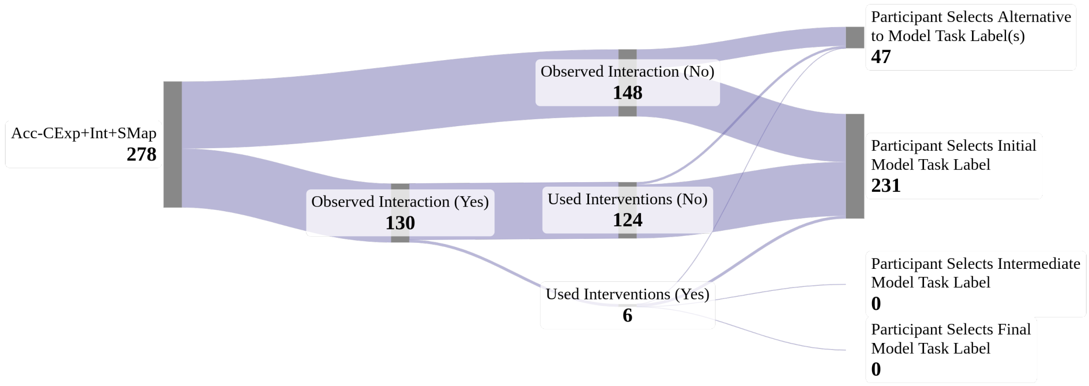


Figure 5.22: Most samples labelled by Acc-CExp+Int+SMap participants agreed with the initial model task prediction. Most labels disagreeing with the initial model task label prediction did not include model interaction.

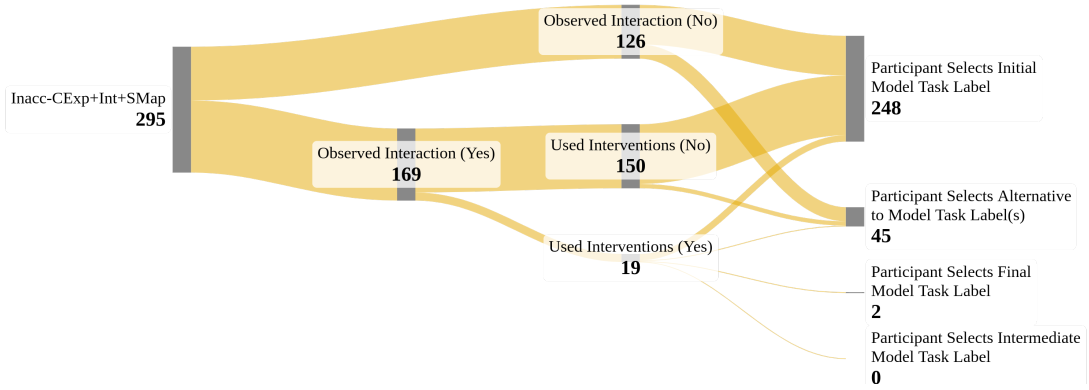


Figure 5.23: Most samples labelled by Inacc-CExp+Int+SMap participants agreed with the initial model task prediction, with most of these for samples that included model interaction but no interventions. As with the accurate model, most alternative task labels selected were for samples without model interaction.

interventions did not achieve higher task accuracy than those who did not, despite the increase in alignment previously discussed. However, accurate models showed an observable improvement in human accuracy. In particular, participants using the accurate model had higher task accuracy when they did not perform interven-

Data Subset	Accuracy (%)
AI disabled	74.4 (±3.9)
All	83.6 (±0.9)
WithInt	83.3 (±1.1)
NoInt	84.7 (±1.8)
Acc	84.6 (±2.7)
Inacc	78.1 (±3.2)
Acc-NoExp	84.6 (±2.7)
Inacc-NoExp	78.1 (±3.2)
Acc-CExp	91.0 (±2.4)
Inacc-CExp	81.4 (±1.6)
Acc-CExp+Int-NoInt	86.6 (±3.3)
Acc-CExp+Int-WithInt	83.8 (±2.9)
Inacc-CExp+Int-NoInt	75.7 (±4.6)
Inacc-CExp+Int-WithInt	84.3 (±1.8)
Acc-CExp+Int+SMap-NoInt	83.4 (±2.4)
Acc-CExp+Int+SMap-WithInt	83.4 (±6.4)
Inacc-CExp+Int+SMap-NoInt	83.5 (±2.9)
Inacc-CExp+Int+SMap-WithInt	86.6 (±3.6)

Table 5.14: Lay-person study task accuracy averaged by participant.

tions, while for the inaccurate model, the opposite was observed. This indicates participants are more likely to trust a model if they perform interventions, which, in the case of the accurate model from this study, leads to over-trust.

We also show accuracy for participants separated by their self-reported Blackjack skill in Table 5.15. As with alignment, there is not a large difference between participants of different skill levels. Participants with a skill level of Agree and Strongly Agree achieved the highest accuracy of any skill level. A one-way Analysis of Variance test resulted in an F-statistic of 1.613 and a p-value of 0.177, showing that participant skill level does not significantly change human task ac-

Data Subset	Accuracy (%)
Acc-Strongly Disagree,	93.1% (±2.8)
Inacc-Strongly Disagree	90.1% (±4.2)
Acc-Disagree	93.8% (±3.0)
Inacc-Disagree	80.5% (±4.9)
Acc-Neutral	91.4% (±2.8)
Inacc-Neutral	90.2% (±3.3)
Acc-Agree	96.1% (±1.7)
Inacc-Agree	89.7% (±3.2)
Acc-Strongly Agree	89.2% (±7.4)
Inacc-Strongly Agree	95.5% (±4.5)

Table 5.15: Lay-person study task accuracy with participant Blackjack skill and averaged by participant.

curacy.

Participants in the CExp group demonstrated a statistically significant improvement in task accuracy compared to those in the NoExp group. Using a one-tailed t-test we comparing the two groups, resulting in a p-value of 0.042, which is below the significance threshold of 0.05. This result suggests that providing detected concepts alone helps participants interpret the model and identify obvious mistakes.

In contrast, participants in the CExp+Int and CExp+Int+SMap groups achieved lower task accuracy with the accurate model compared to CExp participants, but higher task accuracy with the inaccurate model. These findings indicate that interventions improve task accuracy when the model makes incorrect predictions. However, when the model is accurate, interventions appear to mislead participants, reducing their overall accuracy.

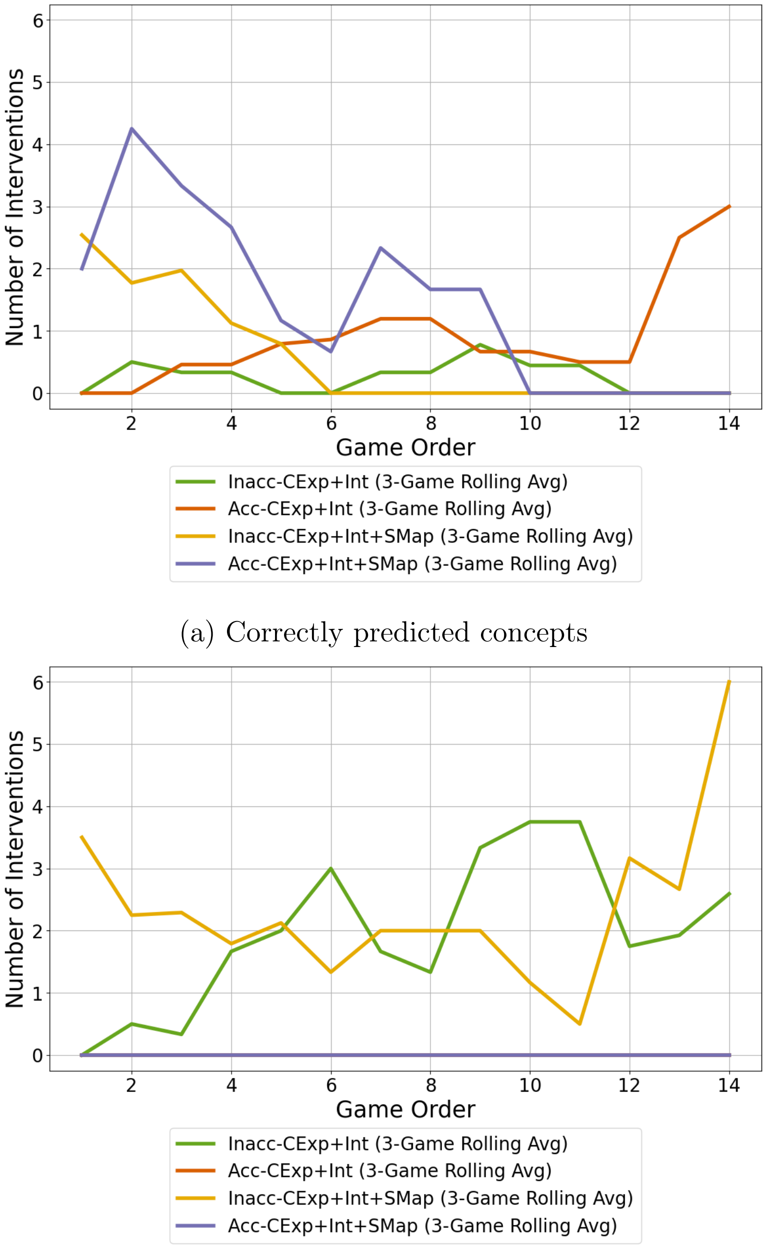
Performing a one-tailed t-test to compare participants who performed interventions to those who did not resulted in a p-value of 0.27, indicating no statistically significant difference in task accuracy for participants who used interventions compared to those who did not across all groups. In addition, only comparing task accuracy for participants who performed interventions to those who did not but had access to interventions, resulted in a p-value of 0.195, which also fails to meet our significance threshold. While we observed a trend of higher average accuracy among participants using interventions, these results are not statistically significant, and we cannot conclude that interventions directly improve task accuracy.

Overall, our findings indicate that concepts are beneficial for improving human-machine alignment and this leads to improvements in human accuracy. Interventions are beneficial for increasing human-machine task alignment but do not result in a statistically significant increase in task accuracy. Overtrust can occur when participants perform interventions with an accurate model. Interventions provide the largest benefit when the model makes incorrect concept predictions and thus the increased alignment from interventions with this model leads to a higher human accuracy.

5.6.2.2 Interventions Over Time

Interventions performed in order of games played are displayed in Figure 5.24. To observe the trend of interventions we average the number of interventions performed over three games. When the models make accurate concept predictions (Figure 5.24a), the average number of interventions per sample starts at 2 for CExp+Int+SMap participants, and 0 for CExp+Int participants. Over time the average number of interventions performed per sample seen decrease to around 0 - 1.

An exception to this general trend of decreasing interventions occurs between games 13 and 14, where participants in the Acc-CExp+Int group display a spike in interventions. This is caused by game 13, where the average number of inter-



(a) Correctly predicted concepts

(b) Incorrectly predicted concepts

Figure 5.24: Interventions performed by participants show a decline over time when the models correctly predict concepts. This is evidence participants initially explore the model's capabilities and sensitivity to concept values. When the models incorrectly predict concepts the number of interventions performed remains constant.

ventions is 7, with a standard error of ±2 and thus, this spike is not representative of all participants.

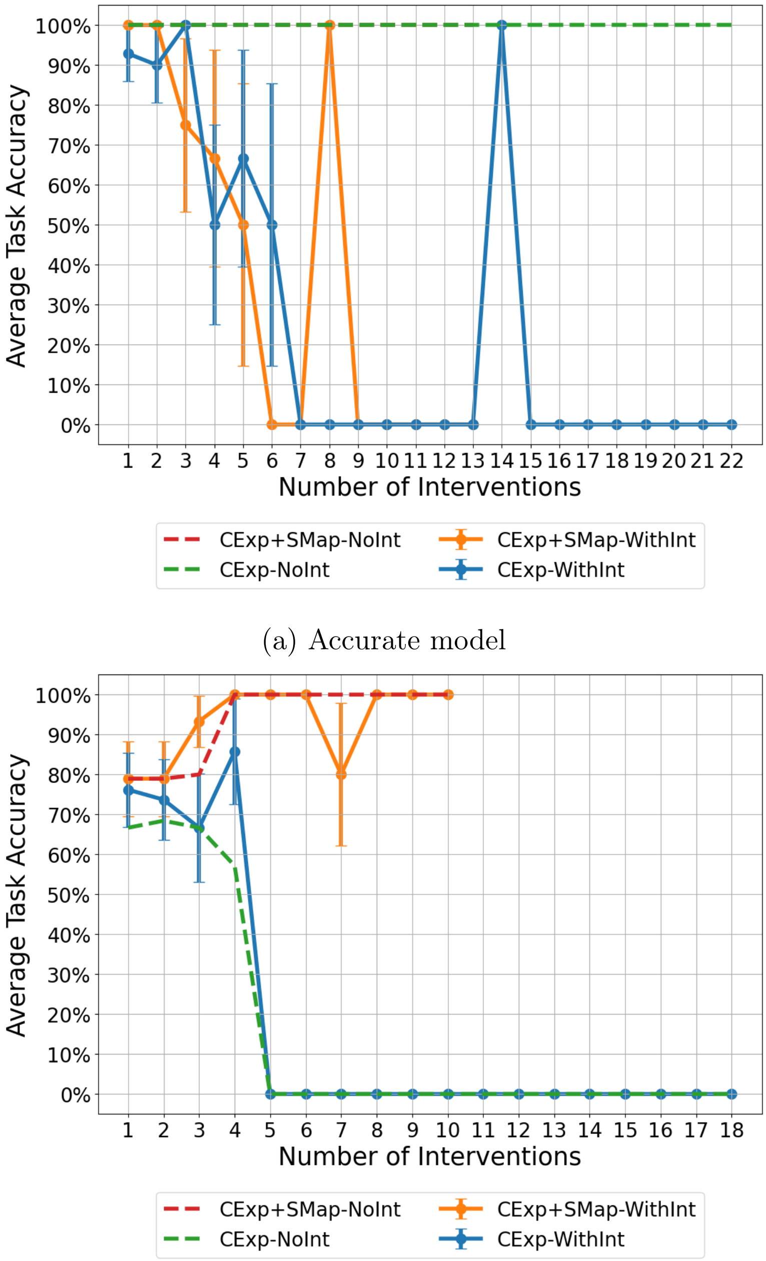
For incorrect concept predictions, both participant groups with the inaccurate model consistently performed around 2 interventions per sample. These results demonstrate participants identify concepts that need to be intervened on while ignoring the concepts that are correctly predicted by the models.

Overall, similar to the expert study, we observe more interventions at the beginning of the study, even if concepts are correctly predicted by the model. This shows that participants initially explore the model's capabilities and sensitivity to concept values before developing a mental model and reducing the number of interventions performed to where it is required.

5.6.2.3 Test-time Intervention and Concept Accuracy

In Figure 5.25 we show the test-time intervention results for both our accurate model and inaccurate model. We compare our participant groups that had access to interventions to the model task accuracy without interventions. As with the expert study, we do not include samples that were not intervened on to calculate the model task accuracy.

Starting with the accurate model in Figure 5.25a, as expected, the model alone is fixed at 100% accuracy. Acc-CExp+Int and Acc-CExp+Int+SMap participants achieve almost the same task accuracy as each other when performing interventions. Between 1 and 3 interventions the task accuracy falls slightly but overall remains around 90-100%. As interventions enable participants to learn about the model's sensitivity to changing concept predictions, this observed task accuracy is expected. While accuracy declines further with more interventions, we cannot draw further conclusions due to limited data. Only a single sample was intervened on 6 or more times.



(a) Accurate model

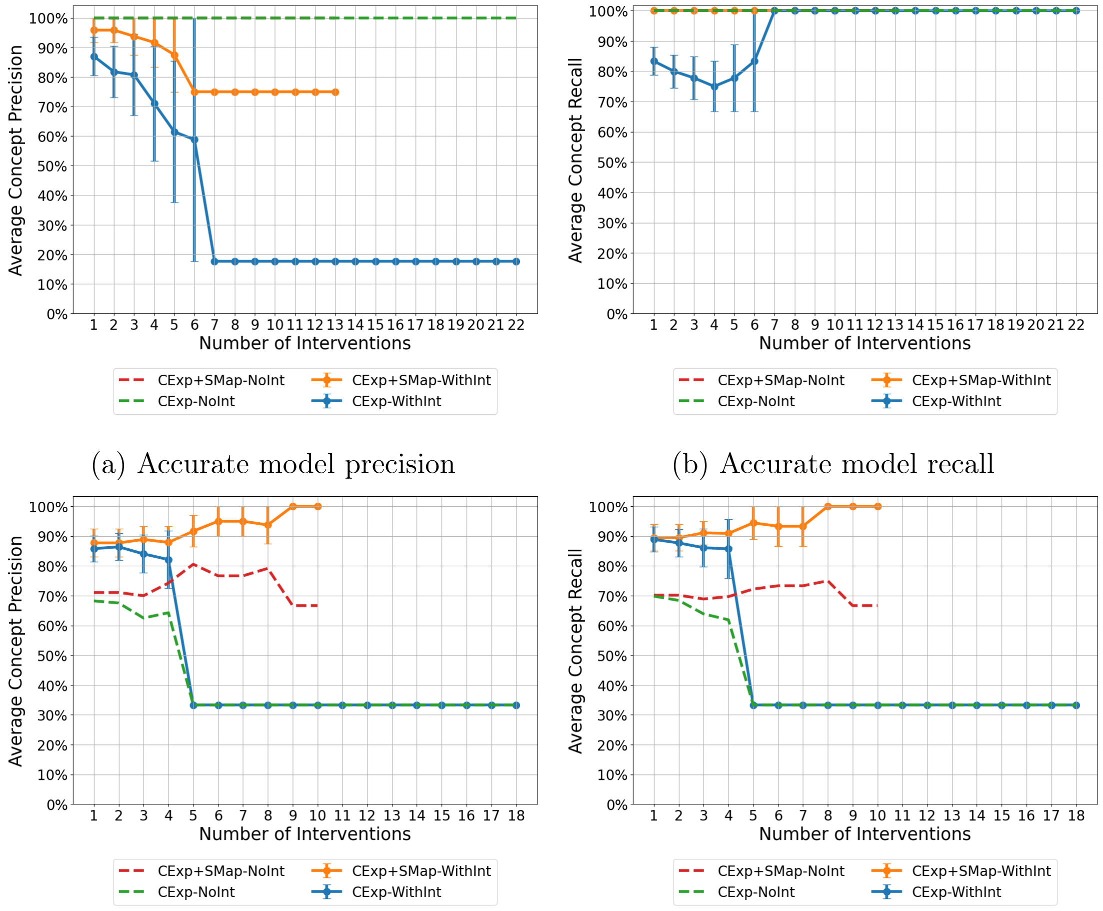
(b) Inaccurate model

Figure 5.25: Test-time interventions for the lay-person study shows interventions match the accurate model accuracy, and makes a small increase to the inaccurate model accuracy.

With the inaccurate model in Figure 5.25b we can see the model task accuracy ranges from 80% to 100% for Inacc-CExp+Int+SMap, and 70% for Inacc-CExp+Int. For participants in these groups, interventions either match or increase the model task accuracy. From 1 to 4 interventions, we can see a clear benefit to participants intervening on concept predictions which increases the model task accuracy. After 4 interventions there is little deviation from the model task accuracy with no interventions. Inacc-CExp+Int+SMap-WithInt stays around 100%, while Inacc-CExp+Int-WithInt falls to 0%. As with the accurate model, conclusions after 4 interventions are limited due to insufficient data. Specifically, Inacc-CExp+Int+SMap-WithInt participants only performed 4 or more interventions on 6 samples which dropped to 1 sample after 7 interventions. Inacc-CExp+Int-WithInt participants performed 4 or more interventions on just one sample.

Concept precision and recall with interventions are shown in Figure 5.26. Starting with the accurate model we have more insight into how interventions change concept accuracy. Figure 5.26a shows interventions are lowering the percentage of correctly predicted present concepts (e.g. by setting concepts from not present to present). Precision only decreases as interventions increase. As the bars for standard error overlap, there is not a meaningful difference between CExp+Int+SMap participants and CExp+Int participants. Moving to Recall in Figure 5.26b, interventions start off by lowering recall before increasing from 4 interventions. This shows that over time participants appear to be attempting to correct concept values. In other words, this is evidence participants are exploring the model's sensitivity to concept values. However, we also need to draw attention to the increased size of error bars which highlights that not all participants followed this pattern.

With the inaccurate model, the precision of concepts with interventions is consistently higher than without interventions. Both participant groups increase concept precision by 15 to 20% over the initial model concept predictions. Overall interventions show a meaningful increase to concept precision when the model incorrectly predicts concepts. Concept recall in Figure 5.26d tells the same story. Interventions improve concept recall over the model with no interventions. With



(a) Accurate model precision

(b) Accurate model recall

(c) Inaccurate model precision

(d) Inaccurate model recall

Figure 5.26: Interventions with the accurate model mostly resulted in an initial decrease for concept precision and recall. However, recall increased after the initial decrease. Interventions with the inaccurate model increased both precision and recall.

saliency maps, this increase with recall is up to 100% accuracy, while without saliency maps it decreases a little over time, but is still far higher than the model with no interventions. As before, after 4 interventions we do not have enough samples to make any further conclusions.

Combining concept test-time intervention results with task accuracy test-time intervention results, we can clearly see interventions lower both concept accuracy and task accuracy when the model is accurate. We hypothesise that participants used interventions to better understand the model's decision boundaries; however, this may lead to over reliance on the model's task predictions and ultimately decrease human task accuracy. This highlights a limitation of current intervention mechanisms: they override the model's concept predictions. As concept predictions also serve as a form of model confidence, once overridden, this information is lost. In contrast, with an inaccurate model, interventions improve concept accuracy and either maintain or increase task accuracy. This may still reflect some over reliance on the model's task predictions. However, as concept accuracy improves and the task predictor does not have any known biases, task performance does not decrease. It is also possible that with inaccurate models, incorrect concept predictions are more apparent to participants, making it easier to decide when to intervene.

5.6.2.4 System Causability Scale and Participant Feedback

Responses from the SCS questions for the lay-person study are shown in Table 5.16. All overall scores are 0.70 or above with the lowest at 0.70 for CExp+Int-WithInt and CExp+Int+SMap-WithInt participants, and 0.71 for Inacc-NoExp participants. The highest score was 0.78 for Acc-CExp participants. Surprisingly, participants with the accurate model but no concept explanations overall score was 0.74 which matches some of the other participant groups who had access to concepts and interventions.

For the *factors in data* question (*I found that the data included all relevant known causal factors with sufficient precision and granularity*), Inacc participants consistently scored lower than Acc participants. However, the score gap narrowed when interventions and saliency maps were introduced, indicating that additional information was helpful for causal evaluation. Interestingly, participants who performed interventions tended to score lower than those who did not, suggesting that performing interventions might not improve a participant's perceived causal understanding.

Question	All Participants	Acc-NoExp	Inacc-NoExp	Acc-CExp	Inacc-CExp	Acc-CExp+Int	Inacc-CExp+Int	Acc-CExp+Int+SMap	Inacc-CExp+Int+SMap	CExp+Int-WithInt and CExp+Int+SMap-WithInt	CExp+Int-NoInt and CExp+Int+SMap-NoInt
Factors in data	3.39	3.08	2.85	**3.92**	3.69	3.62	3.50	3.23	3.25	2.87	3.25
Understood	4.18	4.08	4.15	**4.31**	**4.31**	4.23	4.17	4.08	4.08	4.13	4.08
Change detail level	2.91	2.69	2.31	2.77	2.38	3.38	**3.50**	3.31	3.00	3.13	3.00
Need support	3.77	3.92	3.92	3.77	3.62	3.54	**4.17**	3.54	3.75	3.61	3.75
Understanding causality	3.51	3.15	3.38	3.62	3.46	3.31	3.75	3.46	**4.00**	3.43	**4.00**
Use with knowledge	3.99	4.15	3.92	4.23	3.62	3.92	3.92	3.85	**4.33**	3.91	**4.33**
No inconsistencies	3.59	3.77	3.38	**4.15**	3.54	3.54	3.08	3.77	3.42	2.96	3.42
Learn to understand	4.06	**4.31**	4.15	4.00	4.15	3.92	4.00	3.92	4.00	3.74	4.00
Needs references	3.64	**4.00**	3.77	3.85	3.46	3.54	3.33	3.54	3.58	3.13	3.58
Efficient	4.24	4.08	3.85	4.15	**4.62**	4.15	4.42	4.08	4.58	4.26	4.58
Overall score	0.75	0.74	0.71	**0.78**	0.74	0.74	0.76	0.74	0.76	0.70	0.76

Table 5.16: Lay-person study Likert scores for SCS questions.

Understood (*I understood the explanations within the context of my work.*), *learn to understand* (*I think that most people would learn to understand the explanations very quickly*), and *efficient* (*I received the explanations in a timely and ef-*

ficient manner) received consistently high scores. However, WithInt participants answered *Understood* and *learn to understand* questions with a lower score than NoInt participants. In particular *learn to understand* received the lowest score out of all participants subsets if the participants performed interventions. Similar to our previous discussion, this highlights interventions may not be aligned with human understanding. In this case, this result indicates interventions are not easy to understand.

The mixed responses continue with *Understanding causality* (*I found the explanations helped me to understand causality*) where scores generally improved with the inclusion of explanation techniques. However, for *use with knowledge*, scores were higher for NoInt participants, implying that interventions might have introduced confusion in what concepts mean or reflected a mismatch between model and participant strategies.

The *no inconsistencies* (*I did not find inconsistencies between explanations*) question saw lower scores for inaccurate models, likely reflecting participants' recognition of model errors as inconsistencies. Again, participants performing interventions scored this question lower, reinforcing concerns about the alignment between the model and participants' understanding. Finally, the *needs references question* decreased with additional explanation techniques but was lowest for participants performing interventions, which adds to the growing results suggesting concepts may not be aligned or fully understood by participants.

The SCS results highlight that incorporating concepts improves participants' understanding of causality, with model outputs generally being well understood. Participants also agreed that the explanations allowed for greater control over detail levels. However, interventions also received low scores for multiple questions, suggesting participants struggled to understand how concept values influenced task labels, and how concepts influence the model's decision-making process. This may indicate either a lack of alignment between model reasoning and participants' understanding of the task or insufficient clarity in the intervention design.

We also provide participant comments to provide additional insights. Participants in the NoExp group generally found the AI's suggestions "very helpful" or "useful to a certain degree." However, one participant noted that they "do not remember any explanations as to what the AI was doing." This response aligns with the study design, as only the task label output was provided to these participants. When predicting task labels, participants were not guaranteed to gain insight into the causes of incorrect predictions, leading one participant to conclude that the "AI agent was mostly correct by statistical analysis." However, repeated interactions meant some participants were able to identify some model biases, such as the observation that "the AI seemed to always assume an ace was 1."

Participants in the CExp group appreciated the model's ease of use, stating it was "easy to use and understand." Despite the addition of some concept predictions for the model's decision-making process, participants expressed a desire for a "more detailed explanation," noting that its absence "made it hard to trust the AI suggestions." One participant highlighted that a probabilistic breakdown would have improved their interaction. While concept values were not directly revealed, some participants noticed symptoms of model inaccuracies, such as "the AI was slightly inconsistent with its counting," though they did not determine the underlying cause.

In the CExp+Int group, participants commented "the AI was detailed in explaining its rationale behind decisions". However, others found the suggestions "useful but incomplete", and suggested that "a more complete explanation of why a certain decision was recommended would be appreciated". Participants were better equipped to identify the causes of model bias due to the additional concepts details and intervention capability provided. For instance, they could "easily detect when the system was wrong about the captured concepts" or noted challenges like the model "struggling to read the player card values in a few instances". However, this ability was not universal, as some participants still believed the AI was occasionally incorrect, without expanding to what they believed the model bias was. One participant also found interventions “confusing” and they “didn’t understand the point of it”. This further reinforces our discussion that research practitioners and users should collaborate when designing models for human-machine settings. Participants from the CExp+Int+SMap group provided no explicit feedback on the use or utility of saliency maps, suggesting these were overlooked. From our results previously discussed, saliency maps do not harm human collaboration and show small improvements over no saliency maps.

Presenting participant’s comments quantitatively we show the percentage of participants per group that mention the model or explanations in Table 5.17. In particular, we counted participants’ comments that mentioned the model bias, explanations, and whether they found the model useful or not. If no comment was provided, or it did not cover one of these topics we counted their response as a “no comment”.

Out of the participant groups, those who had access to interventions were most likely to identify the model bias with these participants also finding the concept explanations the most useful. Comments about model usefulness appeared to be primarily tied to the participant’s existing skill, and whether the model was accurate or not. A few participants commented about using the model until they lost a game, at which point they would start ignoring it. No participants identified all biases with the inaccurate model, but a number of participants identified an incorrect bias where they believed the model was miscounting cards. Although this is correlated with the model bias, these participants did not identify the cause of the incorrect card totals.

The participant feedback corroborates our findings that CBMs enhance interpretability by revealing model inner workings by breaking down an end-to-end DNN with an intermediate output of high-level concepts. However, while CBMs are more interpretable than a standard DNN, their decision-making process and sensitivity to concepts are often hidden unless users actively intervene on a model to inspect changes to the task label. As shown by our test-time intervention results, a human will often have to make multiple interventions to reliably correct any model's inaccurate predictions. We have identified future research should evaluate whether the delivery of concepts and models reasoning presents a barrier to human understanding and undermines trust in the model.

As we had previously stated, the model and participants may have learnt different stargates for how to play Blackjack. This was commented on by participants too who said the model "did not fit into my existing strategies for the game." This misalignment highlights a gap in human-machine collaboration, where the AI's decision-making process must be transparent and compatible with human decision-making for effective interaction.

Data Subset	Identified full model bias (%)	Identified partial model bias (%)	Identified incorrect model bias (%)	No comment about model bias (%)	Explanation useful (%)	Explanation partially useful (%)	Explanation not useful (%)	No comment about explanation (%)	Model useful (%)	Model not useful (%)	No comment about model (%)
Acc-NoExp	7.7	0	7.7	84.6	7.7	0	0	92.3	15.4	7.7	76.9
Inacc-NoExp	0	15.4	7.7	76.9	0	0	7.7	92.3	30.8	23.1	41.2
Acc-CExp	0	0	7.7	92.3	15.4	0	0	84.6	30.8	0	69.2
Inacc-CExp	0	15.4	15.4	69.2	7.7	0	7.7	84.6	15.4	15.4	69.2
Acc-CExp+Int	7.7	0	23.1	69.2	**23.1**	0	0	76.9	15.4	15.4	69.2
Inacc-CExp+Int	0	**53.8**	15.4	30.8	7.7	15.4	7.7	69.2	15.4	7.7	76.9
Acc-CExp+Int+SMap	**30.8**	0	7.7	61.5	7.7	7.7	15.4	69.2	38.5	15.4	46.2
Inacc-CExp+Int+SMap	0	**38.5**	7.7	53.8	7.7	15.4	15.4	61.5	38.5	0	61.5

Table 5.17: Summarising comments made by participants in the lay-person study. More participants with access to concepts and interventions identified the model bias, and found concept explanations to be useful.

5.7 Discussion

From our human studies evaluating CBMs, we observed mixed results regarding interpretability and task performance with human collaboration. While our findings reinforce CBMs interpretability with participants who utilised concepts and interventions to explore the concept space and inspect task predictions, task accuracy improvements were inconsistent. Notably, increases in concept accuracy and task alignment through interventions did not consistently translate into an increase in task accuracy. In this section we discuss our results w.r.t. RQ3 and the three related sub-questions.

5.7.1 Do Test-time Interventions Improve Human Task and Concept Accuracy?

Test-time interventions found mixed results across models. In most cases, task accuracy with 1 to 3 interventions matched or slightly underperformed the model's accuracy with no interventions, which further declined in task accuracy as the number of interventions increased. We did not observe any improvements in task accuracy in the expert study, while in the lay-person study, we only observed improvements when the model had a known bias. However, interventions generally helped participants understand the model's decision-making process.

Unlike the automated metrics reported in prior work (Koh et al., 2020), we did not observe consistent accuracy improvements as interventions increased for humans performing interventions. Further, as our expert study is more closely aligned to situations where AI may be deployed in the real world, it suggests interventions may lead to decreases in task accuracy instead of increases for more complex tasks.

Despite the limited increase in task accuracy, test-time interventions with human participants consistently lead to increases in concept accuracy. In the expert study, both precision and recall improved with interventions, outperforming or matching the model task accuracy with no interventions performed. Similarly, in the lay-person study participants improved precision and recall for the inaccurate model, though we, as expected, observed lower precision and recall after interventions for the accurate model which were corrected in the precision results. These findings indicate that participants effectively corrected concept predictions, even when this did not directly lead to an increase in task accuracy.

Our findings support results from (Barker et al., 2023), suggesting that CBMs task predictions may use different concepts than humans use. While interventions successfully correct concept predictions, their limited improvements in task accuracy indicate that task-relevant concepts in CBMs are either not fully understood or misaligned with human intuition. Future research should explore methods to align CBM task reasoning with human understanding to better leverage their interpretability benefits.

5.7.2 Do Interventions Increase the Interpretability of Concept Bottleneck Models?

In the expert study, interventions were almost evenly split between error correction and feature adjustments. Further, intervention frequency decreased over time, suggesting that participants initially relied on interventions to understand the model but required fewer interventions as they developed a mental model of its behaviour. This aligns with the idea that CBMs improve interpretability.

Participants with access to saliency maps performed fewer interventions overall, which were mostly reversed with subsequent interventions. This suggests that saliency maps provided sufficient insight into the model's behaviour to reduce the need for interventions. However, participants also reported they placed little weight on the model's predictions, implying that participants preferred their own intuition than relying on the model. For deployments of CM systems this means the current format of concept explanations and interventions likely will not be utilised to their full extent. Future work should look into how this class of model can be adapted for improved human interaction.

Regarding the lay-person study, for participants using the accurate models, interventions were primarily feature adjustments, with nearly 75% involving changes to concept presence. However, the number of intervention reversals was unexpectedly low compared to total interventions. We would have expected the reversal number to be close to half the total interventions to indicate all interventions are reverted. However, as observed with the concept recall, concepts are corrected over time which indicates participants are using the interpretability of concepts to improve their understanding of the model. More interventions, including reversals, were performed with the inaccurate model. In addition, some participant comments included praise for understanding how the model made decisions.

While we observe a decline in interventions over time in both studies, part of this decline may be attributed to the novelty of interventions, with engagement naturally decreasing as participants became more familiar with the task. Although we cannot entirely rule out this effect, the fact that the decline is not uniform, particularly in the lay-person study, where interventions remained higher when concepts were incorrectly predicted suggests that participants were not merely losing interest but actively leveraging interventions to improve their understanding of the model.

Our findings support the claim that CBMs improve interpretability by allowing users to interactively query and adjust concept predictions. Interventions enable users to understand the model's sensitivity to concept values and the resulting task predictions. However, we identified limitations in the efficiency of this process. Interventions require users to take an active role in seeking explanations and iteratively probing the model's concept sensitivity, which may not be practical or obvious for all users or applications. In addition, providing concept predictions without interventions results in similar improvements in task accuracy compared to a standard DNN, with interventions only adding value when a model has an obvious bias. We suggest future research should look at the delivery of concept explanations to ensure they are efficiently provided to humans.

5.7.3 Are Concept Bottleneck Models Trusted?

As trust is a difficult metric to measure, we have opted to use alignment as a proxy (Rong et al., 2024). We hypothesised that alignment should increase when participants perform interventions as we may assume participants will continue to perform interventions until they agree with the model task output. To quantify if human-machine alignment is justified we compared alignment to participant accuracy. If alignment is high the participant accuracy should also be higher than participant accuracy with low alignment.

In the expert study, participants who did not use interventions aligned to the model's predictions 81% of the time, which is 11% higher than the model's accuracy. This suggests over-trust. In contrast, participants who used interventions were aligned to the model's initial task prediction 66% of the time, 4% lower than the model's accuracy. Alignment then increased by almost 13% after interventions. This shows interventions increased the trust given to the model. Trust for participants who performed interventions is also better justified compared to participants who did not use interventions. Accuracy was higher for participants who used interventions compared to those who did not.

For the lay-person study, alignment was generally lower than the model's accuracy before interventions. This does not necessarily imply a lack of trust in the model as we know participants may use different strategies than that of the models. When participants used interventions, alignment increased significantly across all participant groups. This increase in alignment also translated into improved task accuracy for participants with the inaccurate model, showing the increase in alignment, and thus trust was justified. However, participants using the accurate model did not see the same increase in human task accuracy which shows interventions can also lead to overtrust.

Separating interventions from concept explanations, participants without the ability to perform interventions showed increased alignment for the accurate model compared to the inaccurate model. Participants in both of these groups were also able to show an increase in human task accuracy. This shows by just providing some of the model decision-making process our participants were able to better justify their trust in the models.

The trends of alignment and task accuracy are conflicting between the studies. We hypothesis this is because in the expert study the model outputs were used purely as a second opinion as the participants would have sufficient expertise in the task domain. As this is not guaranteed in the lay-person study we believe participants may follow the model if they are unsure themselves. This is a concerning point if these models are deployed in situations where humans are not domain experts.

5.8 Limitations

This study has several limitations that should be considered when interpreting the results. These limitations fall into three main categories: participant-related, methodological, and task constraints.

5.8.1 Participant-Related Limitations

The expert study involved a small number of participants due to the difficulty involved with recruiting domain specialists. As a result, findings from this study are not statistically significant and should be interpreted as exploratory. To address this, we conducted a larger lay-person study and drew parallels between the two groups to provide additional context.

5.8.2 Data Limitations

The expert study lacked access to patient history, high-quality diagnostic images, or multiple image views that are typically available in clinical settings with tools such as (Nextech Systems, 2024) that combine images and patient history in one place. To mitigate this, the task was simplified to distinguish between "malignant melanoma" and "seborrhoeic keratosis", which have clear visual differences.

Although we collect objective results for participant behaviour, we did not collect all subjective results such as perceived trust, confidence in the model, or predictions of the model's outputs. While participant comments were intended to capture these insights, many responses lacked the desired detail which limited our ability to perform detailed analysis.

5.8.3 Task Limitations

Participants in the lay-person study played games of Blackjack while interacting with a model. While the game simplified the creation of the CBM based agent where we could be sure of its correctness, and that concept was predicted using semantically meaningful input features, we could not assess overall game success because winning or losing can depend on luck, even when optimal moves are made. While this does not limit our ability to evaluate actions based on the optimal moves, it also leads to participants having their own preferences and play styles. As we observed, this resulted in lower-than-expected alignment between participants and the model.

5.9 Conclusion

In this chapter we ran the first studies to evaluate how humans use CBMs. These studies ask participants to complete a task with a CBM as an assistant. Our analysis is focused on how concepts are interacted with and the interpretability of these models. In particular we evaluate (1) if concept interventions increase the models task accuracy, (2) do concepts interventions increase model interpretability, and (3) are CBMs trusted.

This chapter answers RQ3: Do Concept Models improve task accuracy and model interpretability in a human-machine setting? We find CBMs do not translate to increased model task accuracy in a human-machine setting, but this model architecture and other CMs are shown to increase both the interpretability and trust with the model's task label predictions.

We conducted two studies: a small-scale study with dermatology experts, and a larger lay-person study using a Blackjack-based task. A CBM acted as an AI assistant in both settings, providing either diagnostic suggestions or strategic game moves. From these studies, we draw three main conclusions:

Firstly, interventions significantly improved concept accuracy but had limited impact on task accuracy. This suggests a misalignment between the concepts humans find useful, and the concepts the model's task predictor used to label samples. Addressing this misalignment is critical to improving the effectiveness of human-machine teams.

Next, we show the initial promise of interpretability from high-level concepts and interpretability is upheld with CMs. However, as this required participants to engage in interventions, this highlights a need for CBMs to present their decision-making process more proactively, reducing the cognitive effort required from humans. In addition, much of the interpretability can be provided by just providing concept predictions.

Finally, using alignment as a proxy for trust, we found that interventions led to higher trust. In this expert study this higher trust lead to increased task accuracy. In the lay-person study increased trust lead to higher task accuracy for participants with the inaccurate model, but lower task accuracy for participants with the accurate model, thus showing overtrust. Providing just concept explanations without the ability to intervene did not result in overtrust. This highlights the importance of interpretable models, that are evaluated with human participants, to enable the creation of trust that is suitably applied to a model.

Chapter 6

Conclusion and Future Work

In this chapter, we summarise the contributions of this thesis, reflect on how these contributions address the original research questions, and discuss future research in this research area.

This thesis has evaluate the interpretability and effectiveness of CBMs for human-machine collaboration. We have addressed research gaps addressing how these models, and other similar CMs, are trained and analysing their capabilities in real-world human studies. Despite their proposed potential, early research questioned their interpretability, and no human studies had validated their claimed benefits of improvements to model task accuracy or interpretability in a human-machine collaborative environment. This thesis focused on three core areas: (1) identifying the dataset attributes required to train them to learn to predict concepts using semantically meaningful input features, (2) Investigating input feature attribution and information leakage, and (3) Conducting the first human studies to evaluate how CBMs are used in real-world human-machine tasks.

6.1 Research Questions and Contributions

6.1.1 Concept Bottleneck Model Feature Attribution

Our first research question and its linked contributions are as follows:

> **RQ1**: How can we train a CBM to map semantically meaningful input features to concepts, and semantically meaningful concept predictions to task labels?

> **RC1**: We perform qualitative and quantitative analysis of CBMs, finding CBMs are capable of learning semantically meaningful concept representations from input features.
>
> **RC2**: We introduce and publish a new synthetic image dataset with fine-grained concept annotations which we use to demonstrate instances when CBMs can learn semantically meaningful concept representations and when they fail to do so.
>
> **RC3**: We expand on existing literature by looking at feature attribution both from the input to the concept vector and from the concept vector to the task output.

In Chapter 3, we used XAI techniques to analyse the input feature attribution for concept predictions and the concept feature attribution for task predictions. Directly answering RQ1, we can train CBMs to map semantically meaningful input features to concepts by training these models on datasets with a clear link between input features and task labels, void of ambiguous links (e.g. concepts annotated as present but without the corresponding visual representation in sample images). Further, CBMs can be trained to map semantically meaningful concept predictions to task labels by training these models with a sigmoid function between the two model parts and by using the independent training method.

Our findings reveal that the properties of the dataset significantly influence how CBMs learn concept representations. Specifically, we demonstrate that ensuring concepts have consistent and clear representations in input images enables CBMs to predict concepts based on semantically meaningful input features (RC1). The availability of instance-level concept annotations helps to facilitate the training of CBMs. Additionally, we show that the independent training method and including a sigmoid function between the two model parts produces a model that closely aligns feature attribution applied to concept predictions and the ground truth concept values (RC3). The configuration of concepts in a dataset does not significantly effect the alignment between concept predictions feature attribution and ground truth values.

To support these findings, we created a new synthetic image dataset based on playing cards, where the concepts correspond to individual cards (e.g. Three of Hearts), and the task is to classify card hands for the game Three Card Poker (RC2). We identified the need for this dataset because existing image datasets were often too noisy or lacked the required information for evaluation. Our dataset includes multiple variations which changed the inter-concept correlation, and used either class-level concept annotations, or instance-level concept annotations. Importantly for our analysis, we included ground truth segmentations for semantically meaningful pixels.

6.1.2 Input Feature Sensitivity and Information Leakage

Our second research question and its associated contributions are as follows:

> **RQ2**: How does the relationship between concepts and input features in the training dataset influence the information encoded in learned concepts and the model's reliance on input features for predicting those concepts?

> **RC4**: We perform an in-depth evaluation of CBMs revealing CBMs can be trained to minimise the encoding of extraneous information in concept representations, and concepts can be resilient to irrelevant input feature alterations. We demonstrate that CBMs generally learn underlying concept correlations present in the training data.

> **RC5**: We conclude that two factors are critical for CBMs to learn semantically meaningful input features: (i) accuracy of concept an-

notations and (ii) high variability in the combinations of concepts co-occurring, that is, each concept in a dataset should appear alongside a variety of others to help the model distinguish between them.

In Chapter 4, we analysed CBMs by examining the information encoded in learned concept representations and the reliance concept predictions are to the inclusion of semantically meaningful input features. As in Chapter 3, we evaluated CBMs trained on three datasets: CUB, Playing Cards, and CheXpert. This enabled us to investigate how concept configurations influence the representations that CBMs learn with respect to these two key metrics.

To answer RQ2, we identified several critical findings for how the configuration of concepts affects the information encoded into CBMs:

1. CBMs are sensitive to inter-concept correlation in the training data. Our experiments evaluating concept leakage revealed that high inter-concept correlations lead to the encoding of additional information in concept predictions (RC4).

2. CBMs can achieve resilience to modifications of irrelevant input features when trained on instance-level concept annotations (RC4).

Additionally, we detail the properties datasets require to ensure CBMs are trained to predict concepts using semantically meaningful input features, minimise information leakage, and reduce concept predictions changing from the modification of irrelevant input features (RC5). Each concept in the dataset should consistently align with a clear link to, input features to provide an strong training signal. Additionally, the training dataset should include a high verity of co-occurring concept combinations.

These findings pose implications for creating CBMs for real-world applications. Primarily, limiting training to datasets with instance-level concept annotations limits these models for domains with readily available datasets, or datasets where their creation is easy (e.g. synthetic or those with a small number of samples). Future work is required to expand the suitability of these models to datasets outside of these constraints.

6.1.3 Human Studies

Our third research question and its linked contributions are as follows:

> **RQ3**: Do Concept Models improve task accuracy and model interpretability in a human-machine setting?
>
> **RC6**: We perform the first human studies using CBMs in a joint human-machine task setting which analyses the interaction between humans and the CBM. We find participants who performed interventions increased trust in a model, but this trust was sometimes misplaced. Additionally, the CBM decision-making process is not aligned to that of the humans.
>
> **RC7**: We show providing concept explanations to humans increases both model interpretability and task accuracy. In addition, interventions can be used to reveal model bias. This upholds the model's promise of increasing interpretability from high-level concepts.

To address RQ3, we conducted two human studies to evaluate how humans interact with CBMs in real-world tasks. These studies focused on the effectiveness of interventions for improving both task accuracy and the interpretability of the model's decision-making process. The first study was small in scale and involved recruiting dermatology experts to participate. The second study was larger in scale and used lay participants playing games of Blackjack.

We find that the claimed increase in model task accuracy by performing interventions does not translate to a human-machine setting. Participants who performed interventions corrected errors in the model's concept predictions, achieving higher concept accuracy. This highlights a misalignment between the CBM's and human decision-making processes. Despite this misalignment, CBMs prove to be interpretable and are trusted more than non-interpretable models. Additionally, participants who performed interventions achieved a higher task accuracy. However, the corresponding improvement in task accuracy was small and not statistically significant (RC6).

On the other hand, the capability of CBMs to improve interpretability through concept predictions was upheld (RC7). Participants with access to concept explanations trusted the model's predictions more than those without. Participants who had access to concept predictions but could not intervene achieved a statistically significant increase in task accuracy compared to those without concept explanations. This demonstrates that concept predictions alone can improve human understanding of a model, even without direct interaction.

Participants with access to interventions actively explored the concept space, evaluated the models using counterfactual explanations and were able to identify and correct concept errors. Despite this, participants are required to initiate this capability of the model which leads to a small improvement in task accuracy which was not statistically significant.

6.2 Future Work

This section outlines future research directions building on the findings of this thesis. The primary focus is advancing CMs and their integration in to human-machine collaborative settings. We present this in three subsections: Model and dataset enhancements, human-machine collaboration, and generative model enhancements.

6.2.1 Model and Dataset Enhancements

Chapters 3 and 4 demonstrated that while CBMs can learn to predict concepts using semantically meaningful input features, achieving this often requires datasets with specific properties. This includes a comprehensive set of concept annotations, which can be resource-intensive to produce. These limitations restrict scalability beyond specialised contexts, as also identified by (Kazhdan et al., 2021; Oikarinen et al., 2023; Selvaraj et al., 2024; Zang et al., 2024; Ismail et al., 2024). Recent advancements like CLIP (Radford et al., 2021) have been proposed to address this limitation by automating dataset annotation (Kim et al., 2023a). Further, CLIP has been combined with LLMs to generate concepts for training CMs, and filtering irrelevant ones (Oikarinen et al., 2023; Yang et al., 2023; Wang et al., 2024b). However, these methods do not guarantee that generated concepts consistently correspond to semantically meaningful input features, as seen in (Oikarinen et al., 2023), where the concept "long tail with white stripes" failed to align with an image where the tail was obscured.

Future work should focus on improving the alignment of generated concepts to semantically meaningful regions of input data while minimising inter-concept correlation. Generative models, such as Generative Adversarial Networks (Goodfellow et al., 2014) or diffusion models (Dhariwal and Nichol, 2021), is an alternative path of future research that could reduce dependence on large annotated datasets by generating dataset samples with similar visual properties to manually annotated datasets. No matter which direction future research goes, it is important to validate using the same class of metrics introduced in this thesis to ensure they capture the intended representations of concepts.

Additionally, models could benefit from enhancements in architecture and training methods to align their learned representations of concepts with human-interpretable features. For example, Enhanced CBMs (Huang et al., 2024), uses a prototype model to localise concepts to image regions through patch-based comparisons.

Similarly, replacing the standard CNN feature extractor with model architectures designed for object detection, such as You Only Look Once (Redmon et al., 2016), could enable the model to detect the presence and absence of concepts more effectively while removing the limitations caused by the dataset. Models such as (Losch et al., 2019), already employ this approach.

An alternative approach that may align the machine's decision-making process with the humans would be the introduction of hybrid models that combine DNN-based models with interpretable neuro-symbolic approaches (Roig Vilamala et al., 2021). Specifically, the CBM concept encoder would be paired with a symbolic task predictor. This would allow the concept encoder to detect the presence of concepts, while the task predictor will predict task labels based on the concept predictions. Because the symbolic component is rule-based and inherently interpretable, the decision-making process can be inspected to ensure it aligns with domain knowledge or human reasoning, and adjustments can be made if misalignment is found.

6.2.2 Human-machine Collaboration

Chapter 5 demonstrated that CMs with concept explanations and interventions enhance interpretability, but interventions impose the need to be performed multiple times to reveal a model's sensitivity to concepts. Future work should focus on the development of models that transparently communicate their decision-making processes, reducing the need for extensive human interaction.

One of the key limitations of our studies was the limited number of participants for the expert study. Expanding this research to a large-scale expert study in a high-stakes domain, or a domain that requires input from multiple agents, would validate findings and explore whether interventions can be proven to increase task accuracy.

Finally, human studies are not common in AI research (Nauta et al., 2023) despite being crucial for validating interpretability claims. Future research must incorporate evaluation with human participants, ideally in real-world tasks to ensure that interpretability methods are both technically sound and practically applicable (Doshi-Velez and Kim, 2017).

6.2.3 Generative Model Enhancements

As previously discussed, LLMs offer opportunities to enhance the creation of datasets or otherwise facilitate automatic data annotation. A second area of future research we've identified is integrating CMs with generative models. This can enable the generated outputs of a resulting model to contain human-understandable concepts as demonstrated by Ismail et al. (2024). The model architecture Ismail et al. (2024) introduces demonstrates the potential of combining generative models with interpretable concept outputs, which can ensure the outputs a model generates are predictable. Additionally, the concept vector can aid in model debugging. However, this approach still requires training data with concept annotations, which is the same limitation as with CBMs.

Another avenue for generative models to address is the misalignment issues identified in our human studies. Because CBMs are trained on ground truth concept values, applying interventions post-training often results in out-of-distribution concept vectors. However, to the best of our knowledge, there are no datasets with intervention annotations. Incorporating human-in-the-loop methods during training has been shown to reduce annotation dependency (Russakovsky et al., 2015; Chauhan et al., 2023). For CBMs, LLMs could simulate human-performed interventions, including varying confidence levels (concept values between 0 and 1). These simulated interventions could be integrated into a modified training process, enabling models to better handle human collaborative settings.

Bibliography

Amina Adadi and Mohammed Berrada. Peeking inside the black-box: A survey on explainable artificial intelligence (xai). *IEEE Access*, 6:52138–52160, 2018. doi: 10.1109/ACCESS.2018.2870052.

Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, page 9525–9536, Red Hook, NY, USA, 2018. Curran Associates Inc.

Julius Adebayo, Michael Muelly, Ilaria Liccardi, and Been Kim. Debugging tests for model explanations. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

Arjun Akula, Changsong Liu, Sinisa Todorovic, Joyce Chai, and Song-Chun Zhu. Explainable ai as collaborative task solving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019.

Ahmed Alqaraawi, Martin Schuessler, Philipp Weiß, Enrico Costanza, and Nadia Berthouze. Evaluating saliency map explanations for convolutional neural networks: a user study. In *Proceedings of the 25th International Conference on Intelligent User Interfaces*, IUI '20, page 275–285, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450371186. doi: 10.1145/3377325.3377519.

David Alvarez-Melis and Tommi 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, Red Hook, NY, USA, 2018. Curran Associates Inc.

Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador García, Sergio Gil-López, Daniel Molina, Richard Benjamins, Raja Chatila, and Francisco Herrera. Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai, 2019.

Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLOS ONE*, 10(7):1–46, 07 2015. doi: 10.1371/journal.pone.0130140. URL `https://doi.org/10.1371/journal.pone.0130140`.

Gagan Bansal, Besmira Nushi, Ece Kamar, Walter Lasecki, Dan Weld, and Eric Horvitz. Beyond accuracy: The role of mental models in human-ai team performance. In *HCOMP*. AAAI, October 2019.

Matthew Barker, Katherine M. Collins, Krishnamurthy Dvijotham, Adrian Weller, and Umang Bhatt. Selective concept models: Permitting stakeholder customisation at test-time. *ArXiv*, abs/2306.08424, 2023. URL `https://api.semanticscholar.org/CorpusID:259165192`.

Katie Barrett-Powell, Jack Furby, Liam Hiley, Marc Roig Vilamala, Harrison Taylor, Federico Cerutti, Alun Preece, Tianwei Xing, Luis Garcia, Mani Srivastava, and Dave Braines. An experimentation platform for explainable coalition situational understanding, 2020. URL `https://arxiv.org/abs/2010.14388`.

Catarina Belém, Vladimir Balayan, Pedro Saleiro, and Pedro Bizarro. Weakly supervised multi-task learning for concept-based explainability. In *WeaSuL: ICLR 2021 Workshop on Weakly Supervised Learning*, 2021.

Y. Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35:1798–1828, 08 2013. doi: 10.1109/TPAMI.2013.50.

J. M. Benitez, J. L. Castro, and I. Requena. Are artificial neural networks black boxes? *IEEE Transactions on Neural Networks*, 8(5):1156–1164, 1997.

Lukas Biewald. Experiment tracking with weights and biases, 2020. URL `https://www.wandb.com/`. Software available from wandb.com.

L. Breiman, J. Friedman, C.J. Stone, and R.A. Olshen. *Classification and Regression Trees*. Taylor & Francis, 1984. ISBN 9780412048418. URL `https://books.google.co.uk/books?id=JwQx-WOmSyQC`.

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Zac Hatfield-Dodds, Alex Tamkin, Karina Nguyen, Brayden McLean, Josiah E Burke, Tristan Hume, Shan Carter, Tom Henighan, and Christopher Olah. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2023.

John Brooke. Sus: A quick and dirty usability scale. *Usability Eval. Ind.*, 189, 11 1995.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

Carrie J. Cai, Jonas Jongejan, and Jess Holbrook. The effects of example-based explanations in a machine learning interface. In *Proceedings of the 24th International Conference on Intelligent User Interfaces*, IUI '19, page 258–262, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450362726. doi: 10.1145/3301275.3302289.

Arjun Chandrasekaran, Deshraj Yadav, Prithvijit Chattopadhyay, Viraj Prabhu, and Devi Parikh. It takes two to tango: Towards theory of ai's mind, 2017. URL https://arxiv.org/abs/1704.00717.

Prithvijit Chattopadhyay, Deshraj Yadav, Viraj Prabhu, Arjun Chandrasekaran, Abhishek Das, Stefan Lee, Dhruv Batra, and Devi Parikh. Evaluating visual conversational agents via cooperative human-ai games. In *Proceedings of the Fifth AAAI Conference on Human Computation and Crowdsourcing (HCOMP)*, 2017.

Kushal Chauhan, Rishabh Tiwari, Jan Freyberg, Pradeep Shenoy, and Krishnamurthy Dvijotham. Interactive concept bottleneck models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(5):5948–5955, Jun. 2023. doi: 10.1609/aaai.v37i5.25736.

Chaofan Chen, Oscar Li, Chaofan Tao, Alina Jade Barnett, Jonathan Su, and Cynthia Rudin. This looks like that: deep learning for interpretable image recognition. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, Red Hook, NY, USA, 2019. Curran Associates Inc.

Zhi Chen, Yijie Bei, and Cynthia Rudin. Concept whitening for interpretable image recognition. *Nature Machine Intelligence*, 2:1–11, 12 2020. doi: 10.1038/s42256-020-00265-z.

Hao-Fei Cheng, Ruotong Wang, Zheng Zhang, Fiona O'Connell, Terrance Gray, F. Maxwell Harper, and Haiyi Zhu. Explaining decision-making algorithms through ui: Strategies to help non-expert stakeholders. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, page 1–12, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450359702. doi: 10.1145/3290605.3300789.

K. J. W. Craik. *The nature of explanation.* University Press, Macmillan, Oxford, England, 1943. ISBN 0674568826.

Roxana Daneshjou, Kailas Vodrahalli, Roberto A. Novoa, Melissa Jenkins, Weixin Liang, Veronica Rotemberg, Justin Ko, Susan M. Swetter, Elizabeth E. Bailey, Olivier Gevaert, Pritam Mukherjee, Michelle Phung, Kiana Yekrang, Bradley Fong, Rachna Sahasrabudhe, Johan A. C. Allerup, Utako Okata-Karigane, James Zou, and Albert S. Chiou. Disparities in dermatology ai performance on a diverse, curated clinical image set. *Science Advances*, 8(32):eabq6147, 2022a. doi: 10.1126/sciadv.abq6147.

Roxana Daneshjou, Mert Yuksekgonul, Zhuo Ran Cai, Roberto Novoa, and James Y Zou. Skincon: A skin disease dataset densely annotated by domain experts for fine-grained debugging and analysis. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 18157–18167. Curran Associates, Inc., 2022b. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/7318b51b52078e3af28197e725f5068a-Paper-Datasets_and_Benchmarks.pdf.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 8780–8794. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/49ad23d1ec9fa4bd8d77d02681df5cfa-Paper.pdf.

Gabriele Dominici, Pietro Barbiero, Francesco Giannini, Martin Gjoreski, and Marc Langeinrich. Anycbms: How to turn any black box into a concept bottleneck model. In *CEUR Workshop Proceedings*, volume 3793, page 81 – 88, 2024.

Finale Doshi-Velez and Been Kim. Towards a rigorous science of interpretable machine learning, 2017. URL https://arxiv.org/abs/1702.08608.

Jeff Druce, James Niehaus, Vanessa Moody, David Jensen, and Michael L. Littman. Brittle ai, causal confusion, and bad mental models: Challenges and successes in the xai program, 2021. URL https://arxiv.org/abs/2106.05506.

Abhimanyu Dubey, Filip Radenovic, and Dhruv Mahajan. Scalable interpretability via polynomials. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=TwuColwZAVj.

Cian Eastwood and Christopher K. I. Williams. A framework for the quantitative evaluation of disentangled representations. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=By-7dz-AZ.

Mateo Espinosa Zarlenga, Pietro Barbiero, Zohreh Shams, Dmitry Kazhdan, Umang Bhatt, Adrian Weller, and Mateja Jamnik. Towards robust metrics for concept representation evaluation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(10):11791–11799, Jun. 2023. doi: 10.1609/aaai.v37i10.26392.

European Parliament and Council of the European Union. Regulation (EU)

2016/679 of the European Parliament and of the Council, 2016. URL https://data.europa.eu/eli/reg/2016/679/oj.

Thomas Fel, Ivan Rodriguez, and Thomas Serre. Harmonizing the object recognition strategies of deep neural networks with humans. *Advances in neural information processing systems*, 35:9432–9446, 12 2022.

Jack Furby, Daniel Cunnington, Dave Braines, and Alun Preece. Towards a deeper understanding of concept bottleneck models through end-to-end explanation. *R2HCAI: The AAAI 2023 Workshop on Representation Learning for Responsible Human-Centric AI*, 2023.

Jack Furby, Daniel Cunnington, Dave Braines, and Alun Preece. Can we constrain concept bottleneck models to learn semantically meaningful input features?, 2024. URL https://arxiv.org/abs/2402.00912.

Jack Furby, Daniel Cunnington, Dave Braines, and Alun Preece. The impact of concept explanations and interventions on human-machine collaboration. In *Explainable Artificial Intelligence*. Springer Nature, 2025.

Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A. Wichmann, and Wieland Brendel. Imagenet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bygh9j09KX.

Donald Geman, Stuart Geman, Neil Hallonquist, and Laurent Younes. Visual turing test for computer vision systems. *Proceedings of the National Academy of Sciences*, 112(12):3618–3623, 2015. doi: 10.1073/pnas.1422953112.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014. URL https://proceedings.neurips.cc/paper_files/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf.

Frantisek Grezl, Martin Karafiat, Stanislav Kontar, and Jan Cernocky. Probabilistic and bottle-neck features for lvcsr of meetings. In *2007 IEEE International*

Conference on Acoustics, Speech and Signal Processing - ICASSP '07, volume 4, pages IV–757–IV–760, 2007. doi: 10.1109/ICASSP.2007.367023.

Matthew Groh, Caleb Harris, Luis Soenksen, Felix Lau, Rachel Han, Aerin Kim, Arash Koochek, and Omar Badri. Evaluating Deep Neural Networks Trained on Clinical Images in Dermatology with the Fitzpatrick 17k Dataset . In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1820–1828, Los Alamitos, CA, USA, June 2021a. IEEE Computer Society. doi: 10.1109/CVPRW53098.2021.00201.

Matthew Groh, Caleb Harris, Luis Soenksen, Felix Lau, Rachel Han, Aerin Kim, Arash Koochek, and Omar Badri. Evaluating deep neural networks trained on clinical images in dermatology with the fitzpatrick 17k dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 1820–1828, June 2021b.

Jindong Gu, Yinchong Yang, and Volker Tresp. Understanding individual decisions of cnns via contrastive backpropagation. In C. V. Jawahar, Hongdong Li, Greg Mori, and Konrad Schindler, editors, *Computer Vision – ACCV 2018*, pages 119–134, Cham, 2019. Springer International Publishing. ISBN 978-3-030-20893-6.

Frank G. Halasz and Thomas P. Moran. Mental models and problem solving in using a calculator. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '83, page 212–216, New York, NY, USA, 1983. Association for Computing Machinery. ISBN 0897911210. doi: 10.1145/800045.801613.

Lena Heidemann, Maureen Monnet, and Karsten Roscher. Concept correlation and its effects on concept-based models. In *2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 4769–4777, 2023. doi: 10.1109/WACV56688.2023.00476.

Andreas Holzinger, André Carrington, and Heimo Müller. Measuring the quality of explanations: The system causability scale (scs): Comparing human and machine explanations. *KI - Künstliche Intelligenz*, 34, 01 2020. doi: 10.1007/s13218-020-00636-z.

Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE Con-*

ference on Computer Vision and Pattern Recognition (CVPR), July 2017. doi: 10.1109/CVPR.2017.243.

Qihan Huang, Jie Song, Jingwen Hu, Haofei Zhang, Yong Wang, and Mingli Song. On the concept trustworthiness in concept bottleneck models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(19):21161–21168, Mar. 2024. doi: 10.1609/aaai.v38i19.30109.

Jeremy Irvin, Pranav Rajpurkar, Michael Ko, Yifan Yu, Silviana Ciurea-Ilcus, Chris Chute, Henrik Marklund, Behzad Haghgoo, Robyn Ball, Katie Shpanskaya, Jayne Seekins, David A. Mong, Safwan S. Halabi, Jesse K. Sandberg, Ricky Jones, David B. Larson, Curtis P. Langlotz, Bhavik N. Patel, Matthew P. Lungren, and Andrew Y. Ng. Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):590–597, Jul. 2019. doi: 10.1609/aaai.v33i01.3301590.

Aya Abdelsalam Ismail, Julius Adebayo, Hector Corrada Bravo, Stephen Ra, and Kyunghyun Cho. Concept bottleneck generative models. In *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=L9U5MJJleF`.

Alon Jacovi, Ana Marasović, Tim Miller, and Yoav Goldberg. Formalizing trust in artificial intelligence: Prerequisites, causes and goals of human trust in ai. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, page 624–635, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450383097. doi: 10.1145/3442188.3445923.

Jeya Vikranth Jeyakumar, Joseph Noor, Yu-Hsi Cheng, Luis Garcia, and Mani Srivastava. How can i explain this to you? an empirical study of deep neural network explanation methods. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 4211–4222. Curran Associates, Inc., 2020. URL `https://proceedings.neurips.cc/paper/2020/file/2c29d89cc56cdb191c60db2f0bae796b-Paper.pdf`.

Jeya Vikranth Jeyakumar, Luke Dickens, Yu-Hsi Cheng, Joseph Noor, Luis Antonio Garcia, Diego Ramirez Echavarria, Alessandra Russo, Lance M. Kaplan, and Mani Srivastava. Automatic concept extraction for concept bottleneck-based video classification, 2022. URL https://openreview.net/forum?id=66kgCIYQW3.

Jeya Vikranth Jeyakumar, Ankur Sarker, Luis Antonio Garcia, and Mani Srivastava. X-char: A concept-based explainable complex human activity recognition model. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 7(1), mar 2023. doi: 10.1145/3580804.

Zhengbao Jiang, Frank F. Xu, Jun Araki, and Graham Neubig. How can we know what language models know? *Transactions of the Association for Computational Linguistics*, 8:423–438, 07 2020. ISSN 2307-387X. doi: 10.1162/tacl_a_00324.

P. N. Johnson-Laird. *Mental Models: Towards a Cognitive Science of Language, Inference, and Consciousness.* Harvard University Press, USA, 1986. ISBN 0674568826.

Ece Kamar. Directions in hybrid intelligence: complementing ai systems with human intelligence. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, IJCAI'16, page 4070–4073. AAAI Press, 2016. ISBN 9781577357704.

Harmanpreet Kaur, Harsha Nori, Samuel Jenkins, Rich Caruana, Hanna Wallach, and Jennifer Wortman Vaughan. Interpreting interpretability: Understanding data scientists' use of interpretability tools for machine learning. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, page 1–14, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450367080. doi: 10.1145/3313831.3376219.

Dmitry Kazhdan, Botty Dimanov, Helena Andres Terre, Mateja Jamnik, Pietro Liò, and Adrian Weller. Is disentanglement all you need? comparing concept-based & disentanglement approaches. *RAI workshop at The Ninth International Conference on Learning Representations 2021*, 2021.

Been Kim, Martin Wattenberg, Justin Gilmer, Carrie J. Cai, James Wexler, Fernanda B. Viégas, and Rory Sayres. Interpretability Beyond Feature Attribution: Quantitative Testing with Concept Activation Vectors (TCAV). In *Proceedings of the 35th International Conference on Machine Learning*, pages 2673–2682. PMLR, 2018.

Chanwoo Kim, Soham U. Gadgil, Alex J. DeGrave, Zhuo Ran Cai, Roxana Daneshjou, and Su-In Lee. Fostering transparent medical image ai via an image-text foundation model grounded in medical literature. *medRxiv*, 2023a. doi: 10.1101/2023.06.07.23291119. URL https://www.medrxiv.org/content/early/2023/06/12/2023.06.07.23291119.

Eunji Kim, Dahuin Jung, Sangha Park, Siwon Kim, and Sungroh Yoon. Probabilistic concept bottleneck models. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023b.

Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 12 2014.

Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 5338–5348. PMLR, 13–18 Jul 2020. URL http://proceedings.mlr.press/v119/koh20a.html.

Maximilian Kohlbrenner, Alexander Bauer, Shinichi Nakajima, Alexander Binder, Wojciech Samek, and Sebastian Lapuschkin. Towards best practice in explaining neural network decisions with lrp. *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7, 2019. URL https://api.semanticscholar.org/CorpusID:204823968.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012. URL https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

Todd Kulesza, Simone Stumpf, Margaret Burnett, Sherry Yang, Irwin Kwan, and Weng-Keen Wong. Too much, too little, or just right? ways explanations impact end users' mental models. In *2013 IEEE Symposium on Visual Languages and Human Centric Computing*, pages 3–10, 2013. doi: 10.1109/VLHCC.2013.6645235.

Todd Kulesza, Margaret Burnett, Weng-Keen Wong, and Simone Stumpf. Principles of explanatory debugging to personalize interactive machine learning. In *Proceedings of the 20th International Conference on Intelligent User Interfaces*, IUI '15, page 126–137, New York, NY, USA, 2015. Association for Computing Machinery. ISBN 9781450333061. doi: 10.1145/2678025.2701399.

Neeraj Kumar, Alexander C. Berg, Peter N. Belhumeur, and Shree K. Nayar. Attribute and simile classifiers for face verification. In *2009 IEEE 12th International Conference on Computer Vision*, pages 365–372, 2009. doi: 10.1109/ICCV.2009.5459250.

Vivian Lai and Chenhao Tan. On human predictions with explanations and predictions of machine learning models: A case study on deception detection. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, FAT* '19, page 29–38, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450361255. doi: 10.1145/3287560.3287590.

Christoph H. Lampert, Hannes Nickisch, and Stefan Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 951–958, 2009. doi: 10.1109/CVPR.2009.5206594.

Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551, 1989. doi: 10.1162/neco.1989.1.4.541.

Yann Lecun, Sumit Chopra, and Raia Hadsell. *A tutorial on energy-based learning*, chapter 1, page 70. MIT Press, 01 2006.

Q. Vera Liao and Kush R. Varshney. Human-centered explainable ai (xai): From algorithms to user experiences, 2022. URL https://arxiv.org/abs/2110.10790.

Rensis Likert. *A technique for the measurement of attitudes.* Archives of Psychology, New York, 1932.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 740–755, Cham, 2014. Springer International Publishing. ISBN 978-3-319-10602-1.

Zachary C. Lipton. The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue*, 16(3): 31–57, June 2018. ISSN 1542-7730. doi: 10.1145/3236386.3241340.

Joshua Lockhart, Daniele Magazzeni, and Manuela Veloso. Learn to explain yourself, when you can: Equipping concept bottleneck models with the ability to abstain on their concept predictions, 2022. URL `https://arxiv.org/abs/2211.11690`.

Max Losch, Mario Fritz, and Bernt Schiele. Interpretability beyond classification output: Semantic bottleneck networks, 2019. URL `https://arxiv.org/abs/1907.10882`.

A. Mahinpei, J. Clark, I. Lage, F. Doshi-Velez, and P. WeiWei. Promises and pitfalls of black-box concept learning models. In *proceeding at the International Conference on Machine Learning: Workshop on Theoretic Foundation, Criticism, and Application Trend of Explainable AI,*, volume 1, pages 1–13, 2021.

Emanuele Marconato, Andrea Passerini, and Stefano Teso. Glancenets: Interpretable, leak-proof concept-based models. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 21212–21227. Curran Associates, Inc., 2022. URL `https://proceedings.neurips.cc/paper_files/paper/2022/file/85b2ff7574ef265f3a4800db9112ce14-Paper-Conference.pdf`.

Andrei Margeloiu, Matthew Ashman, Umang Bhatt, Yanzhi Chen, Mateja Jamnik, and Adrian Weller. Do concept bottleneck models learn as intended? *ICLR 2021 Workshop on Responsible AI*, 2021. doi: 10.17863/CAM.80941.

Nesta Midavaine, Gregory Hok Tjoan Go, Diego Canez, Ioana Simion, and Satchit Chatterji. [re] on the reproducibility of post-hoc concept bottleneck models. *Transactions on Machine Learning Research*, 2024.

Tim Miller. Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence*, 267:1–38, 2019. ISSN 0004-3702. doi: https://doi.org/10.1016/j.artint.2018.07.007.

Mazda Moayeri, Keivan Rezaei, Maziar Sanjabi, and Soheil Feizi. Text-to-concept (and back) via cross-model alignment. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

Christoph Molnar. *Interpretable Machine Learning*. Christoph Molnar, 2 edition, 2022. URL `https://christophm.github.io/interpretable-ml-book`.

Grégoire Montavon, Alexander Binder, Sebastian Lapuschkin, Wojciech Samek, and Klaus-Robert Müller. *Layer-Wise Relevance Propagation: An Overview*, pages 193–209. Springer International Publishing, Cham, 2019. ISBN 978-3-030-28954-6. doi: 10.1007/978-3-030-28954-6_10.

Sheshera Mysore, Mahmood Jasim, Andrew Mccallum, and Hamed Zamani. Editable user profiles for controllable text recommendations. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '23, page 993–1003, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9781450394086. doi: 10.1145/3539618.3591677.

Meike Nauta, Jan Trienes, Shreyasi Pathak, Elisa Nguyen, Michelle Peters, Yasmin Schmitt, Jörg Schlötterer, Maurice van Keulen, and Christin Seifert. From anecdotal evidence to quantitative evaluation methods: A systematic review on evaluating explainable ai. *ACM Comput. Surv.*, 55(13s), July 2023. ISSN 0360-0300. doi: 10.1145/3583558.

LLC Nextech Systems. Dermatology EHR & Practice Management Software. `https://www.nextech.com/dermatology/ehr-pm-software`, 2024. [Accessed 05-12-2024].

Giang Nguyen, Daeyoung Kim, and Anh Nguyen. The effectiveness of feature attribution methods and its correlation with automatic evaluation scores, 2021.

Giang Nguyen, Mohammad Reza Taesiri, Sunnie S. Y. Kim, and Anh Nguyen. Allowing humans to interactively guide machines where to look does not always improve human-ai team's classification accuracy. In *The 3rd Explainable AI for Computer Vision (XAI4CV) Workshop at CVPR 2024*, 2024.

D. Norman. Some observations on mental models. *Human-Computer Interaction*, pages 241–244, 1983.

Wizard of Odds. Three card poker online for real money or free. `https://wizardofodds.com/play/three-card-poker/`, 2024. [Accessed 16-05-2025].

Tuomas Oikarinen, Subhro Das, Lam M. Nguyen, and Tsui-Wei Weng. Label-free concept bottleneck models. In *The Eleventh International Conference on Learning Representations*, 2023.

Scott Ososky, David Schuster, Elizabeth Phillips, and Florian Jentsch. Building appropriate trust in human-robot teams. *AAAI Spring Symposium - Technical Report*, pages 60–65, 01 2013.

Rohan Paleja, Muyleng Ghuy, Nadun R. Arachchige, Reed Jensen, and Matthew Gombolay. The utility of explainable ai in ad hoc human-machine teaming. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, NIPS '21, Red Hook, NY, USA, 2024. Curran Associates Inc. ISBN 9781713845393.

Andrea Papenmeier, Gwenn Englebienne, and Christin Seifert. How model accuracy and explanation fidelity influence user trust in ai. In *IJCAI Workshop on Explainable Artificial Intelligence (XAI)*, 2019. URL `https://sites.google.com/view/xai2019/home`.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, 2021.

Naveen Raman, Mateo Espinosa Zarlenga, Juyeon Heo, and Mateja Jamnik. Do concept bottleneck models respect localities?, 2024.

Vikram V. Ramaswamy, Sunnie S. Y. Kim, Ruth Fong, and Olga Russakovsky. Overlooked factors in concept-based explanations: Dataset choice, concept learnability, and human capability. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10932–10941, June 2023.

Sukrut Rao, Sweta Mahajan, Moritz Böhle, and Bernt Schiele. Discover-then-name: Task-agnostic concept bottlenecks via automated concept discovery. In *European Conference on Computer Vision*, 2024.

Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection, 2016. URL `https://arxiv.org/abs/1506.02640`.

Carlo Reverberi, Tommaso Rigon, Aldo Solari, Cesare Hassan, Paolo Cherubini, Giulio Antonelli, Halim Awadie, Sebastian Bernhofer, Sabela Carballal, Mário Dinis-Ribeiro, Agnès Fernández-Clotett, Glòria Esparrach, Ian Gralnek, Yuta Higasa, Taku Hirabayashi, Tatsuki Hirai, Mineo Iwatate, Miki Kawano, Markus Mader, and Andrea Cherubini. Experimental evidence of effective human–ai collaboration in medical decision-making. *Scientific Reports*, 12, 09 2022. doi: 10.1038/s41598-022-18751-2.

Marco Ribeiro, Sameer Singh, and Carlos Guestrin. Model-agnostic interpretability of machine learning. *ICML Workshop on Human Interpretability in Machine Learning*, 06 2016a. doi: 10.48550/arXiv.1606.05386.

Marco Tulio Ribeiro, Sameer Singh, and Carlos 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*, KDD '16, page 1135–1144, New York, NY, USA, 2016b. Association for Computing Machinery. ISBN 9781450342322. doi: 10.1145/2939672.2939778.

Marc Roig Vilamala, Tianwei Xing, Harrison Taylor, Luis Garcia, Mani Srivastava, Lance Kaplan, Alun Preece, Angelika Kimmig, and Federico Cerutti. Using DeepProbLog to perform Complex Event Processing on an Audio Stream. *arXiv e-prints*, art. arXiv:2110.08090, October 2021. doi: 10.48550/arXiv.2110.08090.

Y. Rong, T. Leemann, T. Nguyen, L. Fiedler, P. Qian, V. Unhelkar, T. Seidel, G. Kasneci, and E. Kasneci. Towards human-centered explainable ai: A survey of user studies for model explanations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(04):2104–2122, apr 2024. ISSN 1939-3539. doi: 10.1109/TPAMI.2023.3331846.

Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65 6:386–408, 1958. URL `https://api.semanticscholar.org/CorpusID:12781225`.

William Rouse and Nancy Morris. On looking into the black box. prospects and limits in the search for mental models. *Psychological Bulletin*, 100, 11 1986. doi: 10.1037/0033-2909.100.3.349.

Olga Russakovsky, Li-Jia Li, and Li Fei-Fei. Best of both worlds: Human-machine collaboration for object annotation. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2121–2131, 2015. doi: 10.1109/CVPR.2015.7298824.

Wojciech Samek, Grégoire Montavon, Sebastian Lapuschkin, Christopher J. Anders, and Klaus-Robert Müller. Explaining deep neural networks and beyond: A review of methods and applications. *Proceedings of the IEEE*, 109(3):247–278, 2021. doi: 10.1109/JPROC.2021.3060483.

Adriel Saporta, Xiaotong Gui, Ashwin Agrawal, Anuj Pareek, Steven Q. H. Truong, Chanh D. T. Nguyen, Van-Doan Ngo, Jayne Seekins, Francis G. Blankenberg, Andrew Y. Ng, Matthew P. Lungren, and Pranav Rajpurkar. Benchmarking saliency methods for chest x-ray interpretation. *Nature Machine Intelligence*, 4, 2022. ISSN 2522-5839. doi: 10.1038/s42256-022-00536-x.

Jan B. Schmutz, Neal Outland, Sophie Kerstan, Eleni Georganta, and Anna-Sophie Ulfert. Ai-teaming: Redefining collaboration in the digital era. *Current Opinion in Psychology*, 58:101837, 2024. ISSN 2352-250X. doi: 10.1016/j.copsyc.2024.101837.

Nithish Muthuchamy Selvaraj, Xiaobao Guo, Adams Wai-Kin Kong, and Alex Kot. Improving concept alignment in vision-language concept bottleneck models, 2024.

Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 618–626, 2017. doi: 10.1109/ICCV.2017.74.

Michael Shackleford. Blackjack Single Desk Strategy, 10 2023. URL `https://wizardofodds.com/games/blackjack/strategy/1-deck/`.

David Silver, Aja Huang, Christopher Maddison, Arthur Guez, Laurent Sifre, George Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of go with deep neural networks and tree search. *Nature*, 529:484–489, 01 2016. doi: 10.1038/nature16961.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *Workshop at International Conference on Learning Representations*, 2014.

Leon Sixt, Maximilian Granz, and Tim Landgraf. When explanations lie: Why many modified BP attributions fail. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 9046–9057. PMLR, 13–18 Jul 2020. URL `https://proceedings.mlr.press/v119/sixt20a.html`.

Leon Sixt, Martin Schuessler, Oana-Iuliana Popescu, Philipp Weiß, and Tim Landgraf. Do users benefit from interpretable vision? a user study, baseline, and dataset. In *International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=v6s3HVjPerv`.

Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda B. Viégas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise. *CoRR*, abs/1706.03825, 2017. URL `http://arxiv.org/abs/1706.03825`.

X. Sun, Z. Yang, C. Zhang, K. Ling, and G. Peng. Conditional gaussian distribution learning for open set recognition. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13477–13486, Los Alamitos, CA, USA, jun 2020. IEEE Computer Society. doi: 10.1109/CVPR42600.2020.01349.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 3319–3328. JMLR.org, 2017.

Mohammad Reza Taesiri, Giang Nguyen, and Anh Nguyen. Visual correspondence-based explanations improve ai robustness and human-ai team accuracy. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information*

Processing Systems, volume 35, pages 34287–34301. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/ddb8486bf9ee0fdeca1866a13a96e98e-Paper-Conference.pdf.

Harrison Taylor, Liam Hiley, Jack Furby, Alun Preece, and Dave Braines. Vadr: Discriminative multimodal explanations for situational understanding. In *2020 IEEE 23rd International Conference on Information Fusion (FUSION)*, pages 1–8, 2020. doi: 10.23919/FUSION45008.2020.9190215.

Richard Tomsett, Alun Preece, Dave Braines, Federico Cerutti, Supriyo Chakraborty, Mani Srivastava, Gavin Pearson, and Lance Kaplan. Rapid trust calibration through interpretable and uncertainty-aware ai. *Patterns*, 1(4): 100049, 2020. ISSN 2666-3899. doi: 10.1016/j.patter.2020.100049.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.

B. Wang, L. Li, Y. Nakashima, and H. Nagahara. Learning bottleneck concepts in image classification. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10962–10971, Los Alamitos, CA, USA, jun 2023a. IEEE Computer Society. doi: 10.1109/CVPR52729.2023.01055.

Bor-Shiun Wang, Chien-Yi Wang, and Wei-Chen Chiu. Mcpnet: An interpretable classifier via multi-level concept prototypes. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10885–10894, 2024a. doi: 10.1109/CVPR52733.2024.01035.

Jiaqi Wang, Pichao Wang, Yi Feng, Huafeng Liu, Chang Gao, and Liping Jing. Align2concept: Language guided interpretable image recognition by visual prototype and textual concept alignment. In *Proceedings of the 32nd ACM International Conference on Multimedia*, MM '24, page 8972–8981, New York, NY, USA, 2024b. Association for Computing Machinery. ISBN 9798400706868. doi: 10.1145/3664647.3680707.

Jingqi Wang, Peng Jiajie, Zhiming Liu, and Hengjun Zhao. Hqprotopnet: An evidence-based model for interpretable image recognition. In *2023 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2023b. doi: 10.1109/IJCNN54540.2023.10191863.

Xinru Wang and Ming Yin. Are explanations helpful? a comparative study of the effects of explanations in ai-assisted decision-making. In *Proceedings of the 26th International Conference on Intelligent User Interfaces*, IUI '21, page 318–328, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450380171. doi: 10.1145/3397481.3450650.

Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. doi: 10.1109/TIP.2003.819861.

Xinyue Xu, Yi Qin, Lu Mi, Hao Wang, and Xiaomeng Li. Energy-based concept bottleneck models: Unifying prediction, concept intervention, and probabilistic interpretations. In *International Conference on Learning Representations*, 2024.

Deshraj Yadav, Rishabh Jain, Harsh Agrawal, Prithvijit Chattopadhyay, Taranjeet Singh, Akash Jain, Shivkaran Singh, Stefan Lee, and Dhruv Batra. Evalai: Towards better evaluation systems for ai agents. *ArXiv*, abs/1902.03570, 2019. URL `https://api.semanticscholar.org/CorpusID:60440584`.

Yue Yang, Artemis Panagopoulou, Shenghao Zhou, Daniel Jin, Chris Callison-Burch, and Mark Yatskar. Language in a bottle: Language model guided concept bottlenecks for interpretable image classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19187–19197, 2023.

Wenwu Ye, Jin Yao, Hui Xue, and Yi Li. Weakly supervised lesion localization with probabilistic-cam pooling, 2020.

Ming Yin, Jennifer Wortman Vaughan, and Hanna Wallach. Understanding the effect of accuracy on trust in machine learning models. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, page 1–12, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450359702. doi: 10.1145/3290605.3300509.

Mert Yuksekgonul, Maggie Wang, and James Zou. Post-hoc concept bottleneck models. In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=nA5AZ8CEyow`.

Yuan Zang, Tian Yun, Hao Tan, Trung Bui, and Chen Sun. Pre-trained vision-language models learn discoverable visual concepts. *arXiv preprint arXiv:2404.12652*, 2024.

Mateo Espinosa Zarlenga, Pietro Barbiero, Gabriele Ciravegna, Giuseppe Marra, Francesco Giannini, Michelangelo Diligenti, Zohreh Shams, Frederic Precioso, Stefano Melacci, Adrian Weller, Pietro Lio, and Mateja Jamnik. Concept embedding models: beyond the accuracy-explainability trade-off. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2024. Curran Associates Inc. ISBN 9781713871088.

Jianming Zhang, Sarah Adel Bargal, Zhe Lin, Jonathan Brandt, Xiaohui Shen, and Stan Sclaroff. Top-down neural attention by excitation backprop. *International Journal of Computer Vision*, 2018. doi: 10.1007/s11263-017-1059-x.

Appendix

Appendices

In this appendix, we provide further details on datasets, model architectures and training methods, and additional results to supplement the content of this thesis. All experiments were run on two workstations. The first has two 12GB NVIDIA GeForce GTX 1080 Ti GPUs, Intel(R) Core(TM) i9-7900X CPU @ 3.30GHz and 64GB of system memory. The second workstation has a single 24GB NVIDIA Quadro RTX 6000 GPU, Intel(R) Core(TM) i9-10900K CPU @ 3.70GHz and 64GB of system memory. The machines run Ubuntu LTS. We estimate around 700 hours are required to train models and run all experiments, which includes training multiple models for each dataset. All random seeds to train models were selected using the shuf command to select a number between 0 and 1000.

A Feature Attribution

In Chapter 3 we evaluated our models with saliency maps. Extending the results in this chapter we have present additional saliency maps here that represent the feature attribution applied to input features from concepts.

A.0.1 CUB Input Feature Attribution

We show several concept predictions that cover concepts correctly predicted as present, concepts incorrectly predicted as present, concepts correctly predicted as not present and concepts incorrectly predicted as not present. Concepts correctly predicted as present are all concepts in Figure A1, “has_bill_shape ::dagger” and “has_underparts_color::white” in Figure A2 and “has_bill_length ::shorter_than_head” in Figure A3. Concept incorrectly predicted as present are “has_tail_pattern::solid” in Figure A2 and “has_upperparts_color::brown”, “has_breast_pattern::striped”, “has_bill_color::grey” and “has_breast_pattern ::striped” in Figure A3. Concepts correctly predicted as not present include “has _wing_pattern::spotted” in Figure A2 and “has_bill_shape::dagger” and “has _wing_color::grey” in Figure A4. Finally, concepts incorrectly predicted as not present are “has_wing_pattern::multi-colored” in Figure A2 and “has_leg_color ::buff”, “has_underparts_color::white” and “has_forehead_color::black” in Figure A4.

The general case for feature attribution from the concept vector back to the input image most salient regions, input features the model used for prediction(s), do not align to what a human may apply feature attribution values to. Instead, we can see the model makes concept predictions from the entire bird such as in Figure A1 for the independent and sequential models, or seemingly unrelated parts of the bird image as shown in Figure A2 where the eye of the bird is a particularly important feature to the concept predictions. With our models and LRP feature attribution results, the eye of the bird appears to be a common input feature to receive relevance.

Concepts with similar prediction values often share similar saliency maps, as shown in Figure A1 for the Independent and Sequential models and the concepts has_upperparts_color::brown, has_breast_pattern::striped and has_bill _length::shorter_than_head for the joint-with-sigmoid model in Figure A3. Although not a perfect match, and not occurring every time, positive and negative feature attribution values can reverse from concept to concept for the same input image and model, such that if a concept is predicted as present, then the areas that are have positive feature attribution values may have negative values for a concept predicted as not present. An example of this can be seen in Figure A2 for the model joint-without-sigmoid and the concepts has_underparts_color::white and has_wing_pattern::spotted.

Rarely concepts can appear to map to regions aligned with a human’s own understanding of a concept such as in Figure A2 for the concept has_bill_shape ::dagger and the model joint-with-sigmoid. In isolation, this may mislead a human to believe the model has made a prediction using the correct input features when in reality the same input features are also being used for other concept predictions which can be seen in some of the other saliency maps for the same input image and model.

Finally, from these saliency maps, we can see the models do not need to see the presence of a bird part to make a prediction about it, such as with the concept has_bill_length::shorter_than_head in Figure A3.

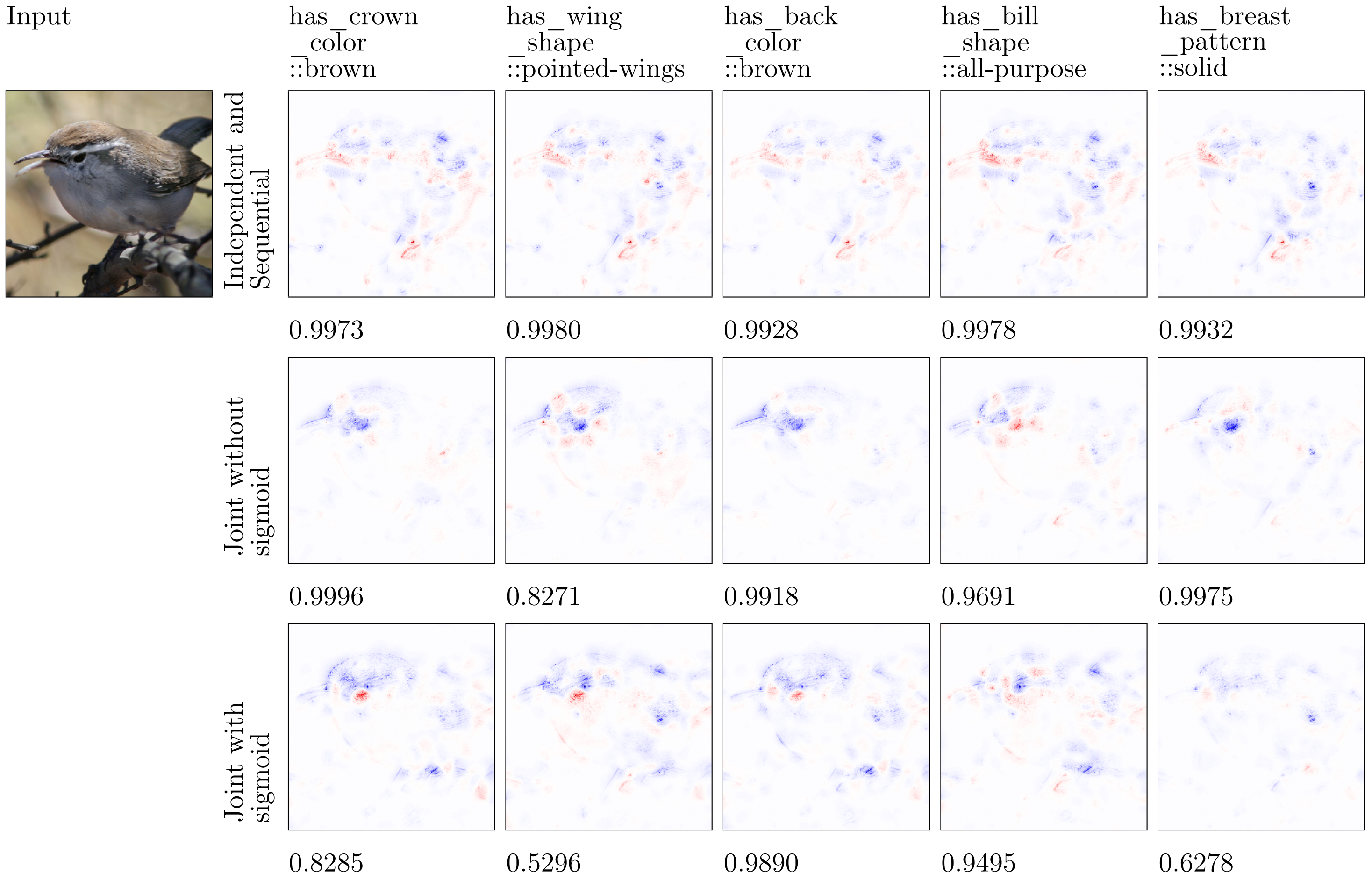


Figure A1: All concepts were correctly predicted as present. The input class was Bewick wren.

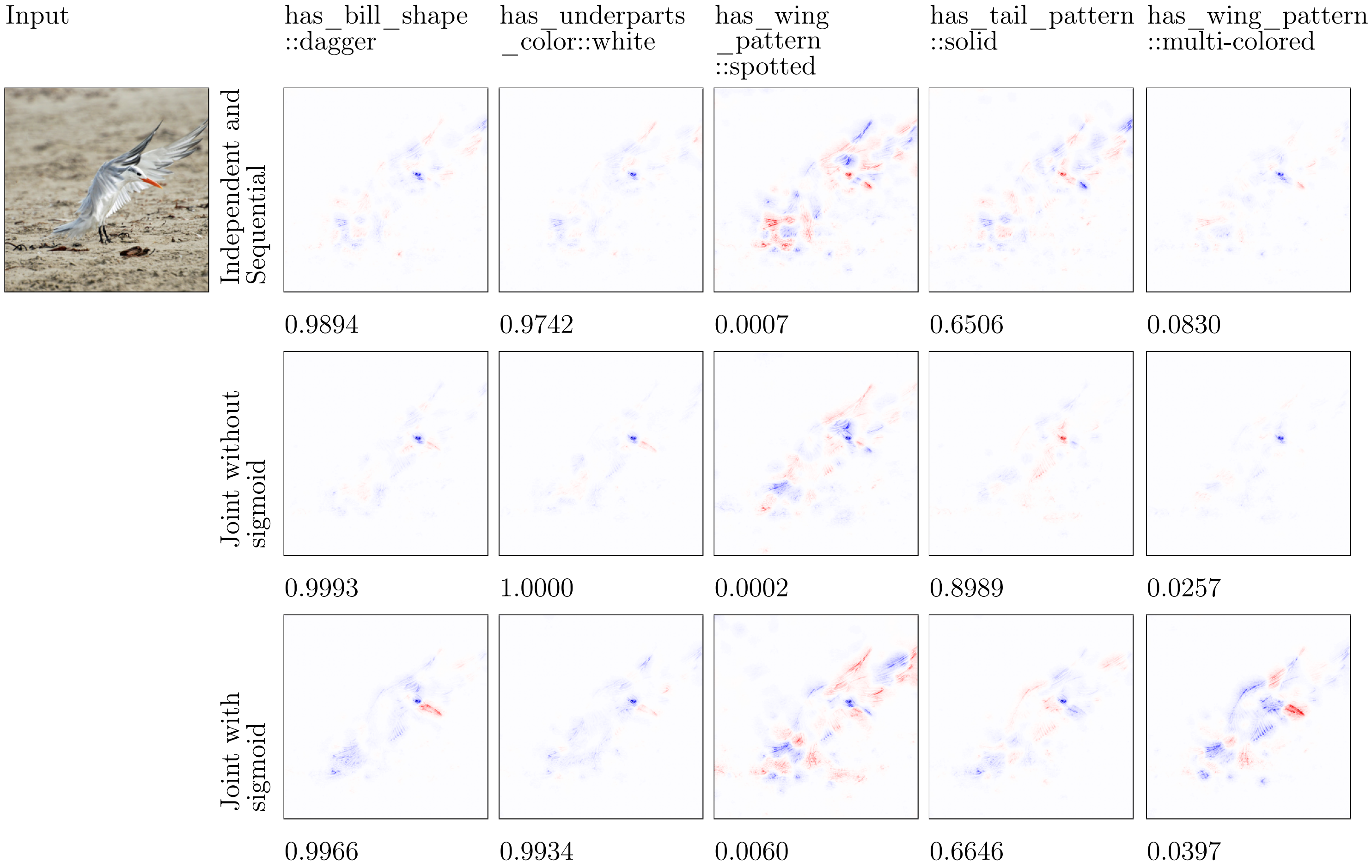


Figure A2: Concepts are a mixture of present, not present, correctly predicted and not correctly predicted. The input class was Elegant_tern.

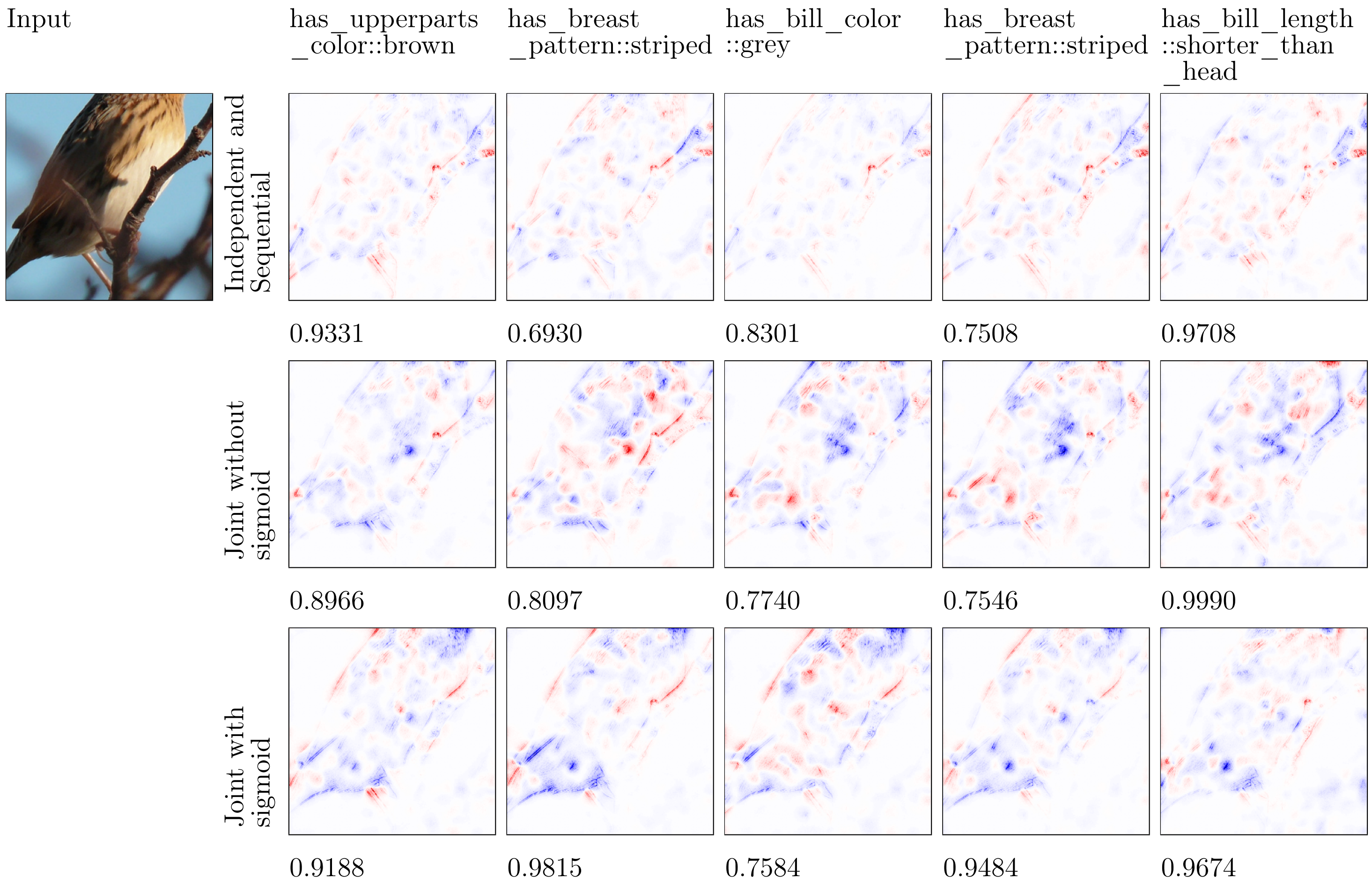


Figure A3: Concepts shown are either incorrectly predicted as present or correctly predicted as present but not visible in the input image. The input class was Le Conte Sparrow.

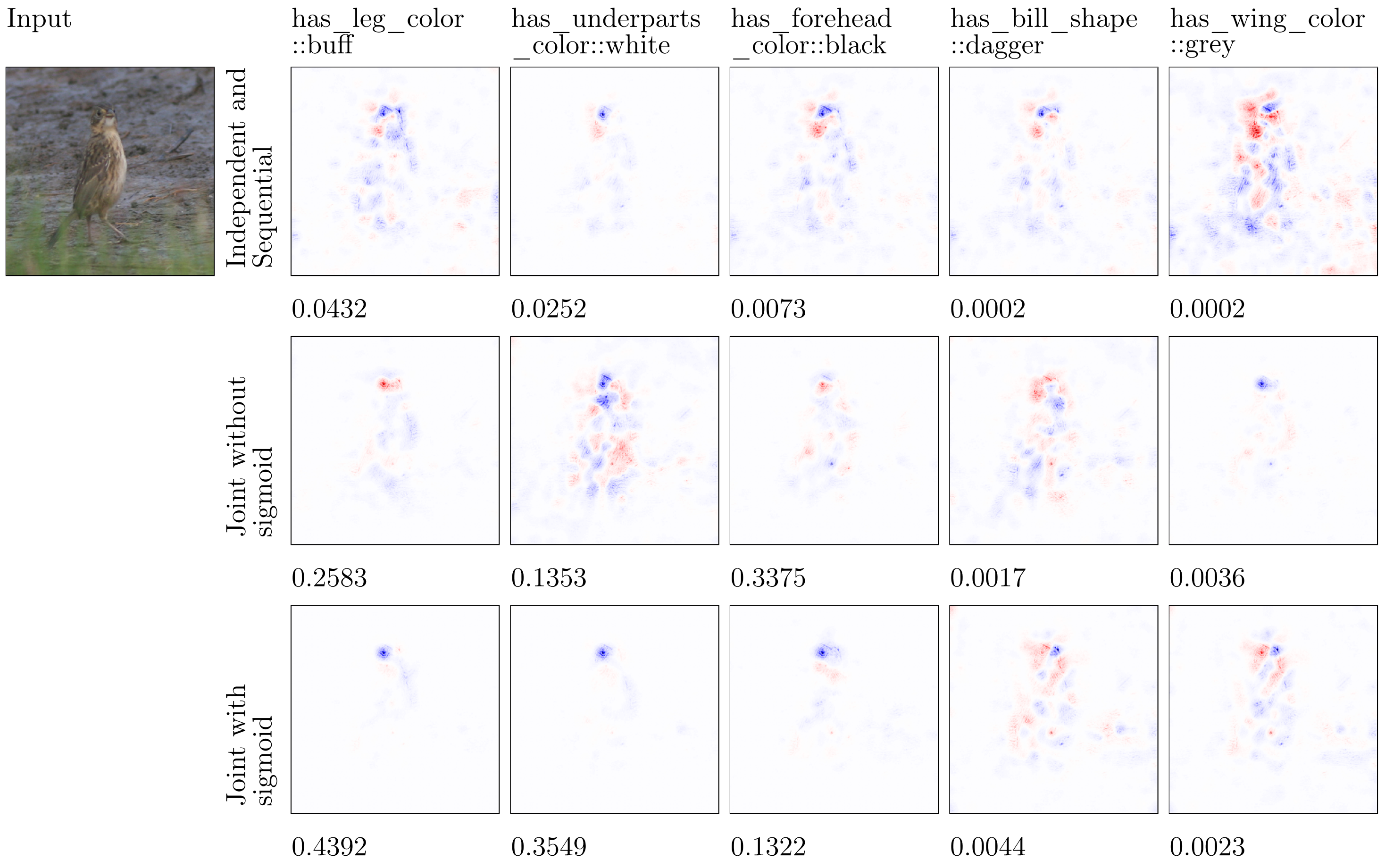
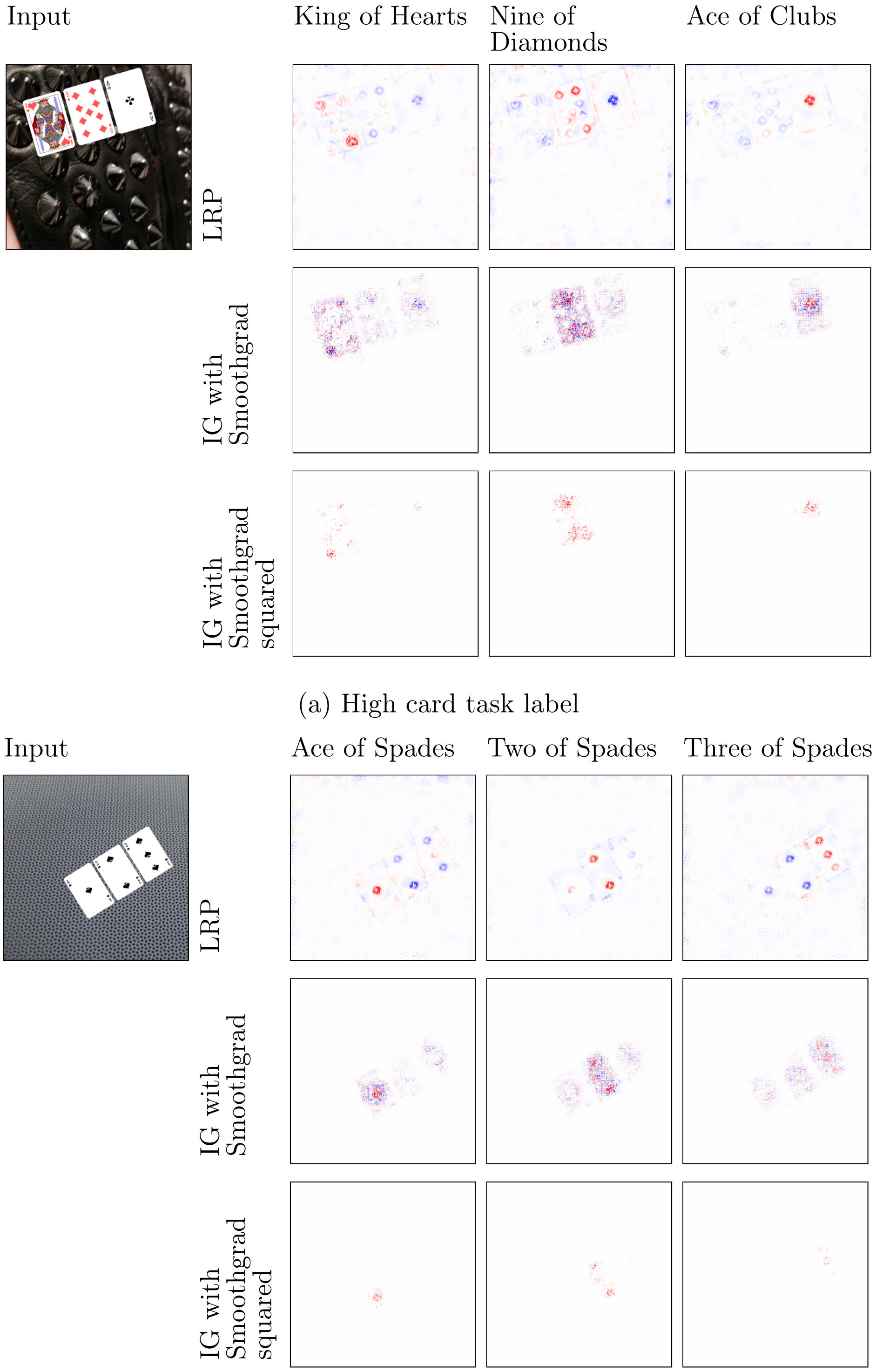


Figure A4: Concepts shown are either correctly or incorrectly predicted as not present. The input class was Nelson Sharp tailed Sparrow.

A.1 Playing Cards Input Feature Attribution

Here we show additional concept saliency map results for our instance-level playing card models in Figure A5, A6, A7, and A8. Most saliency maps focus on the playing card that are semantically meaningful to its corresponding concept. All saliency maps generated with LRP show positive relevancy attributed to the correct playing card for each concept to satisfy semantically meaningful feature mapping, while both IG variations show good localisation. IG with a smoothgrad squared noise tunnel applies little relevance in general and highlights a different card for the concept "King of Hearts" in Figure A7.

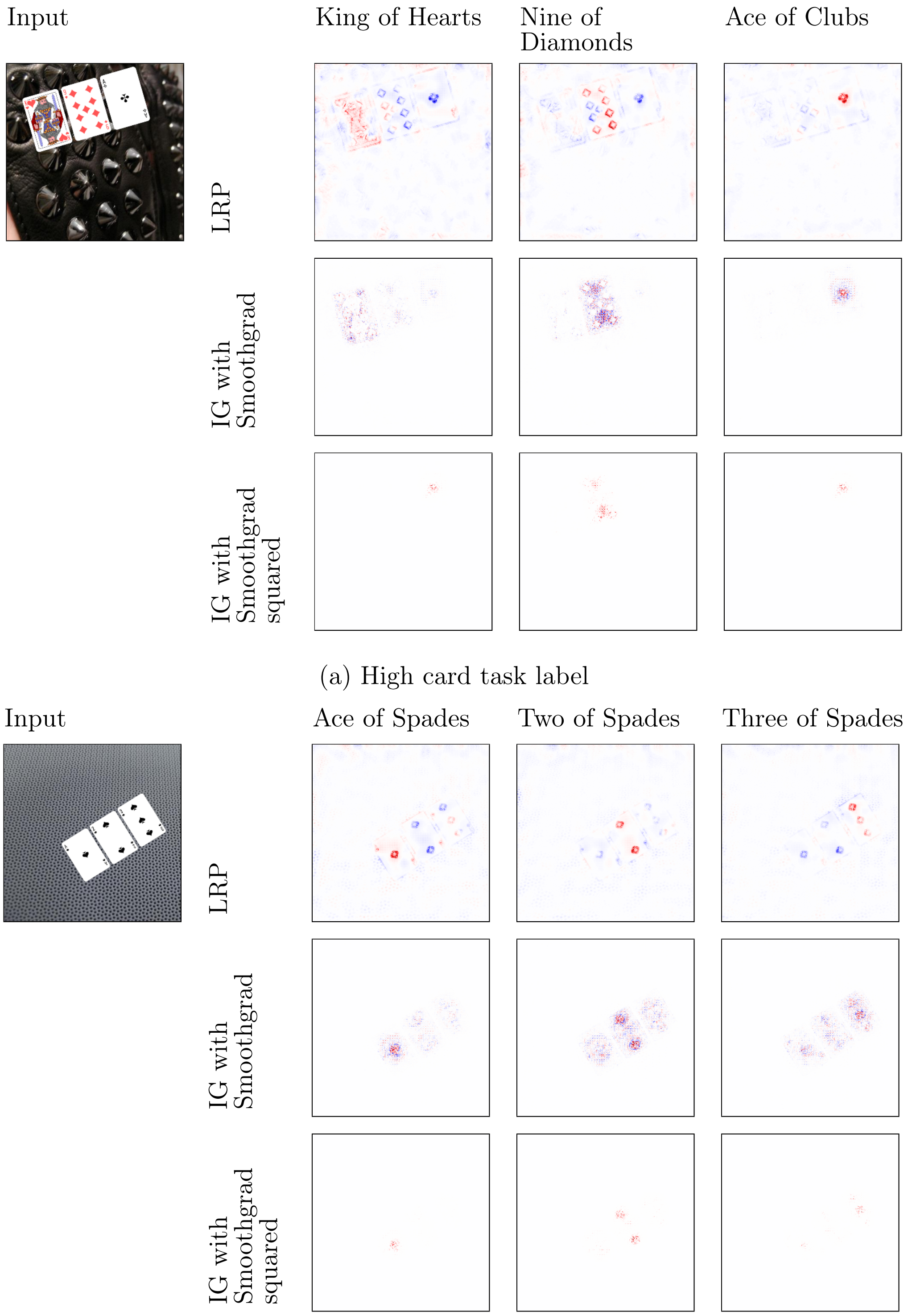
We provide examples of all task classes for class-level poker card saliency maps to show variations between all concept predictions. These can be seen in Figure A9, A10, A11, A12, A13 and A14. Most concepts do not apply relevance to the semantically meaningful input features. An exception is the concept "Four of Clubs" as seen in Figure A11.



(a) High card task label

(b) Straight flush task label

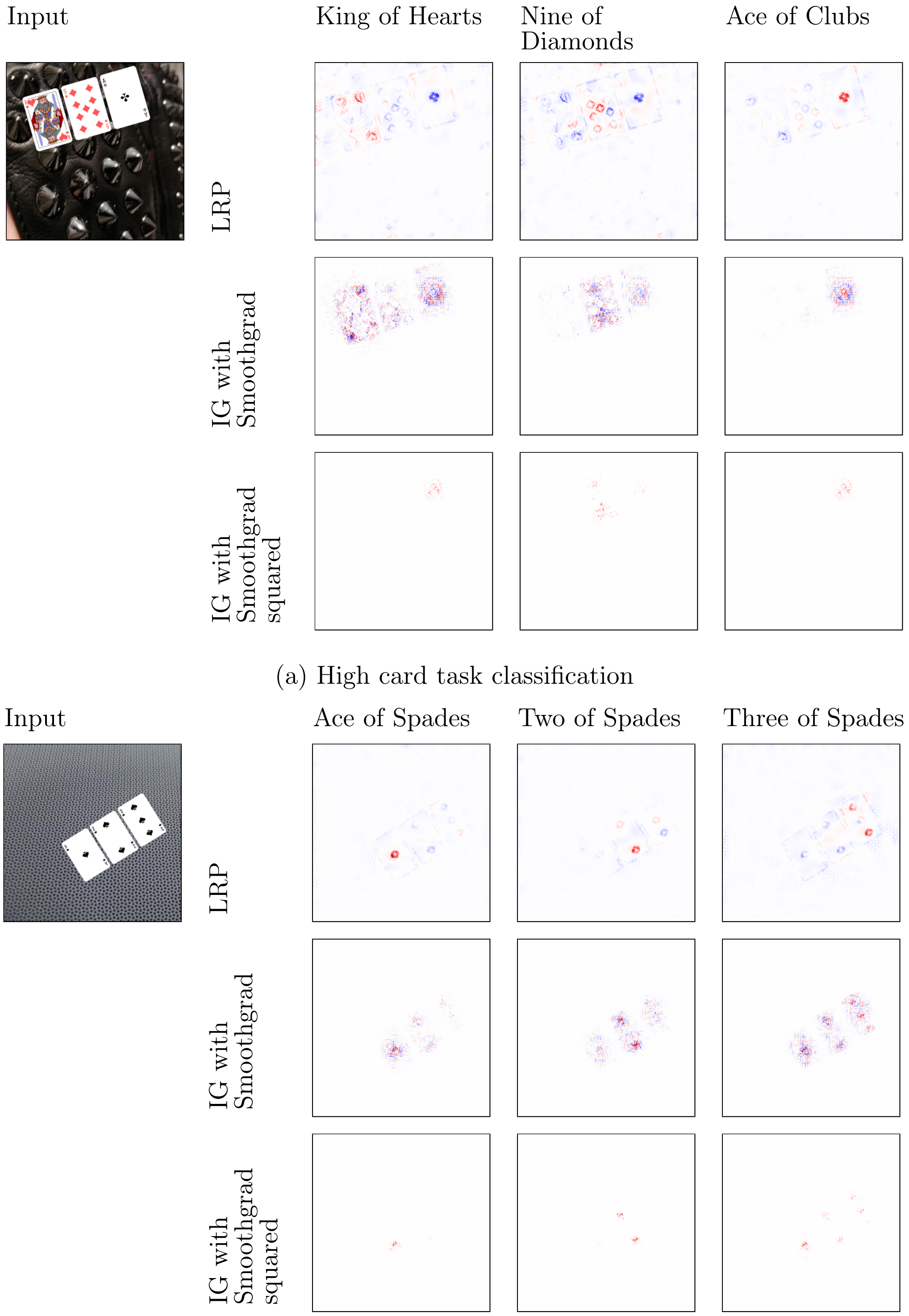
Figure A5: Independent / sequential model trained on Poker cards.



(a) High card task label

(b) Straight flush task label

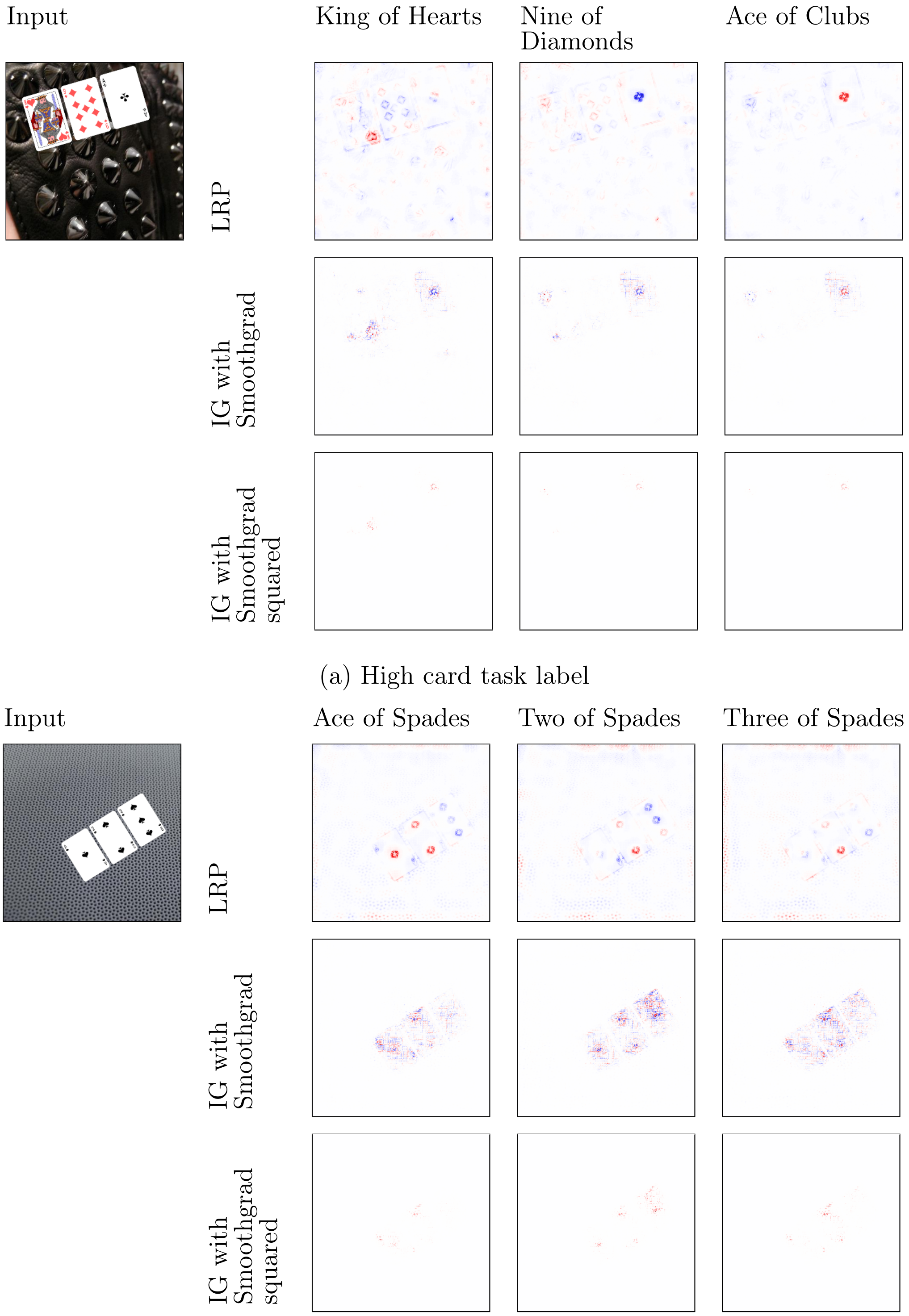
Figure A6: Independent / sequential model trained on Random cards.



(a) High card task classification

(b) Straight flush task classification

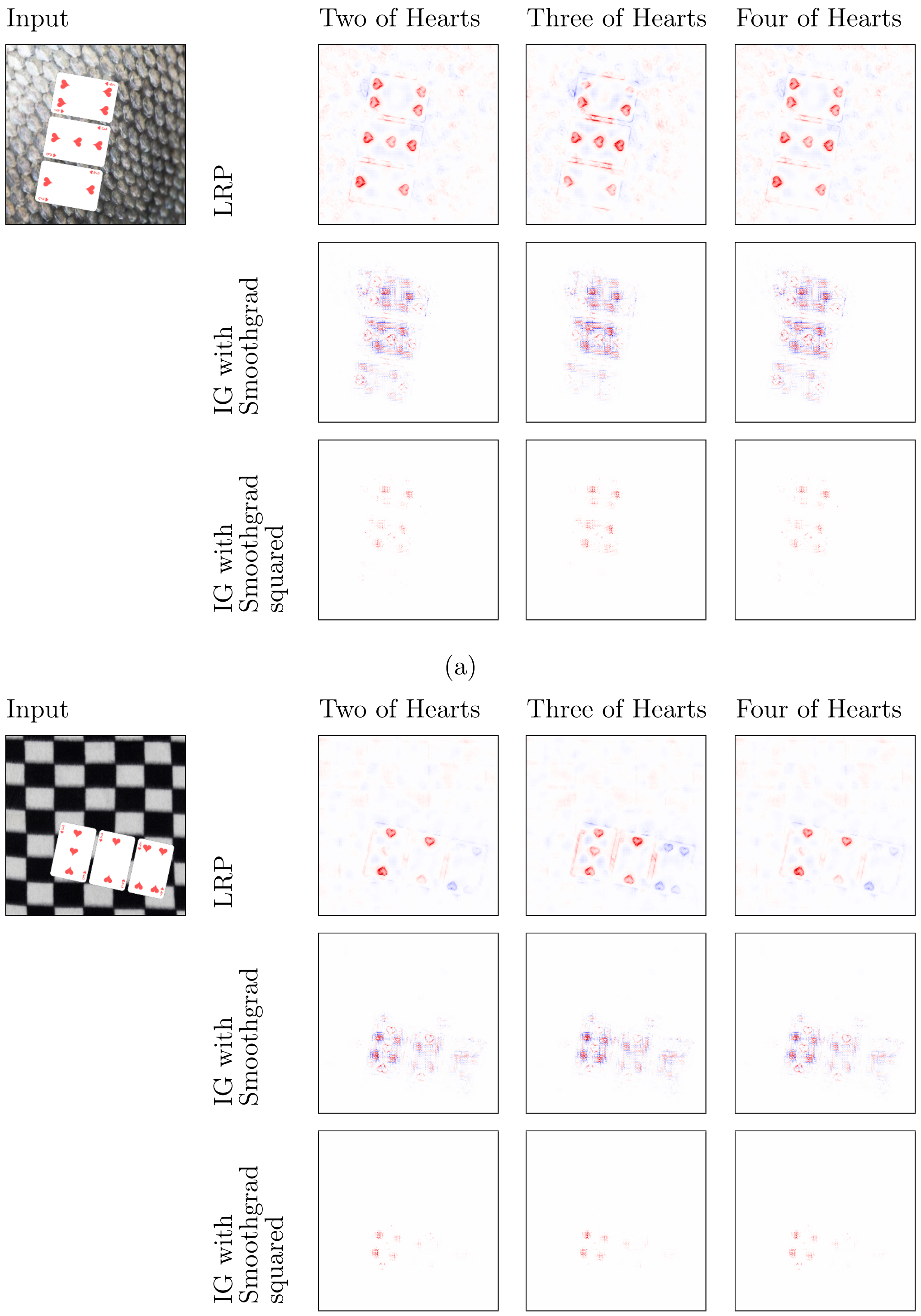
Figure A7: Joint model trained on Poker cards.



(a) High card task label

(b) Straight flush task label

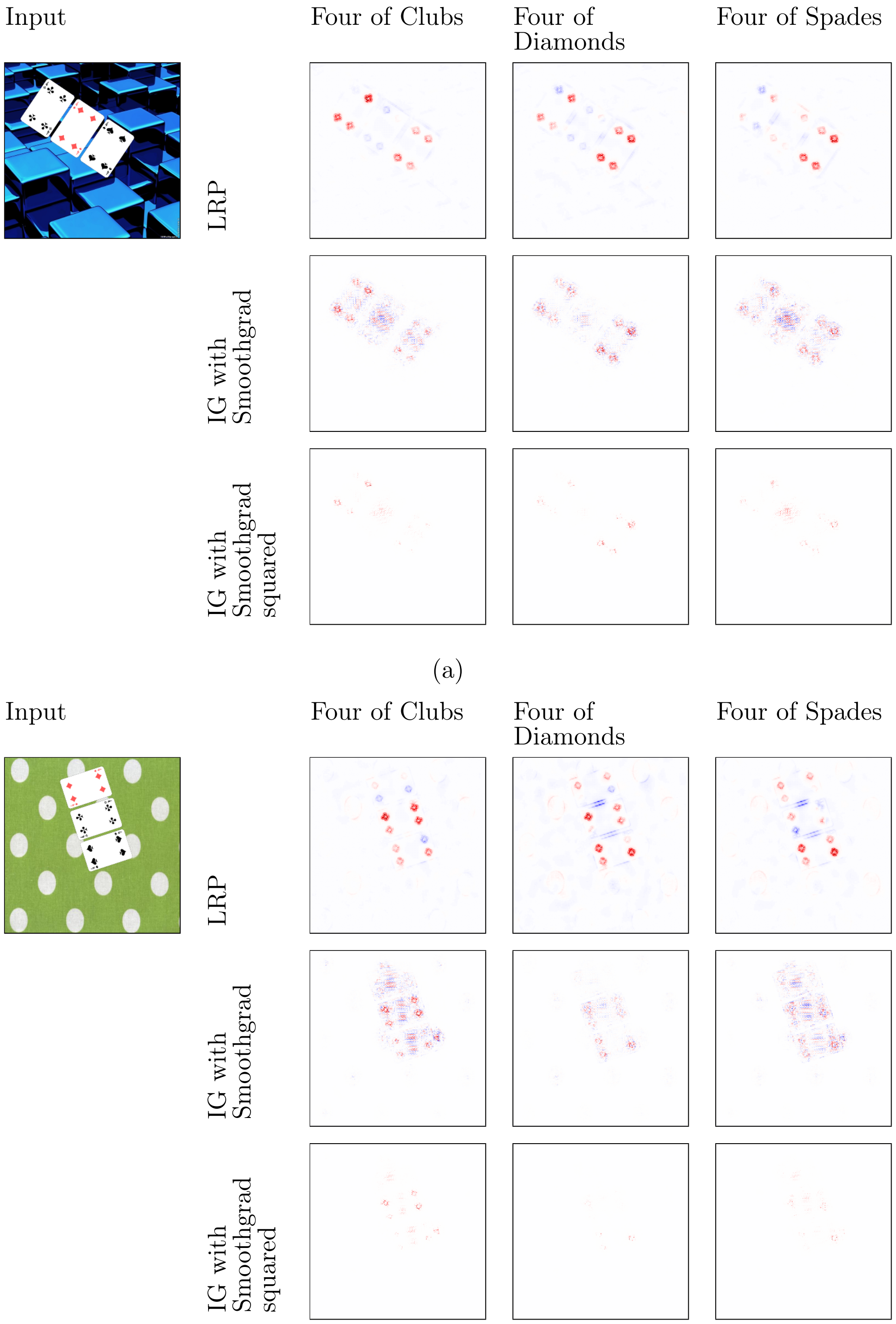
Figure A8: Standard DNN model trained on Poker cards.



(a)

(b)

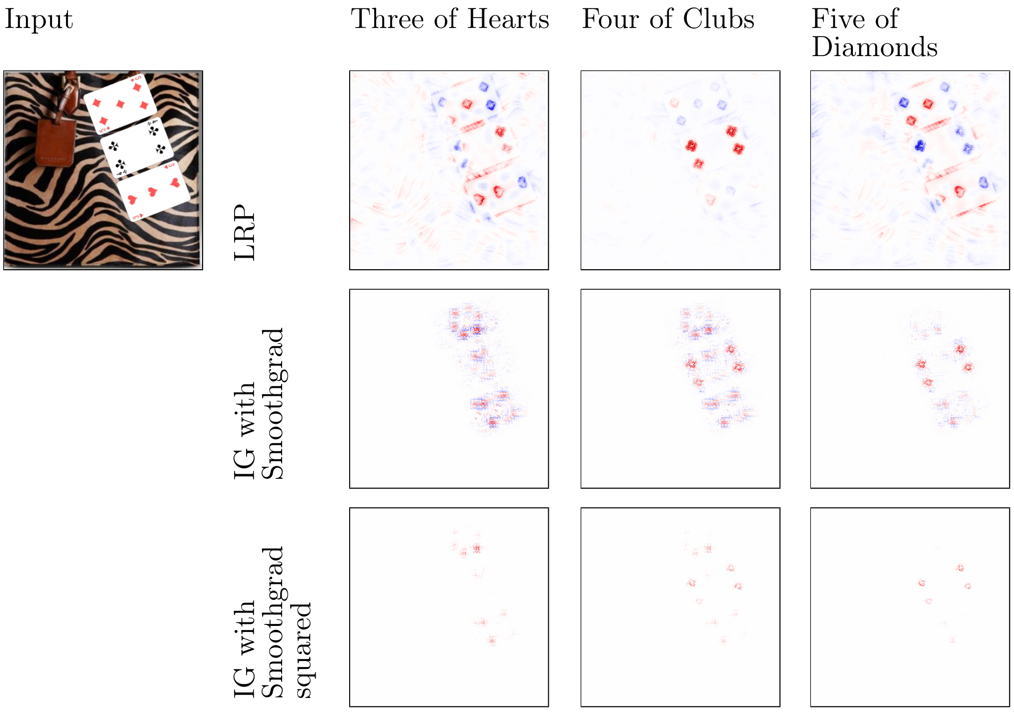
Figure A9: Class-level poker cards predicting concepts with the task label Straight flush.



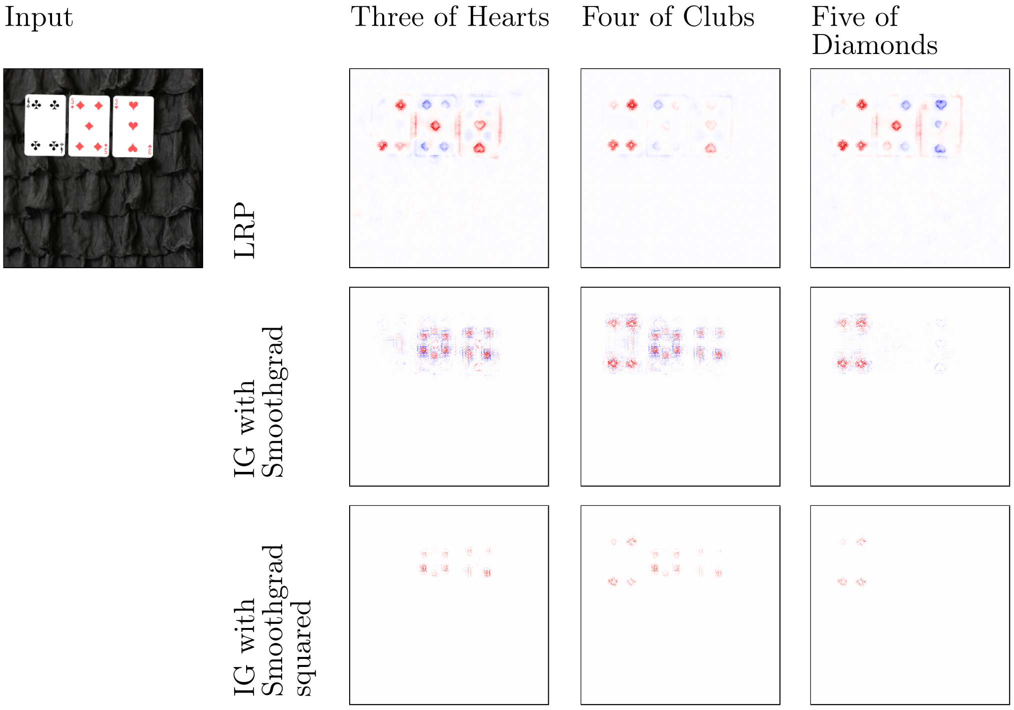
(a)

(b)

Figure A10: Class-level poker cards predicting concepts with the task classification of Three of a kind.

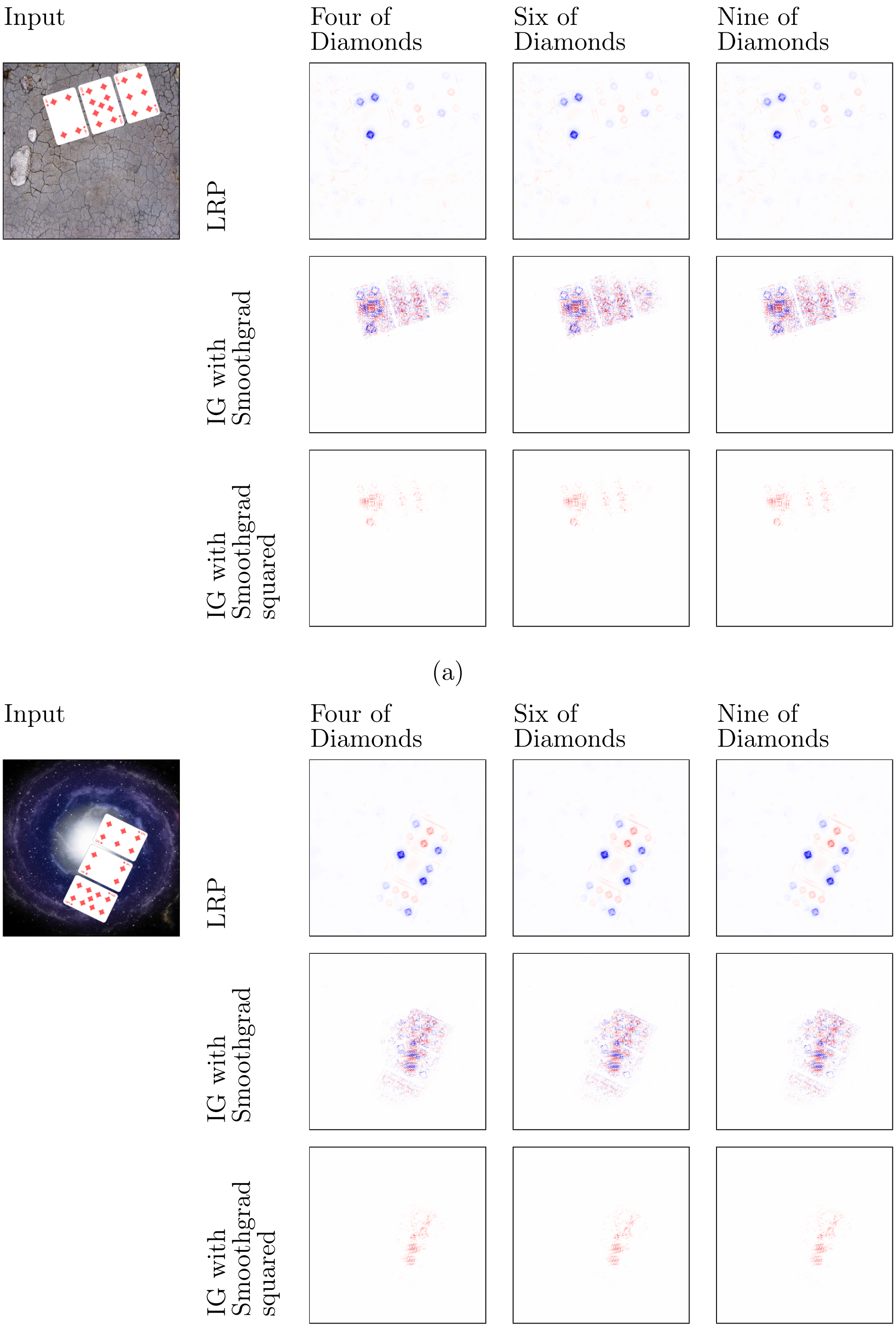


(a)



(b)

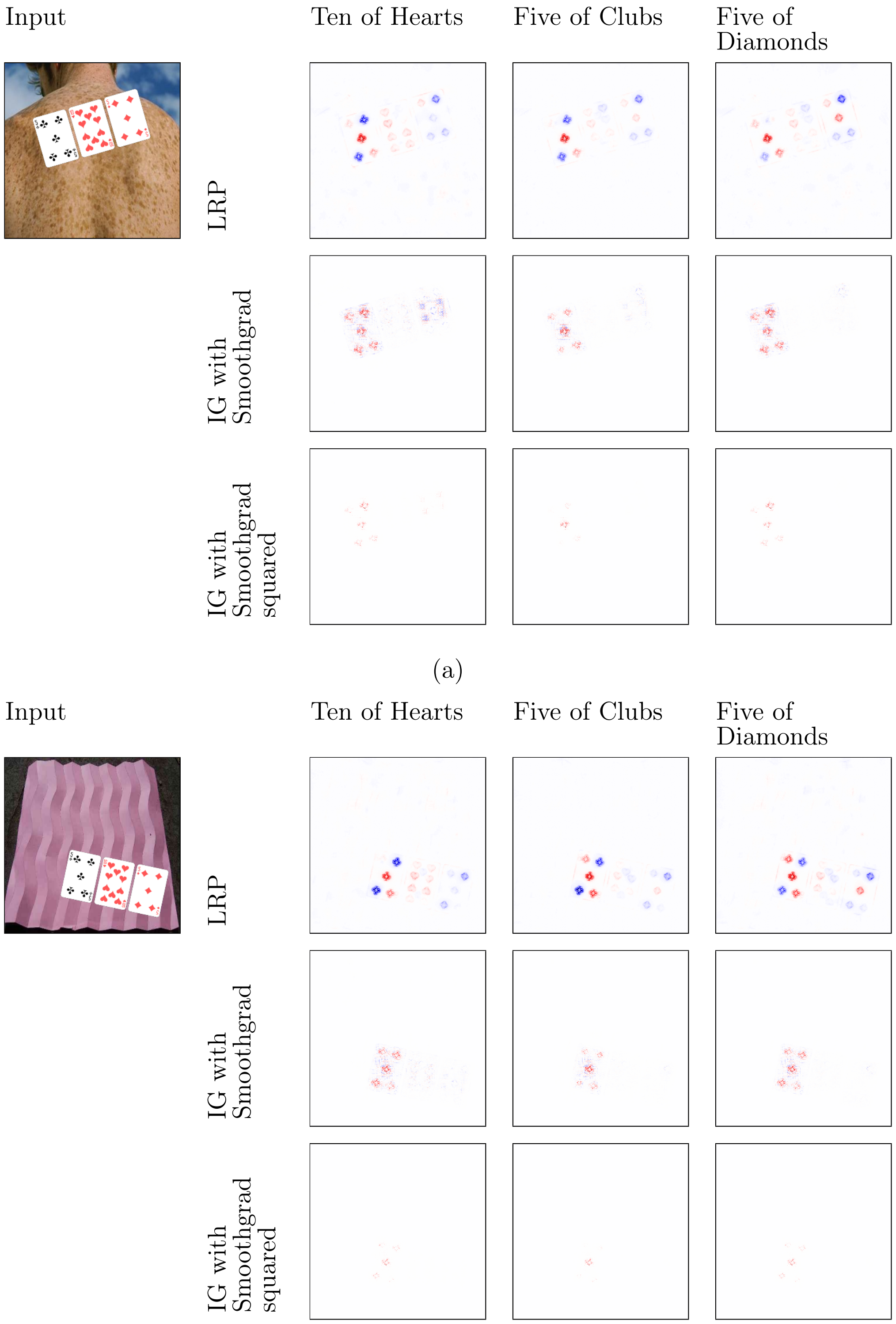
Figure A11: Class-level poker cards predicting concepts with the task label Straight.



(a)

(b)

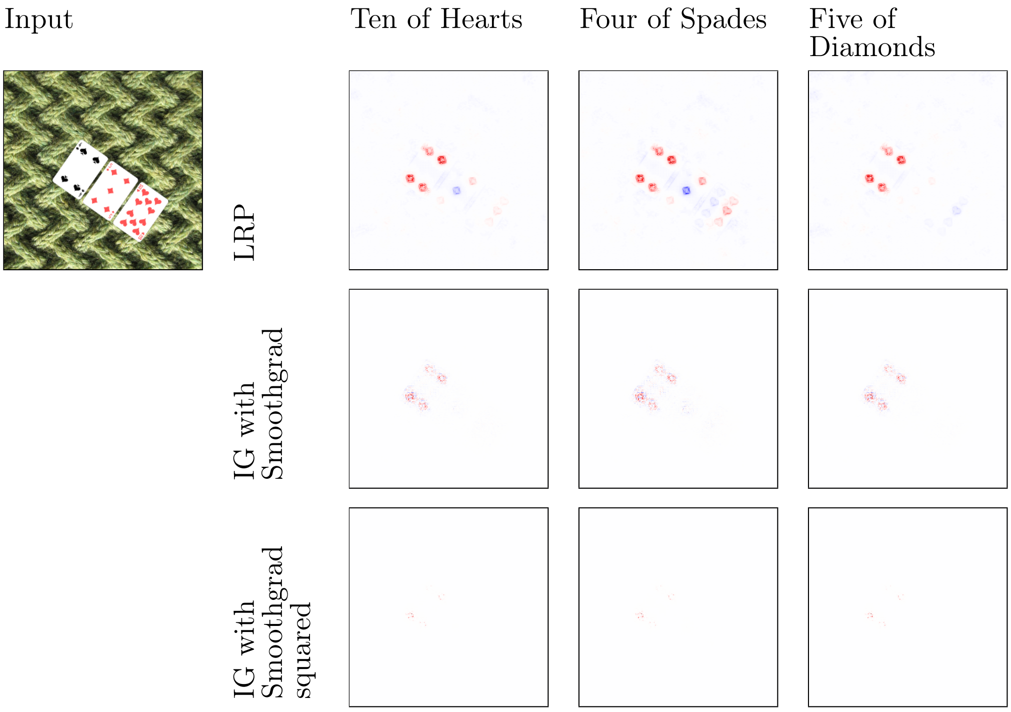
Figure A12: Class-level poker cards predicting concepts with the task label Flush.



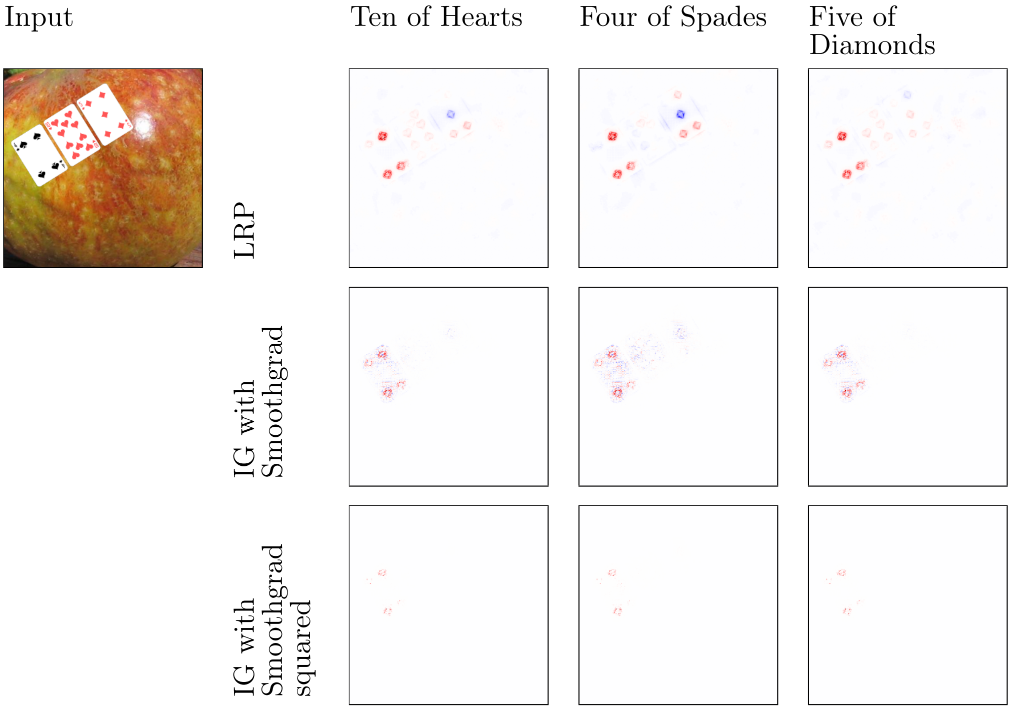
(a)

(b)

Figure A13: Class-level poker cards predicting concepts with the task label Pair.



(a)



(b)

Figure A14: Class-level poker cards predicting concepts with the task label High card.

A.2 CheXpert Input Feature Attribution

Here we show additional saliency map results for our CheXpert models to visualise feature attribution from concept predictions propagated to input features. Most saliency maps for models trained on instance-level CheXpert apply feature attribution values to input features within ground truth segmentations regions. Models trained on Class-level CheXpert apply feature attribution values to the same input features irrespective of which concept prediction is being analysed.

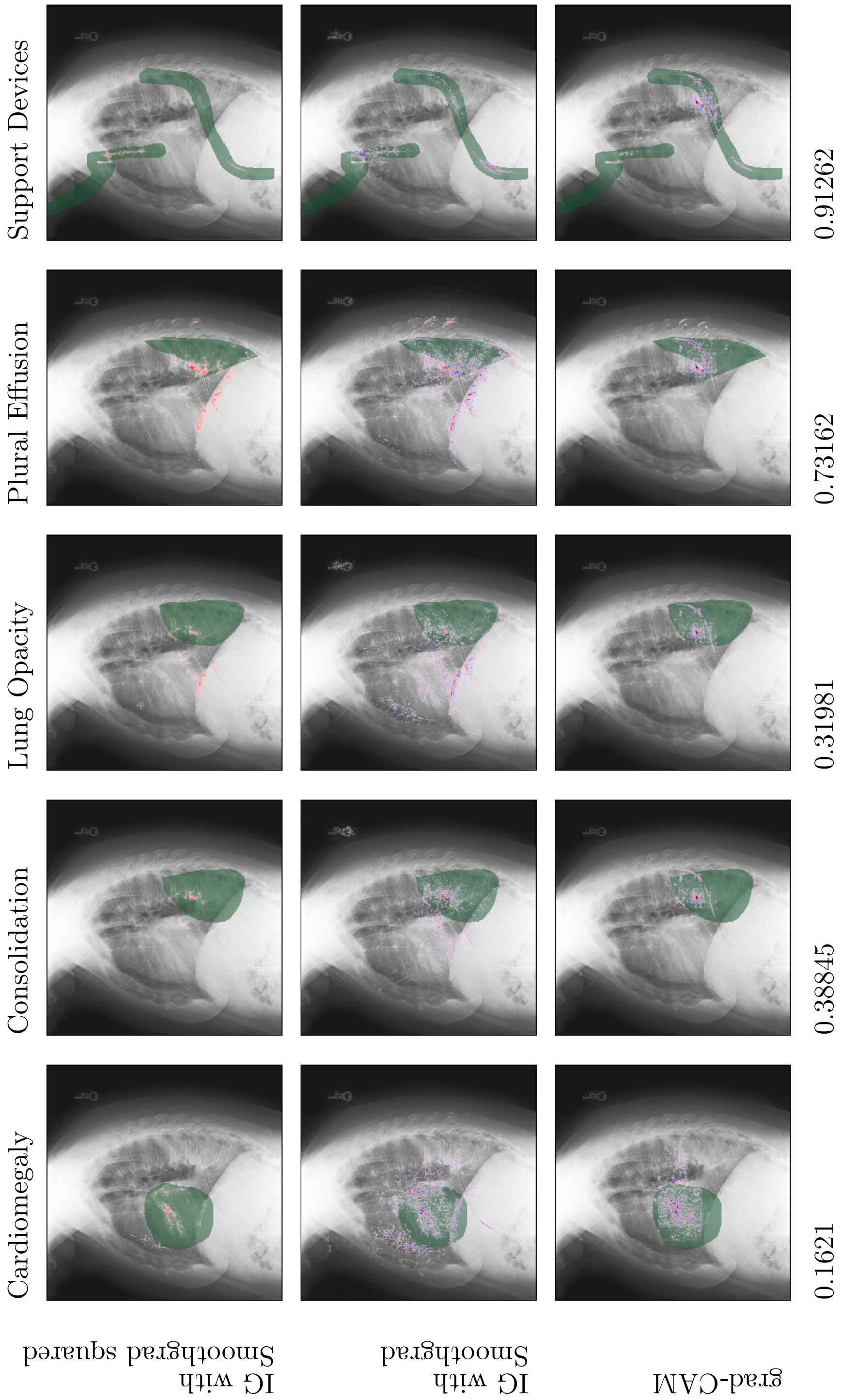


Figure A15: Saliency maps from a instance-level CheXpert trained using the joint CBM method. The number beneath the saliency map is the concept prediction which is in the range of 0 and 1.

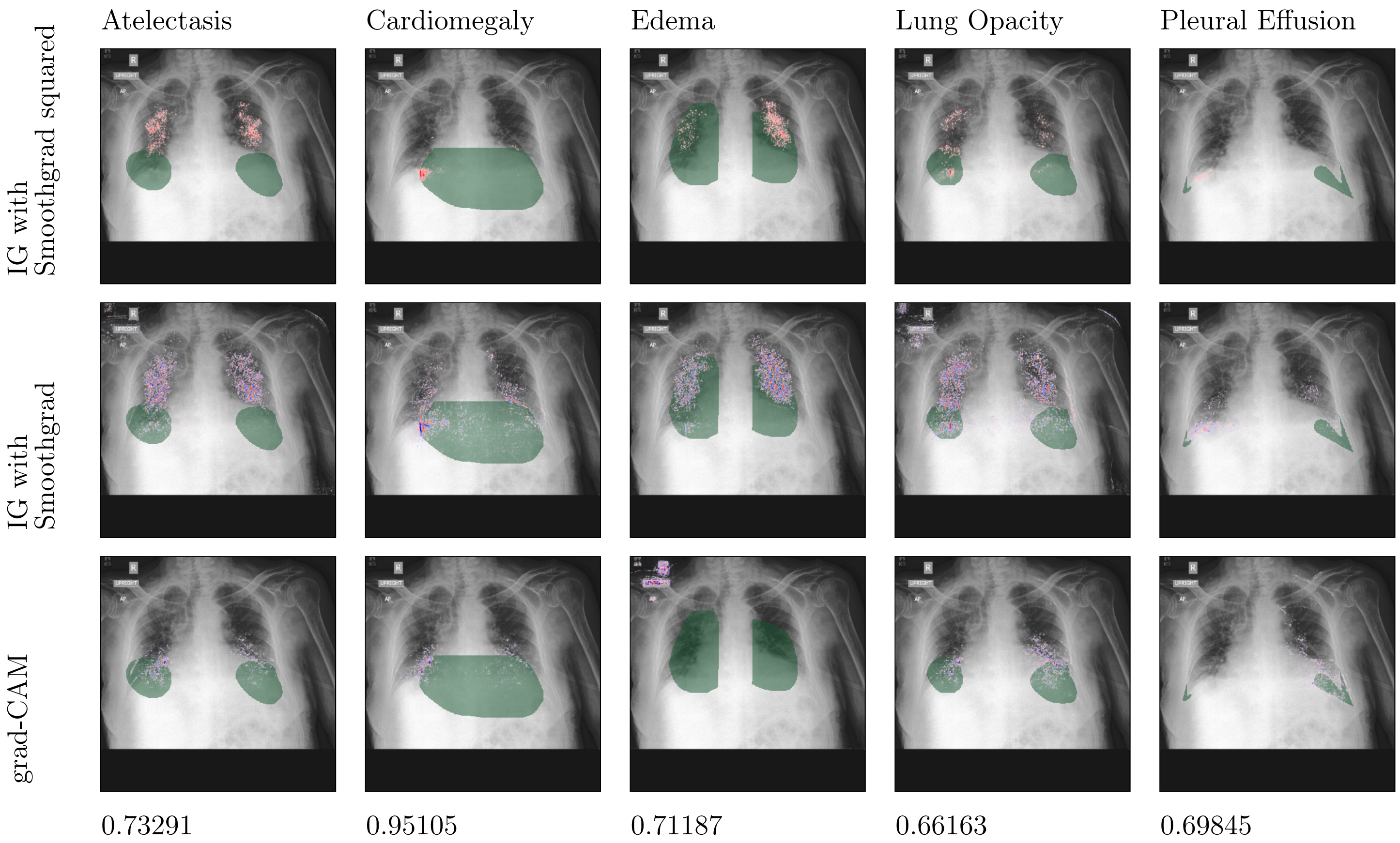


Figure A16: Saliency maps from a instance-level CheXpert trained using the joint CBM method. The number beneath the saliency map is the concept prediction which is in the range of 0 and 1.

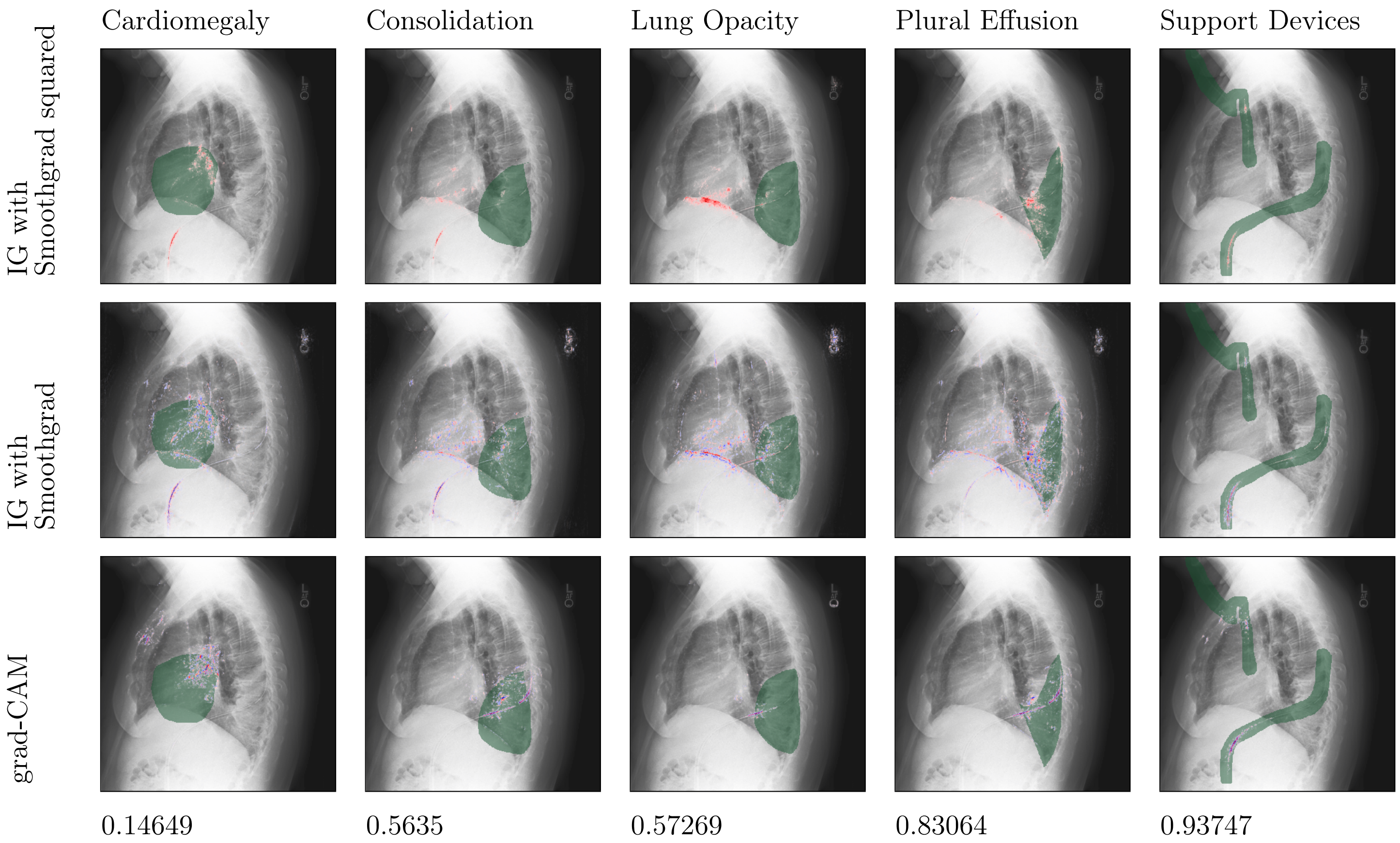


Figure A17: Saliency maps from a instance-level CheXpert trained using the sequential CBM method.

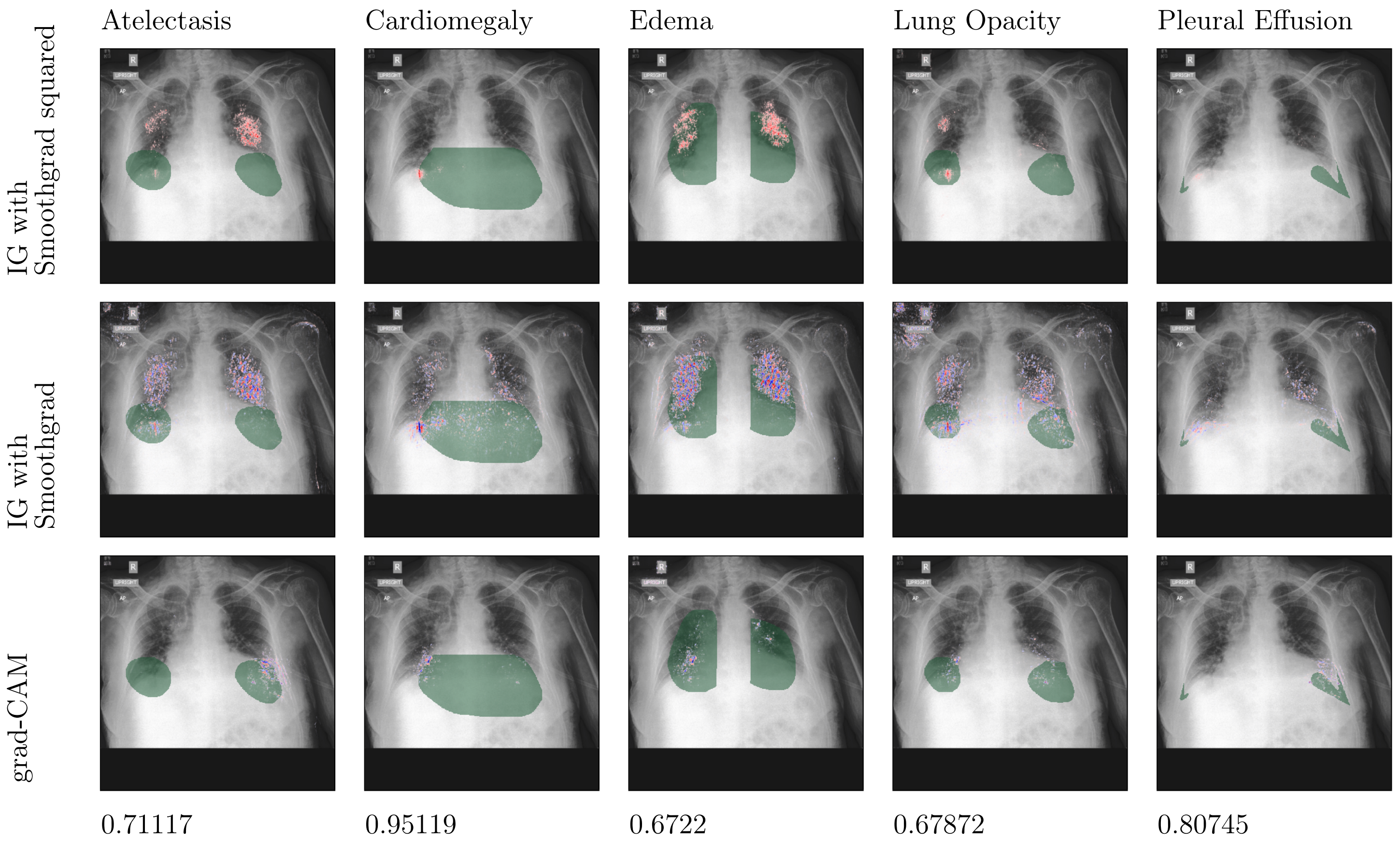


Figure A18: Saliency maps from a instance-level CheXpert trained using the sequential CBM method. The number beneath the saliency map is the concept prediction which is in the range of 0 and 1.

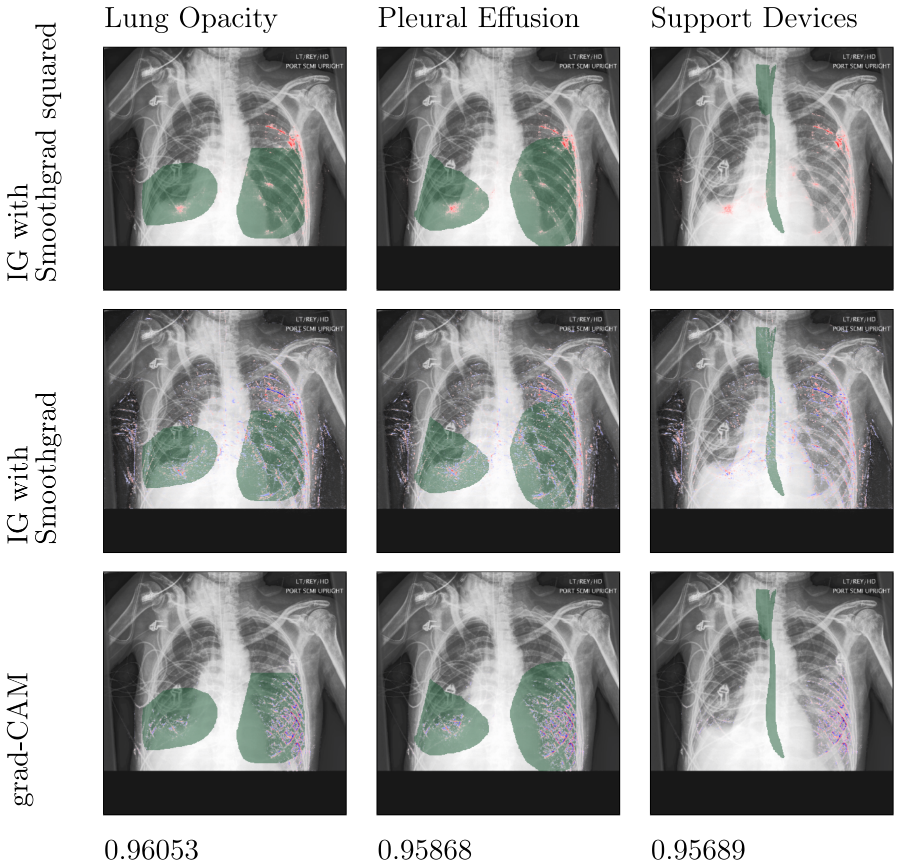


Figure A19: Saliency maps from a class-level CheXpert with three concept present, trained using the sequential CBM method. The number beneath the saliency map is the concept prediction which is in the range of 0 and 1.

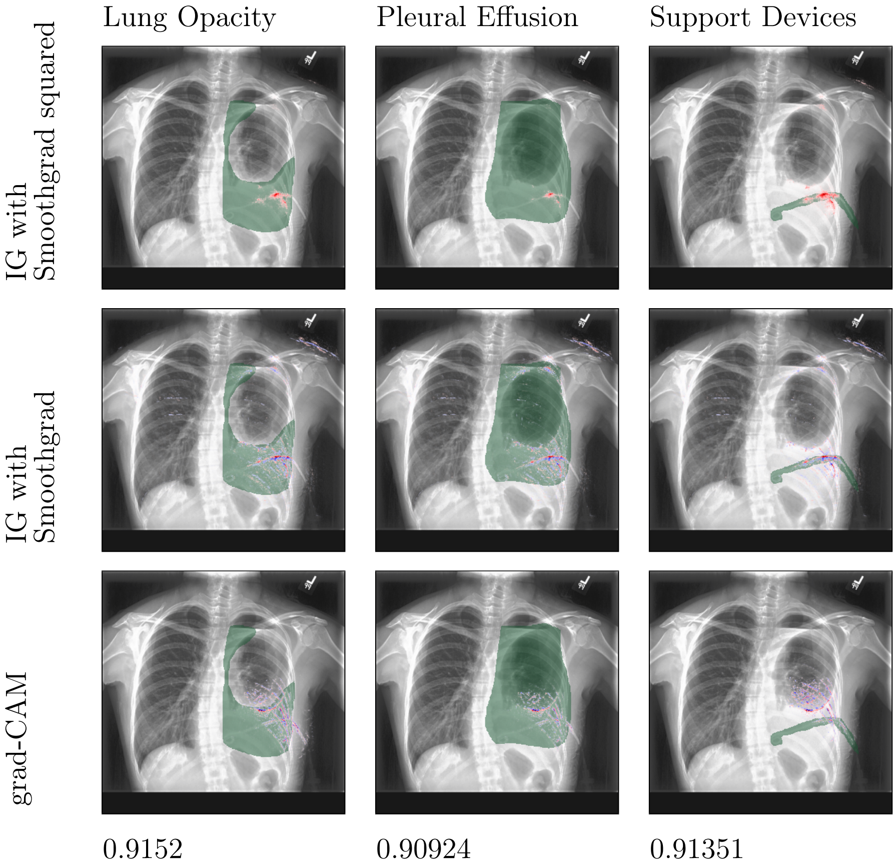


Figure A20: Saliency maps from a class-level CheXpert with three concept present, trained using the sequential CBM method. The number beneath the saliency map is the concept prediction which is in the range of 0 and 1.

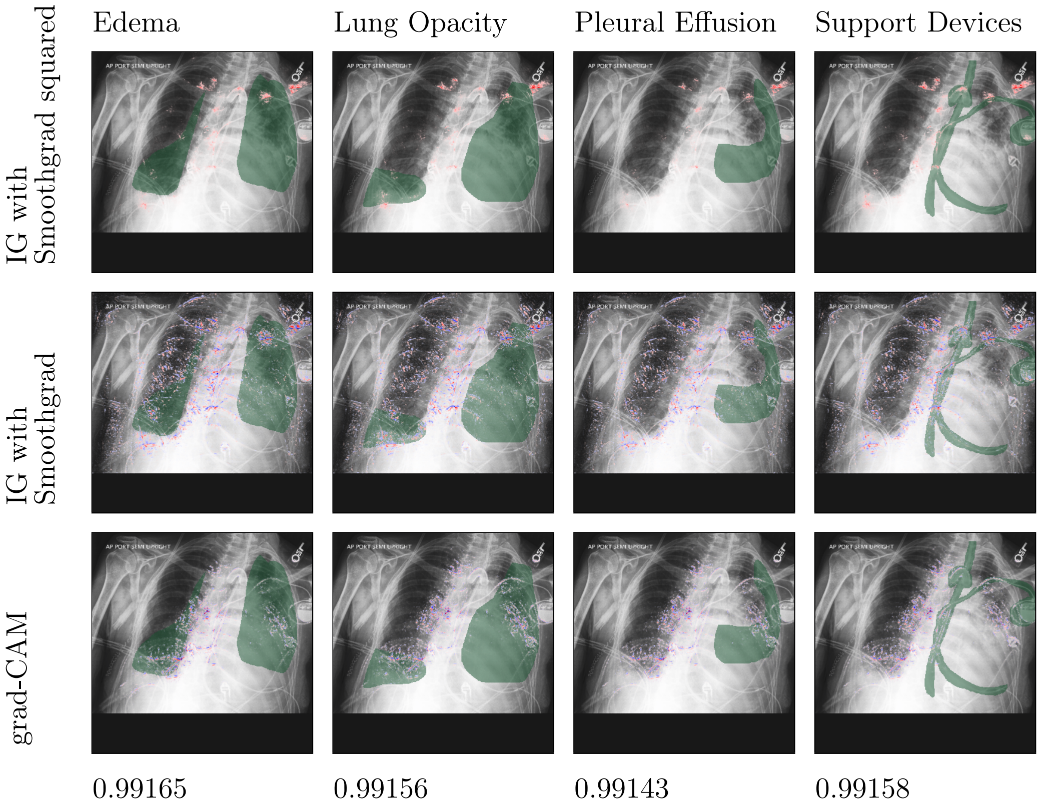


Figure A21: Saliency maps from a class-level CheXpert with four concept present, trained using the sequential CBM method. The number beneath the saliency map is the concept prediction which is in the range of 0 and 1.

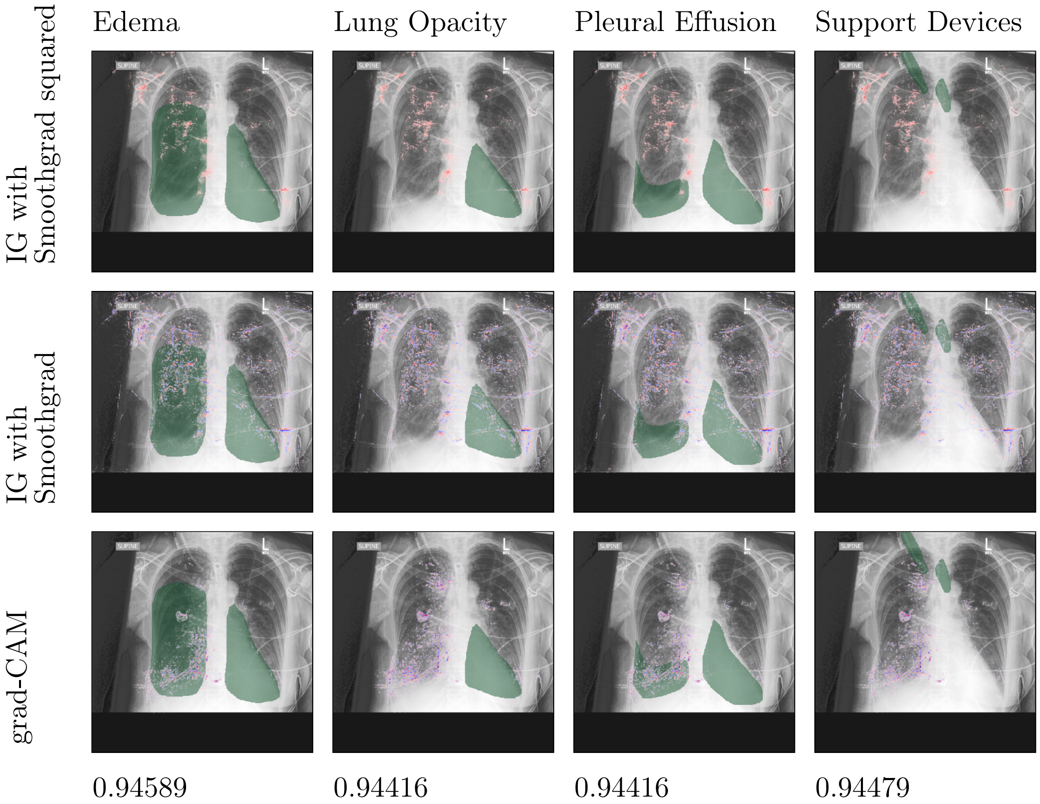


Figure A22: Saliency maps from a class-level CheXpert with four concept present, trained using the sequential CBM method. The number beneath the saliency map is the concept prediction which is in the range of 0 and 1.

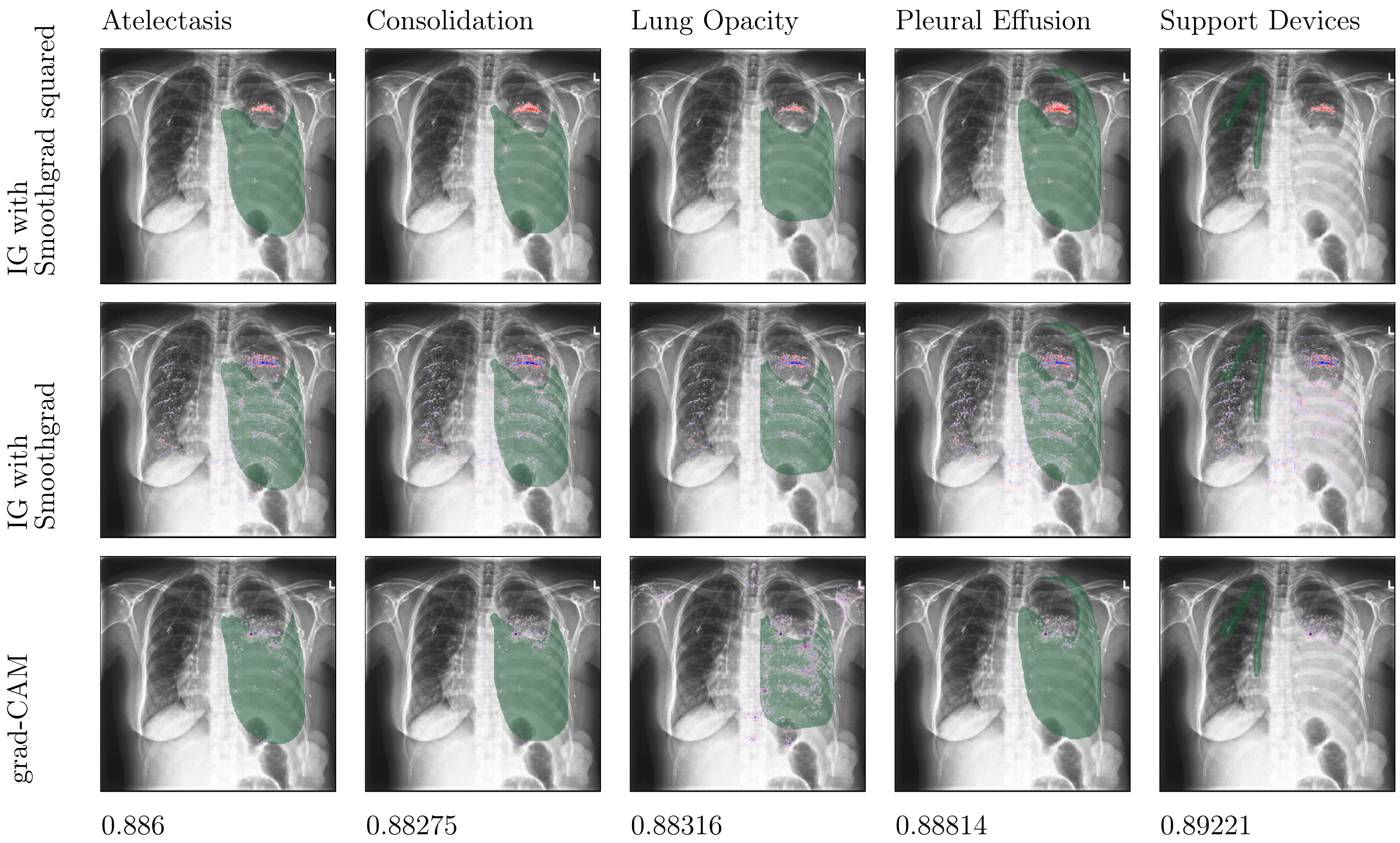


Figure A23: Saliency maps from a class-level CheXpert with five concept present, trained using the sequential CBM method. The number beneath the saliency map is the concept prediction which is in the range of 0 and 1.

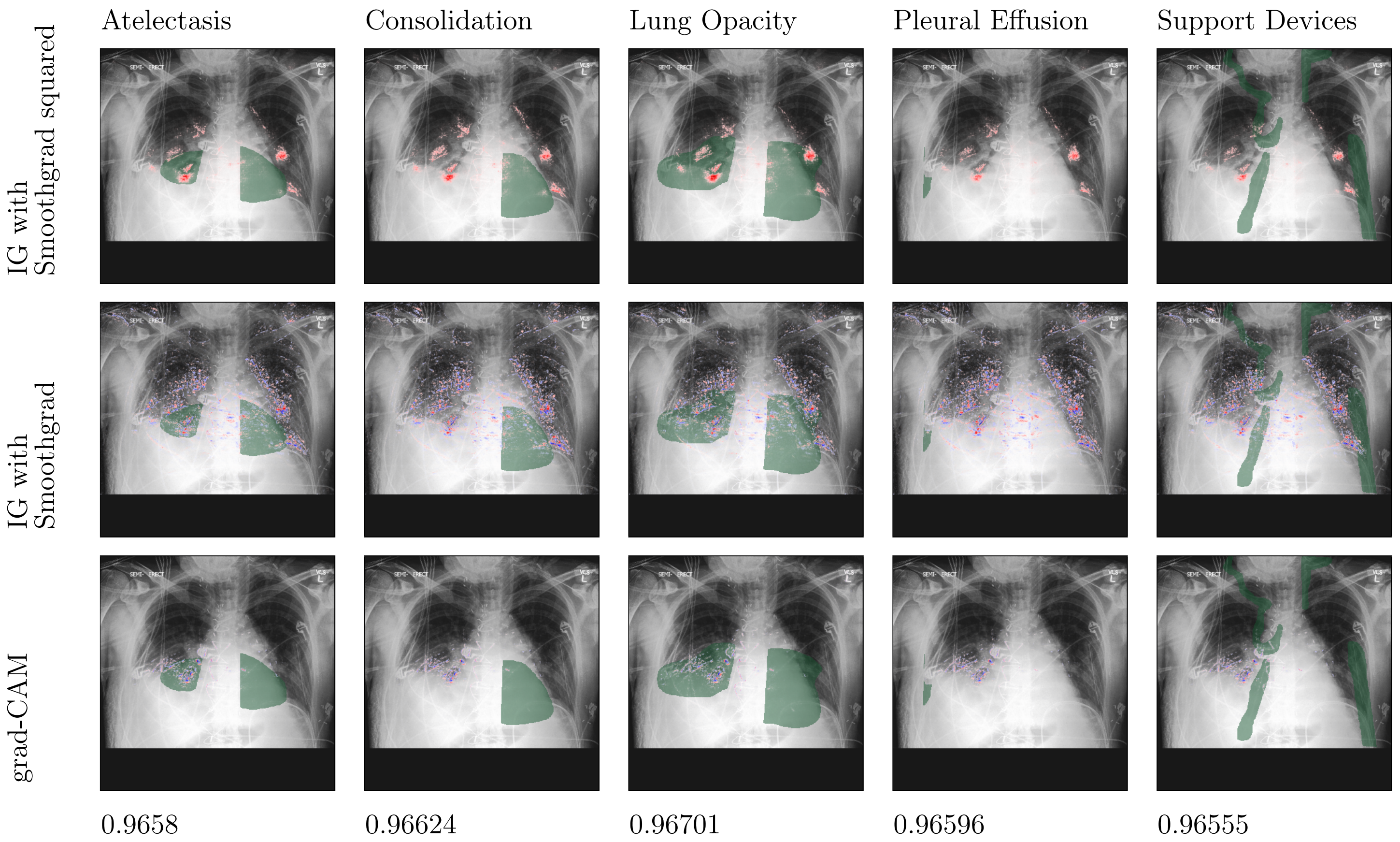


Figure A24: Saliency maps from a class-level CheXpert with five concept present, trained using the sequential CBM method. The number beneath the saliency map is the concept prediction which is in the range of 0 and 1.

A.3 CUB Concept Feature Attribution Proportions

Feature attribution values propagated from the task label prediction to the concept vector can also be visualised with saliency maps. We include saliency maps results in Section 3.6. As feature attribution is conserved, we also computed the contribution each concept prediction made to the predicted task label. We include a complete breakdown of concept contribution for the saliency maps in Figure 3.19 in Table A1, Table A2, Table A3 and Table A4.

The table of concept contributions contains each concept ID, concept vector predictions, LRP feature attribution value, and concept contributions. The tables are sorted to display the concept with the highest contribution to the predicted task label first, followed by concepts in descending order of contribution. The concept vectors for each model are split into 112 segments, one for each concept in the dataset, with the segment most to the left of each saliency map in Figure 3.19 representing the concept with ID 0 and the segment most to the right representing the concept with ID 111.

Concept ID	Concept name	Concept value	Feature attribution value	Concept contribution
18	has_underparts_color::yellow	1.0	0.0728	6.933%
110	has_wing_pattern::striped	1.0	0.0696	6.623%
88	has_tail_pattern::multi-colored	1.0	0.0658	6.27%
99	has_bill_color::grey	1.0	0.0628	5.977%
72	has_belly_color::yellow	1.0	0.0615	5.855%
47	has_throat_color::black	1.0	0.0525	4.995%
31	has_tail_shape::notched_tail	1.0	0.0516	4.911%
96	has_leg_color::grey	1.0	0.0498	4.745%
61	has_under_tail_color::black	1.0	0.0476	4.533%
8	has_wing_color::white	1.0	0.047	4.476%
38	has_head_pattern::plain	1.0	0.0469	4.463%
106	has_crown_color::black	1.0	0.043	4.095%

67	has_nape_color::black	1.0	0.0424	4.04%
93	has_primary_color::black	1.0	0.0423	4.031%
7	has_wing_color::black	1.0	0.0416	3.961%
28	has_back_color::black	1.0	0.0412	3.925%
57	has_forehead_color::black	1.0	0.0374	3.56%
22	has_breast_pattern::solid	1.0	0.0274	2.612%
13	has_upperparts_color::black	1.0	0.024	2.286%
52	has_bill_length::shorter_than_head	1.0	0.023	2.187%
89	has_belly_pattern::solid	1.0	0.0198	1.881%
82	has_shape::perching-like	1.0	0.0128	1.218%
78	has_size::small_(5_-_9_in)	1.0	0.0094	0.897%
81	has_shape::duck-like	0.0	0.0	0.837%
54	has_forehead_color::brown	0.0	0.0	0.77%
50	has_eye_color::black	1.0	0.0079	0.751%
27	has_back_color::yellow	0.0	0.0	0.509%
24	has_breast_pattern::multi-colored	0.0	0.0	0.502%
64	has_nape_color::brown	0.0	0.0	0.339%
102	has_crown_color::blue	0.0	0.0	0.281%
59	has_under_tail_color::brown	0.0	0.0	0.272%
66	has_nape_color::yellow	0.0	0.0	0.221%
95	has_primary_color::buff	0.0	0.0	0.19%
30	has_back_color::buff	0.0	0.0	0.189%
44	has_breast_color::buff	0.0	0.0	0.187%
85	has_back_pattern::multi-colored	0.0	0.0	0.18%
29	has_back_color::white	0.0	0.0	0.154%
9	has_wing_color::buff	0.0	0.0	0.142%
0	has_bill_shape::dagger	0.0	0.0	0.0%
1	has_bill_shape::hooked_seabird	0.0	0.0	0.0%
2	has_bill_shape::all-purpose	0.0	0.0	0.0%
3	has_bill_shape::cone	0.0	0.0	0.0%
4	has_wing_color::brown	0.0	0.0	0.0%
5	has_wing_color::grey	0.0	0.0	0.0%
6	has_wing_color::yellow	0.0	0.0	0.0%
10	has_upperparts_color::brown	0.0	0.0	0.0%
11	has_upperparts_color::grey	0.0	0.0	0.0%
12	has_upperparts_color::yellow	0.0	0.0	0.0%

14	has_upperparts_color::white	0.0	0.0	0.0%
15	has_upperparts_color::buff	0.0	0.0	0.0%
16	has_underparts_color::brown	0.0	0.0	0.0%
17	has_underparts_color::grey	0.0	0.0	0.0%
19	has_underparts_color::black	0.0	0.0	0.0%
20	has_underparts_color::white	0.0	0.0	0.0%
21	has_underparts_color::buff	0.0	0.0	0.0%
23	has_breast_pattern::striped	0.0	0.0	0.0%
25	has_back_color::brown	0.0	0.0	0.0%
26	has_back_color::grey	0.0	0.0	0.0%
32	has_upper_tail_color::brown	0.0	0.0	0.0%
33	has_upper_tail_color::grey	0.0	0.0	0.0%
34	has_upper_tail_color::black	0.0	0.0	0.0%
35	has_upper_tail_color::white	0.0	0.0	0.0%
36	has_upper_tail_color::buff	0.0	0.0	0.0%
37	has_head_pattern::eyebrow	0.0	0.0	0.0%
39	has_breast_color::brown	0.0	0.0	0.0%
40	has_breast_color::grey	0.0	0.0	0.0%
41	has_breast_color::yellow	0.0	0.0	0.0%
42	has_breast_color::black	0.0	0.0	0.0%
43	has_breast_color::white	0.0	0.0	0.0%
45	has_throat_color::grey	0.0	0.0	0.0%
46	has_throat_color::yellow	0.0	0.0	0.0%
48	has_throat_color::white	0.0	0.0	0.0%
49	has_throat_color::buff	0.0	0.0	0.0%
51	has_bill_length::about_the_same_as_head	0.0	0.0	0.0%
53	has_forehead_color::blue	0.0	0.0	0.0%
55	has_forehead_color::grey	0.0	0.0	0.0%
56	has_forehead_color::yellow	0.0	0.0	0.0%
58	has_forehead_color::white	0.0	0.0	0.0%
60	has_under_tail_color::grey	0.0	0.0	0.0%
62	has_under_tail_color::white	0.0	0.0	0.0%
63	has_under_tail_color::buff	0.0	0.0	0.0%
65	has_nape_color::grey	0.0	0.0	0.0%
68	has_nape_color::white	0.0	0.0	0.0%
69	has_nape_color::buff	0.0	0.0	0.0%

70	has_belly_color::brown	0.0	0.0	0.0%
71	has_belly_color::grey	0.0	0.0	0.0%
73	has_belly_color::black	0.0	0.0	0.0%
74	has_belly_color::white	0.0	0.0	0.0%
75	has_belly_color::buff	0.0	0.0	0.0%
76	has_wing_shape::rounded-wings	0.0	0.0	0.0%
77	has_wing_shape::pointed-wings	0.0	0.0	0.0%
79	has_size::medium_(9_-_16_in)	0.0	0.0	0.0%
80	has_size::very_small_(3_-_5_in)	0.0	0.0	0.0%
83	has_back_pattern::solid	0.0	0.0	0.0%
84	has_back_pattern::striped	0.0	0.0	0.0%
86	has_tail_pattern::solid	0.0	0.0	0.0%
87	has_tail_pattern::striped	0.0	0.0	0.0%
90	has_primary_color::brown	0.0	0.0	0.0%
91	has_primary_color::grey	0.0	0.0	0.0%
92	has_primary_color::yellow	0.0	0.0	0.0%
94	has_primary_color::white	0.0	0.0	0.0%
97	has_leg_color::black	0.0	0.0	0.0%
98	has_leg_color::buff	0.0	0.0	0.0%
100	has_bill_color::black	0.0	0.0	0.0%
101	has_bill_color::buff	0.0	0.0	0.0%
103	has_crown_color::brown	0.0	0.0	0.0%
104	has_crown_color::grey	0.0	0.0	0.0%
105	has_crown_color::yellow	0.0	0.0	0.0%
107	has_crown_color::white	0.0	0.0	0.0%
108	has_wing_pattern::solid	0.0	0.0	0.0%
109	has_wing_pattern::spotted	0.0	0.0	0.0%
111	has_wing_pattern::multi-colored	0.0	0.0	0.0%

Table A1: Concept predictions, feature attribution value and contribution shown in Figure 3.19 (a).

Concept ID	Concept name	Concept value	Feature attribution value	Concept contribution
18	has_underparts_color::yellow	1.0	-0.1877	5.833%
110	has_wing_pattern::striped	1.0	-0.1963	5.464%
61	has_under_tail_color::black	1.0	-0.2436	4.412%
88	has_tail_pattern::multi-colored	1.0	-0.1373	3.972%
72	has_belly_color::yellow	1.0	-0.1321	3.865%
99	has_bill_color::grey	1.0	-0.1431	3.845%
7	has_wing_color::black	1.0	-0.1715	3.504%
27	has_back_color::yellow	0.0	0.1915	3.079%
102	has_crown_color::blue	0.0	0.1872	2.879%
31	has_tail_shape::notched_tail	1.0	-0.068	2.83%
66	has_nape_color::yellow	0.0	0.1447	2.508%
4	has_wing_color::brown	0.0	0.1324	2.492%
89	has_belly_pattern::solid	1.0	-0.0759	2.394%
50	has_eye_color::black	1.0	-0.0686	2.385%
24	has_breast_pattern::multi-colored	0.0	0.1207	2.295%
92	has_primary_color::yellow	0.0	0.1228	2.272%
54	has_forehead_color::brown	0.0	0.1943	2.249%
33	has_upper_tail_color::grey	0.0	0.1258	2.154%
52	has_bill_length::shorter_than_head	1.0	-0.0771	2.061%
22	has_breast_pattern::solid	1.0	-0.0534	2.05%
81	has_shape::duck-like	0.0	0.1482	2.008%
106	has_crown_color::black	1.0	-0.1019	1.972%
94	has_primary_color::white	0.0	0.0939	1.931%
59	has_under_tail_color::brown	0.0	0.145	1.911%
29	has_back_color::white	0.0	0.1195	1.828%
74	has_belly_color::white	0.0	0.0612	1.741%
44	has_breast_color::buff	0.0	0.1213	1.73%
47	has_throat_color::black	1.0	-0.0882	1.672%
64	has_nape_color::brown	0.0	0.1165	1.582%
85	has_back_pattern::multi-colored	0.0	0.0994	1.414%

19	has_underparts_color::black	0.0	0.0716	1.41%
38	has_head_pattern::plain	1.0	-0.0705	1.41%
75	has_belly_color::buff	0.0	0.0689	1.354%
96	has_leg_color::grey	1.0	-0.0554	1.174%
14	has_upperparts_color::white	0.0	0.0525	1.141%
62	has_under_tail_color::white	0.0	0.0591	1.117%
93	has_primary_color::black	1.0	-0.0594	1.094%
98	has_leg_color::buff	0.0	0.074	1.046%
5	has_wing_color::grey	0.0	0.0671	1.028%
95	has_primary_color::buff	0.0	0.0753	0.978%
57	has_forehead_color::black	1.0	-0.0476	0.957%
9	has_wing_color::buff	0.0	0.0784	0.945%
30	has_back_color::buff	0.0	0.07	0.86%
104	has_crown_color::grey	0.0	0.0548	0.859%
0	has_bill_shape::dagger	0.0	0.054	0.842%
87	has_tail_pattern::striped	0.0	0.0585	0.744%
26	has_back_color::grey	0.0	0.0418	0.702%
28	has_back_color::black	1.0	-0.0373	0.616%
80	has_size::very_small_(3_-_5_in)	0.0	0.0236	0.397%
10	has_upperparts_color::brown	0.0	0.0133	0.269%
107	has_crown_color::white	0.0	0.0162	0.239%
67	has_nape_color::black	1.0	-0.0101	0.184%
90	has_primary_color::brown	0.0	0.0093	0.149%
39	has_breast_color::brown	0.0	0.011	0.138%
55	has_forehead_color::grey	0.0	0.0011	0.017%
1	has_bill_shape::hooked_seabird	0.0	-0.0	-0.0%
2	has_bill_shape::all-purpose	0.0	-0.0	-0.0%
3	has_bill_shape::cone	0.0	-0.0	-0.0%
6	has_wing_color::yellow	0.0	-0.0	-0.0%
8	has_wing_color::white	1.0	0.0	-0.0%
11	has_upperparts_color::grey	0.0	-0.0	-0.0%
12	has_upperparts_color::yellow	0.0	-0.0	-0.0%
13	has_upperparts_color::black	1.0	0.0	-0.0%
15	has_upperparts_color::buff	0.0	-0.0	-0.0%
16	has_underparts_color::brown	0.0	-0.0	-0.0%
17	has_underparts_color::grey	0.0	-0.0	-0.0%

20	has_underparts_color::white	0.0	-0.0	-0.0%
21	has_underparts_color::buff	0.0	-0.0	-0.0%
23	has_breast_pattern::striped	0.0	-0.0	-0.0%
25	has_back_color::brown	0.0	-0.0	-0.0%
32	has_upper_tail_color::brown	0.0	-0.0	-0.0%
34	has_upper_tail_color::black	0.0	-0.0	-0.0%
35	has_upper_tail_color::white	0.0	-0.0	-0.0%
36	has_upper_tail_color::buff	0.0	-0.0	-0.0%
37	has_head_pattern::eyebrow	0.0	-0.0	-0.0%
40	has_breast_color::grey	0.0	-0.0	-0.0%
41	has_breast_color::yellow	0.0	-0.0	-0.0%
42	has_breast_color::black	0.0	-0.0	-0.0%
43	has_breast_color::white	0.0	-0.0	-0.0%
45	has_throat_color::grey	0.0	-0.0	-0.0%
46	has_throat_color::yellow	0.0	-0.0	-0.0%
48	has_throat_color::white	0.0	-0.0	-0.0%
49	has_throat_color::buff	0.0	-0.0	-0.0%
51	has_bill_length::about_the_same_as_head	0.0	-0.0	-0.0%
53	has_forehead_color::blue	0.0	-0.0	-0.0%
56	has_forehead_color::yellow	0.0	-0.0	-0.0%
58	has_forehead_color::white	0.0	-0.0	-0.0%
60	has_under_tail_color::grey	0.0	-0.0	-0.0%
63	has_under_tail_color::buff	0.0	-0.0	-0.0%
65	has_nape_color::grey	0.0	-0.0	-0.0%
68	has_nape_color::white	0.0	-0.0	-0.0%
69	has_nape_color::buff	0.0	-0.0	-0.0%
70	has_belly_color::brown	0.0	-0.0	-0.0%
71	has_belly_color::grey	0.0	-0.0	-0.0%
73	has_belly_color::black	0.0	-0.0	-0.0%
76	has_wing_shape::rounded-wings	0.0	-0.0	-0.0%
77	has_wing_shape::pointed-wings	0.0	-0.0	-0.0%
78	has_size::small_(5_-_9_in)	1.0	0.0	-0.0%
79	has_size::medium_(9_-_16_in)	0.0	-0.0	-0.0%
82	has_shape::perching-like	1.0	0.0	-0.0%
83	has_back_pattern::solid	0.0	-0.0	-0.0%
84	has_back_pattern::striped	0.0	-0.0	-0.0%

86	has_tail_pattern::solid	0.0	-0.0	-0.0%
91	has_primary_color::grey	0.0	-0.0	-0.0%
97	has_leg_color::black	0.0	-0.0	-0.0%
100	has_bill_color::black	0.0	-0.0	-0.0%
101	has_bill_color::buff	0.0	-0.0	-0.0%
103	has_crown_color::brown	0.0	-0.0	-0.0%
105	has_crown_color::yellow	0.0	-0.0	-0.0%
108	has_wing_pattern::solid	0.0	-0.0	-0.0%
109	has_wing_pattern::spotted	0.0	-0.0	-0.0%
111	has_wing_pattern::multi-colored	0.0	-0.0	-0.0%

Table A2: Concept predictions, feature attribution value and contribution shown in Figure 3.19 (b).

Concept ID	Concept name	Concept value	Feature attribution value	Concept contribution
72	has_belly_color::yellow	0.9978	-0.2795	5.794%
96	has_leg_color::grey	0.9965	-0.2105	4.723%
88	has_tail_pattern::multi-colored	0.9989	-0.2308	4.306%
57	has_forehead_color::black	0.9999	-0.2625	3.759%
99	has_bill_color::grey	0.9851	-0.1078	3.269%
69	has_nape_color::buff	0.0001	0.2265	3.119%
24	has_breast_pattern::multi-colored	0.0026	0.1461	3.112%
89	has_belly_pattern::solid	0.9941	-0.1176	2.912%
7	has_wing_color::black	0.9999	-0.2148	2.835%
27	has_back_color::yellow	0.0009	0.1558	2.812%
95	has_primary_color::buff	0.0	0.2461	2.79%
25	has_back_color::brown	0.0003	0.1782	2.751%
8	has_wing_color::white	0.9896	-0.0925	2.578%

34	has_upper_tail_color::black	0.0649	0.053	2.525%
110	has_wing_pattern::striped	0.9894	-0.089	2.489%
18	has_underparts_color::yellow	0.9971	-0.1111	2.419%
66	has_nape_color::yellow	0.0014	0.1234	2.38%
38	has_head_pattern::plain	0.9989	-0.1259	2.352%
0	has_bill_shape::dagger	0.0	0.1915	2.278%
61	has_under_tail_color::black	0.9996	-0.1301	2.141%
2	has_bill_shape::all-purpose	0.0947	0.0373	2.098%
35	has_upper_tail_color::white	0.0002	0.1445	2.094%
3	has_bill_shape::cone	0.0007	0.1081	1.878%
29	has_back_color::white	0.0001	0.1338	1.77%
93	has_primary_color::black	0.989	-0.0611	1.726%
5	has_wing_color::grey	0.0	0.1405	1.675%
75	has_belly_color::buff	0.0	0.1441	1.674%
46	has_throat_color::yellow	0.0001	0.1172	1.672%
70	has_belly_color::brown	0.0	0.1598	1.602%
106	has_crown_color::black	0.9992	-0.0889	1.596%
13	has_upperparts_color::black	0.9996	-0.0959	1.564%
105	has_crown_color::yellow	0.0	0.1339	1.499%
82	has_shape::perching-like	0.9999	-0.1037	1.429%
45	has_throat_color::grey	0.0001	0.1096	1.424%
20	has_underparts_color::white	0.0018	0.0646	1.3%
87	has_tail_pattern::striped	0.0002	0.0683	1.037%
58	has_forehead_color::white	0.0	0.1111	0.993%
50	has_eye_color::black	0.9955	-0.0417	0.984%
6	has_wing_color::yellow	0.0	0.1033	0.949%
90	has_primary_color::brown	0.0	0.1013	0.949%
47	has_throat_color::black	0.9982	-0.044	0.883%
60	has_under_tail_color::grey	0.0002	0.0559	0.852%
28	has_back_color::black	0.9996	-0.0524	0.84%
53	has_forehead_color::blue	0.0	0.0669	0.75%
100	has_bill_color::black	0.0675	0.0148	0.716%
71	has_belly_color::grey	0.0	0.0588	0.699%
65	has_nape_color::grey	0.0	0.0624	0.626%
49	has_throat_color::buff	0.0	0.0678	0.624%
37	has_head_pattern::eyebrow	0.0	0.0537	0.532%

78	has_size::small_(5_-_9_in)	0.9997	-0.0252	0.399%
52	has_bill_length::shorter_than_head	0.9854	-0.0105	0.317%
109	has_wing_pattern::spotted	0.0	0.0365	0.315%
101	has_bill_color::buff	0.0	0.0297	0.313%
98	has_leg_color::buff	0.0	0.032	0.304%
31	has_tail_shape::notched_tail	0.9097	-0.0053	0.294%
1	has_bill_shape::hooked_seabird	0.0	0.0201	0.215%
17	has_underparts_color::grey	0.0	0.0036	0.038%
76	has_wing_shape::rounded-wings	0.1	0.0005	0.027%
4	has_wing_color::brown	0.0	-0.0	-0.0%
9	has_wing_color::buff	0.0	-0.0	-0.0%
10	has_upperparts_color::brown	0.0	-0.0	-0.0%
11	has_upperparts_color::grey	0.0	-0.0	-0.0%
12	has_upperparts_color::yellow	0.0003	-0.0	-0.0%
14	has_upperparts_color::white	0.0002	-0.0	-0.0%
15	has_upperparts_color::buff	0.0	-0.0	-0.0%
16	has_underparts_color::brown	0.0	-0.0	-0.0%
19	has_underparts_color::black	0.0	-0.0	-0.0%
21	has_underparts_color::buff	0.0	-0.0	-0.0%
22	has_breast_pattern::solid	0.808	0.0	-0.0%
23	has_breast_pattern::striped	0.0	-0.0	-0.0%
26	has_back_color::grey	0.0	-0.0	-0.0%
30	has_back_color::buff	0.0	-0.0	-0.0%
32	has_upper_tail_color::brown	0.0	-0.0	-0.0%
33	has_upper_tail_color::grey	0.0	-0.0	-0.0%
36	has_upper_tail_color::buff	0.0	-0.0	-0.0%
39	has_breast_color::brown	0.0	-0.0	-0.0%
40	has_breast_color::grey	0.0	-0.0	-0.0%
41	has_breast_color::yellow	0.0015	-0.0	-0.0%
42	has_breast_color::black	0.0022	-0.0	-0.0%
43	has_breast_color::white	0.0	-0.0	-0.0%
44	has_breast_color::buff	0.0	-0.0	-0.0%
48	has_throat_color::white	0.0	-0.0	-0.0%
51	has_bill_length::about_the_same_as_head	0.001	-0.0	-0.0%
54	has_forehead_color::brown	0.0	-0.0	-0.0%
55	has_forehead_color::grey	0.0	-0.0	-0.0%

56	has_forehead_color::yellow	0.0	-0.0	-0.0%
59	has_under_tail_color::brown	0.0	-0.0	-0.0%
62	has_under_tail_color::white	0.0	-0.0	-0.0%
63	has_under_tail_color::buff	0.0	-0.0	-0.0%
64	has_nape_color::brown	0.0	-0.0	-0.0%
67	has_nape_color::black	0.9982	0.0	-0.0%
68	has_nape_color::white	0.0	-0.0	-0.0%
73	has_belly_color::black	0.0	-0.0	-0.0%
74	has_belly_color::white	0.0058	-0.0	-0.0%
77	has_wing_shape::pointed-wings	0.0001	-0.0	-0.0%
79	has_size::medium_(9_-_16_in)	0.0	-0.0	-0.0%
80	has_size::very_small_(3_-_5_in)	0.0001	-0.0	-0.0%
81	has_shape::duck-like	0.0	-0.0	-0.0%
83	has_back_pattern::solid	0.1099	-0.0	-0.0%
84	has_back_pattern::striped	0.0001	-0.0	-0.0%
85	has_back_pattern::multi-colored	0.0	-0.0	-0.0%
86	has_tail_pattern::solid	0.0072	-0.0	-0.0%
91	has_primary_color::grey	0.0	-0.0	-0.0%
92	has_primary_color::yellow	0.0008	-0.0	-0.0%
94	has_primary_color::white	0.0004	-0.0	-0.0%
97	has_leg_color::black	0.0004	-0.0	-0.0%
102	has_crown_color::blue	0.0	-0.0	-0.0%
103	has_crown_color::brown	0.0	-0.0	-0.0%
104	has_crown_color::grey	0.0	-0.0	-0.0%
107	has_crown_color::white	0.0	-0.0	-0.0%
108	has_wing_pattern::solid	0.0001	-0.0	-0.0%
111	has_wing_pattern::multi-colored	0.0202	-0.0	-0.0%

Table A3: Concept predictions, feature attribution value and contribution shown in Figure 3.19 (c).

Concept ID	Concept name	Concept value	Feature attribution value	Concept contribution
99	has_bill_color::grey	0.9999	0.0758	7.551%
88	has_tail_pattern::multi-colored	1.0	0.0691	6.883%
72	has_belly_color::yellow	1.0	0.0655	6.523%
18	has_underparts_color::yellow	1.0	0.0623	6.204%
110	has_wing_pattern::striped	1.0	0.0594	5.919%
47	has_throat_color::black	1.0	0.0525	5.227%
8	has_wing_color::white	0.9999	0.0517	5.153%
96	has_leg_color::grey	1.0	0.0514	5.124%
38	has_head_pattern::plain	1.0	0.0509	5.069%
67	has_nape_color::black	1.0	0.0485	4.833%
28	has_back_color::black	1.0	0.0447	4.455%
93	has_primary_color::black	1.0	0.0366	3.643%
61	has_under_tail_color::black	1.0	0.0365	3.634%
106	has_crown_color::black	1.0	0.0347	3.456%
13	has_upperparts_color::black	0.9999	0.0313	3.117%
50	has_eye_color::black	0.9997	0.0306	3.045%
31	has_tail_shape::notched_tail	0.9987	0.0298	2.972%
7	has_wing_color::black	1.0	0.0296	2.946%
89	has_belly_pattern::solid	0.9998	0.0283	2.821%
82	has_shape::perching-like	1.0	0.0269	2.683%
22	has_breast_pattern::solid	0.999	0.0267	2.663%
57	has_forehead_color::black	0.9999	0.0235	2.341%
78	has_size::small_(5_-_9_in)	0.9996	0.0208	2.074%
52	has_bill_length::shorter_than_head	0.9985	0.0129	1.292%
16	has_underparts_color::brown	0.0	0.0	0.311%
81	has_shape::duck-like	0.0	0.0	0.05%
87	has_tail_pattern::striped	0.0	0.0	0.009%
0	has_bill_shape::dagger	0.0	0.0	0.0%
1	has_bill_shape::hooked_seabird	0.0	0.0	0.0%
2	has_bill_shape::all-purpose	0.0004	0.0	0.0%

3	has_bill_shape::cone	0.0	0.0	0.0%
4	has_wing_color::brown	0.0	0.0	0.0%
5	has_wing_color::grey	0.0	0.0	0.0%
6	has_wing_color::yellow	0.0	0.0	0.0%
9	has_wing_color::buff	0.0	0.0	0.0%
10	has_upperparts_color::brown	0.0	0.0	0.0%
11	has_upperparts_color::grey	0.0	0.0	0.0%
12	has_upperparts_color::yellow	0.0	0.0	0.0%
14	has_upperparts_color::white	0.0001	0.0	0.0%
15	has_upperparts_color::buff	0.0	0.0	0.0%
17	has_underparts_color::grey	0.0	0.0	0.0%
19	has_underparts_color::black	0.0	0.0	0.0%
20	has_underparts_color::white	0.0001	0.0	0.0%
21	has_underparts_color::buff	0.0	0.0	0.0%
23	has_breast_pattern::striped	0.0	0.0	0.0%
24	has_breast_pattern::multi-colored	0.0	0.0	0.0%
25	has_back_color::brown	0.0	0.0	0.0%
26	has_back_color::grey	0.0	0.0	0.0%
27	has_back_color::yellow	0.0	0.0	0.0%
29	has_back_color::white	0.0	0.0	0.0%
30	has_back_color::buff	0.0	0.0	0.0%
32	has_upper_tail_color::brown	0.0	0.0	0.0%
33	has_upper_tail_color::grey	0.0	0.0	0.0%
34	has_upper_tail_color::black	0.0	0.0	0.0%
35	has_upper_tail_color::white	0.0	0.0	0.0%
36	has_upper_tail_color::buff	0.0	0.0	0.0%
37	has_head_pattern::eyebrow	0.0	0.0	0.0%
39	has_breast_color::brown	0.0	0.0	0.0%
40	has_breast_color::grey	0.0	0.0	0.0%
41	has_breast_color::yellow	0.0002	0.0	0.0%
42	has_breast_color::black	0.0	0.0	0.0%
43	has_breast_color::white	0.0	0.0	0.0%
44	has_breast_color::buff	0.0	0.0	0.0%
45	has_throat_color::grey	0.0	0.0	0.0%
46	has_throat_color::yellow	0.0	0.0	0.0%
48	has_throat_color::white	0.0	0.0	0.0%

49	has_throat_color::buff	0.0	0.0	0.0%
51	has_bill_length::about_the_same_as_head	0.0003	0.0	0.0%
53	has_forehead_color::blue	0.0	0.0	0.0%
54	has_forehead_color::brown	0.0	0.0	0.0%
55	has_forehead_color::grey	0.0	0.0	0.0%
56	has_forehead_color::yellow	0.0	0.0	0.0%
58	has_forehead_color::white	0.0	0.0	0.0%
59	has_under_tail_color::brown	0.0	0.0	0.0%
60	has_under_tail_color::grey	0.0	0.0	0.0%
62	has_under_tail_color::white	0.0	0.0	0.0%
63	has_under_tail_color::buff	0.0	0.0	0.0%
64	has_nape_color::brown	0.0	0.0	0.0%
65	has_nape_color::grey	0.0	0.0	0.0%
66	has_nape_color::yellow	0.0	0.0	0.0%
68	has_nape_color::white	0.0	0.0	0.0%
69	has_nape_color::buff	0.0	0.0	0.0%
70	has_belly_color::brown	0.0	0.0	0.0%
71	has_belly_color::grey	0.0	0.0	0.0%
73	has_belly_color::black	0.0	0.0	0.0%
74	has_belly_color::white	0.0001	0.0	0.0%
75	has_belly_color::buff	0.0	0.0	0.0%
76	has_wing_shape::rounded-wings	0.0006	0.0	0.0%
77	has_wing_shape::pointed-wings	0.0	0.0	0.0%
79	has_size::medium_(9_-_16_in)	0.0	0.0	0.0%
80	has_size::very_small_(3_-_5_in)	0.0	0.0	0.0%
83	has_back_pattern::solid	0.0002	0.0	0.0%
84	has_back_pattern::striped	0.0	0.0	0.0%
85	has_back_pattern::multi-colored	0.0	0.0	0.0%
86	has_tail_pattern::solid	0.0005	0.0	0.0%
90	has_primary_color::brown	0.0	0.0	0.0%
91	has_primary_color::grey	0.0	0.0	0.0%
92	has_primary_color::yellow	0.0001	0.0	0.0%
94	has_primary_color::white	0.0	0.0	0.0%
95	has_primary_color::buff	0.0	0.0	0.0%
97	has_leg_color::black	0.0	0.0	0.0%
98	has_leg_color::buff	0.0	0.0	0.0%

100	has_bill_color::black	0.001	0.0	0.0%
101	has_bill_color::buff	0.0	0.0	0.0%
102	has_crown_color::blue	0.0	0.0	0.0%
103	has_crown_color::brown	0.0	0.0	0.0%
104	has_crown_color::grey	0.0	0.0	0.0%
105	has_crown_color::yellow	0.0	0.0	0.0%
107	has_crown_color::white	0.0	0.0	0.0%
108	has_wing_pattern::solid	0.0	0.0	0.0%
109	has_wing_pattern::spotted	0.0	0.0	0.0%
111	has_wing_pattern::multi-colored	0.0001	0.0	0.0%

Table A4: Concept predictions, feature attribution value and contribution shown in Figure 3.19 (d).

B Human Studies

B.1 Results

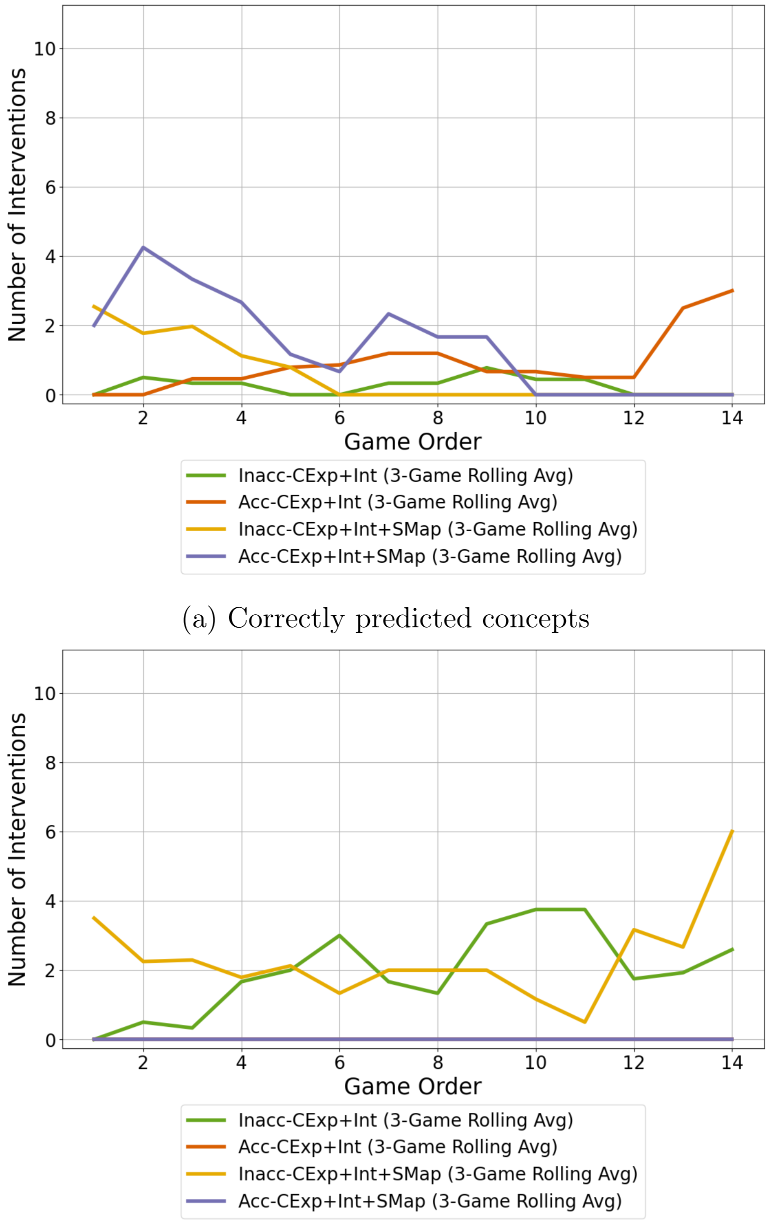
Here we include additional details regarding our human studies. In Table B1 we expand on participant task accuracy from the lay-person study by including accuracy per task labels. Overall participants across the different groups correctly identify when they should select the “Hit” label. There is a small drop in task accuracy for participants with the inaccurate model. However, this drop is completely removed for participants who performed interventions.

Participants were less accurate at labelling samples with the “Stand” option, possibly reflecting differences in the AI and participants’ game tactics. Participants who performed interventions improved their task accuracy if they also were using the inaccurate model. The reverse is true for participants with the accurate model.

The “Surrender” label is rarely the best label to select Shackleford (2023). In addition, participants may be motivated to avoid using it as they do not have any money or other motivation to maximise their score as it has no value outside of the study. This can be used to explain this label achieving a 0% accuracy for most participant groups. We do not have a large set of samples to draw any further conclusions from this label.

Data Subset	Overall Accuracy (%)	Hit (%)	Stand (%)	Surrender (%)
AI disabled	74.4 (±3.9)	90.3 (±3.6)	68.5 (±5.0)	0.0 (±0.0)
All	83.6 (±0.9)	90.8 (±1.2)	78.8 (±1.2)	16.7 (±3.8)
WithInt	83.3 (±1.1)	89.8 (±1.5)	78.9 (±1.4)	21.4 (±4.7)
NoInt	84.7 (±1.8)	94.3 (±1.4)	78.1 (±2.8)	0.0 (±0.0)
Acc	84.6 (±2.7)	93.5 (±2.6)	77.8 (±3.8)	0.0 (±0.0)
Inacc	78.1 (±3.2)	87.6 (±4.5)	72.1 (±3.2)	50.0 (±19.6)
Acc-NoExp	84.6 (±2.7)	93.5 (±2.6)	77.8 (±3.8)	0.0 (±0.0)
Inacc-NoExp	78.1 (±3.2)	87.6 (±4.5)	72.1 (±3.2)	50.0 (±19.6)
Acc-CExp	91.0 (±2.4)	93.4 (±3.0)	89.7 (±3.8)	100.0 (±0.0)
Inacc-CExp	81.4 (±1.6)	89.6 (±2.9)	77.0 (±2.6)	0.0 (±0.0)
Acc-CExp+Int-NoInt	86.6 (±3.3)	87.9 (±4.1)	85.9 (±4.1)	100.0 (±0.0)
Acc-CExp+Int-WithInt	83.8 (±2.9)	95.3 (±3.0)	75.4 (±4.5)	0.0 (±0.0)
Inacc-CExp+Int-NoInt	75.7 (±4.6)	79.4 (±10.0)	73.3 (±2.9)	0.0 (±0.0)
Inacc-CExp+Int-WithInt	84.3 (±1.8)	92.6 (±1.5)	78.2 (±3.0)	0.0 (±0.0)
Acc-CExp+Int+SMap-NoInt	83.4 (±2.4)	93.4 (±2.7)	75.8 (±4.2)	0.0 (±0.0)
Acc-CExp+Int+SMap-WithInt	83.4 (±6.4)	100.0 (±0.0)	71.3 (±10.2)	-
Inacc-CExp+Int+SMap-NoInt	83.5 (±2.9)	89.1 (±4.5)	78.7 (±2.0)	-
Inacc-CExp+Int+SMap-WithInt	86.6 (±3.6)	91.0 (±3.6)	84.9 (±4.2)	0.0 (±0.0)

Table B1: lay-person study human task accuracy averaged by participant.

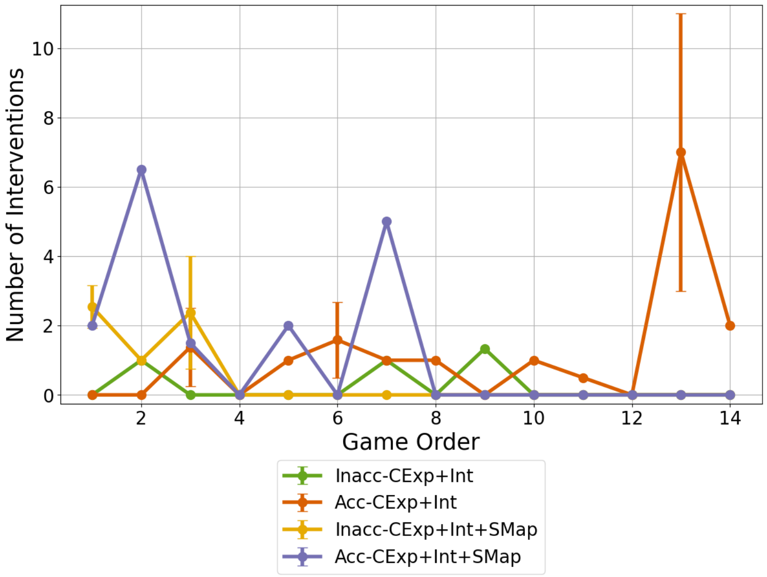


(a) Correctly predicted concepts

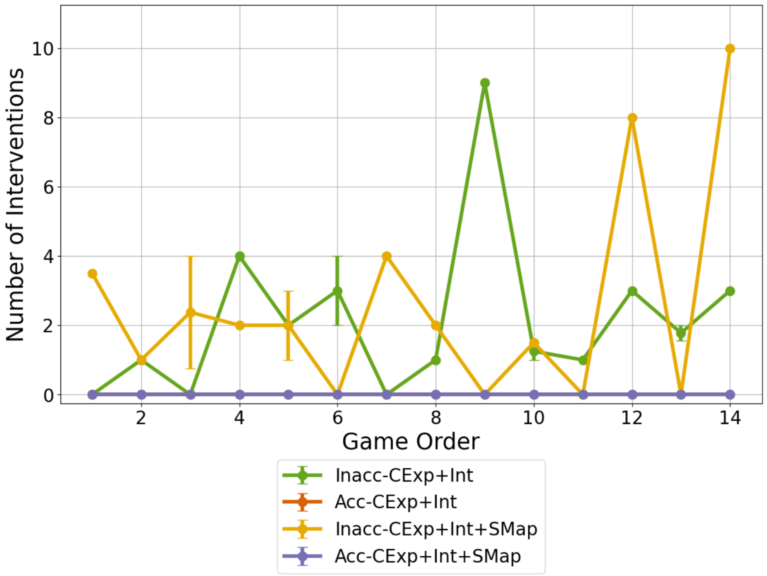
(b) Incorrectly predicted concepts

Figure B1: The rolling average of interventions performed by participants show a decline over time when the models correctly predict concepts, and a consistent number when the model incorrectly predicts concepts.

In Chapter 5, we analysed the number of interventions performed by each participant group in our lay-person study. Here we include these results again in Figure B1 in addition to including the average number of interventions performed per game in Figure B2. Continuing the trend observed with a rolling average, the



(a) Correctly predicted concepts



(b) Incorrectly predicted concepts

Figure B2: Interventions performed by participants show a decline over time with a few spikes in interventions counts.

per game average also shows the number of interventions performed declines over time when the model correctly predicts concepts, and remains consistent when the model incorrectly predicts concepts.

B.2 System Causability Scale

The System Causability Scale was used to analyse the suitability of model explanations (Holzinger et al., 2020). The full set of questions are included in Table B2. We made a minor modification to the questions in the lay-person study by changing “my work” to “the game” in Question 2, and including the example “strategy guides” instead of “medical guidelines” in Question 9.

Number	Tag	Question
1	Factors in data	I found that the data included all relevant known causal factors with sufficient precision and granularity.
2	Understood	I understood the explanations within the context of my work.
3	Change detail level	I could change the level of detail on demand.
4	Need support	I did not need support to understand the explanations.
5	Understanding causality	I found the explanations helped me to understand causality.
6	Use with knowledge	I was able to use the explanations with my knowledge base.
7	No inconsistencies	I did not find inconsistencies between explanations.
8	Learn to understand	I think that most people would learn to understand the explanations very quickly.
9	Needs references	I did not need more references in the explanations: e.g., medical guidelines, regulations.
10	Efficient	I received the explanations in a timely and efficient manner.

Table B2: System Causability Scale questions.

B.3 Study details

Both the expert and lay-person studies shared similar interfaces and model interaction design, ensuring consistency across both experiments. This section provides screenshots of the interface for both studies and outlines key details from the ethics documentation. The studies received favourable ethical opinions from the School of Computer Science and Informatics at Cardiff University. The reference numbers for the studies are *COMSC/Ethics/2023/144* (expert study) and *COMSC/Ethics/2023/146* (lay-person study).

B.3.1 Expert Study

Participant Recruitment: Participants for the expert study were recruited based on their professional or educational background in dermatology. This included individuals who were doctors, and consultants and trainees with experience in dermatology.

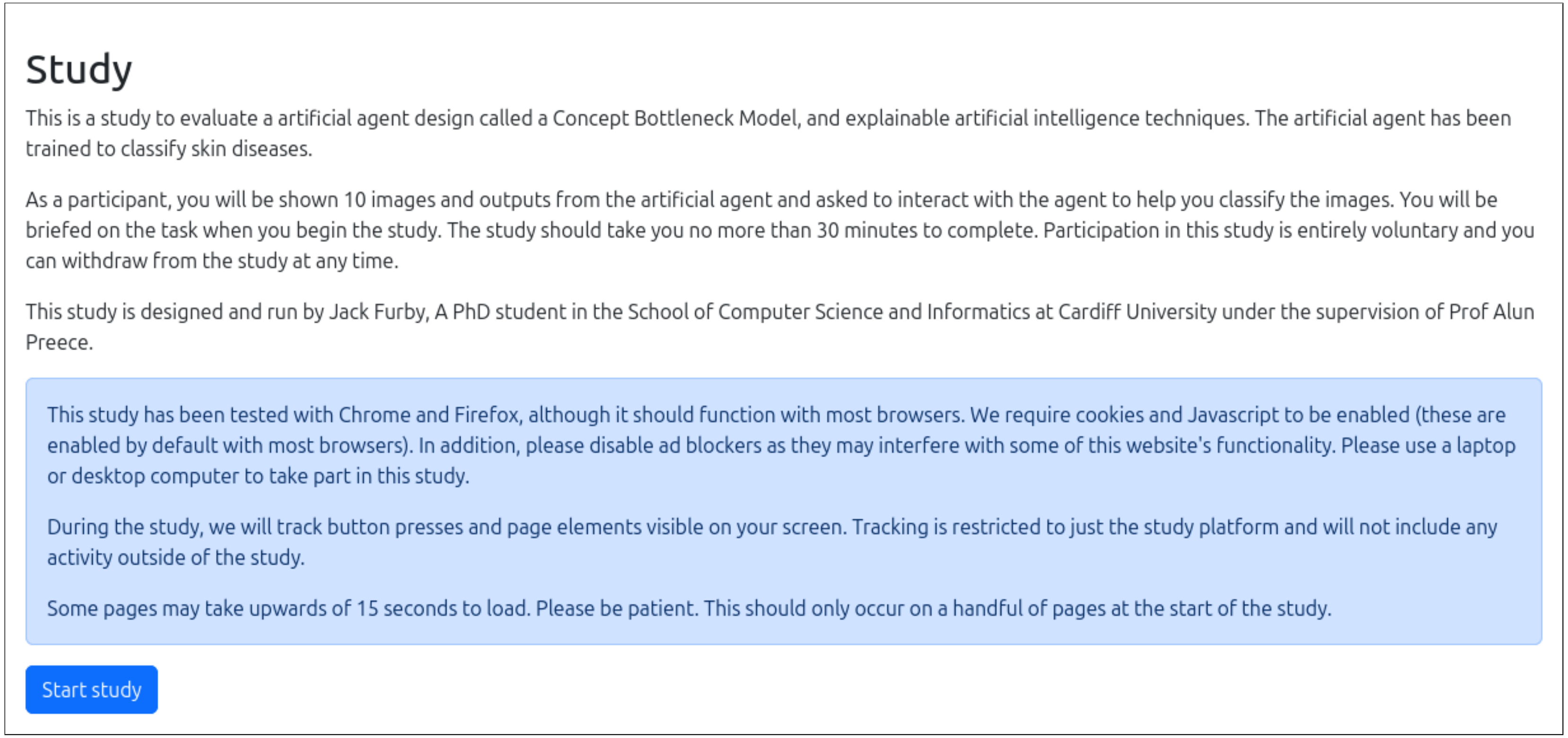
Inclusion and Exclusion Criteria:

- **Inclusion Criteria**: Participants must be over the age of 18, fluent in English, and have a background in medicine.
- **Exclusion Criteria**: Individuals with visual impairments were excluded, as participants needed the ability to see input images and differentiate between the colours red and blue.

Personal Data Collected: During the study we collected the following information from participants:

- Self declared experience level in skin disease identification
- Self declared experience in Computer Science
- Age
- Gender

Below are screenshots of the expert study interface that were shown to participants:



Study

This is a study to evaluate a artificial agent design called a Concept Bottleneck Model, and explainable artificial intelligence techniques. The artificial agent has been trained to classify skin diseases.

As a participant, you will be shown 10 images and outputs from the artificial agent and asked to interact with the agent to help you classify the images. You will be briefed on the task when you begin the study. The study should take you no more than 30 minutes to complete. Participation in this study is entirely voluntary and you can withdraw from the study at any time.

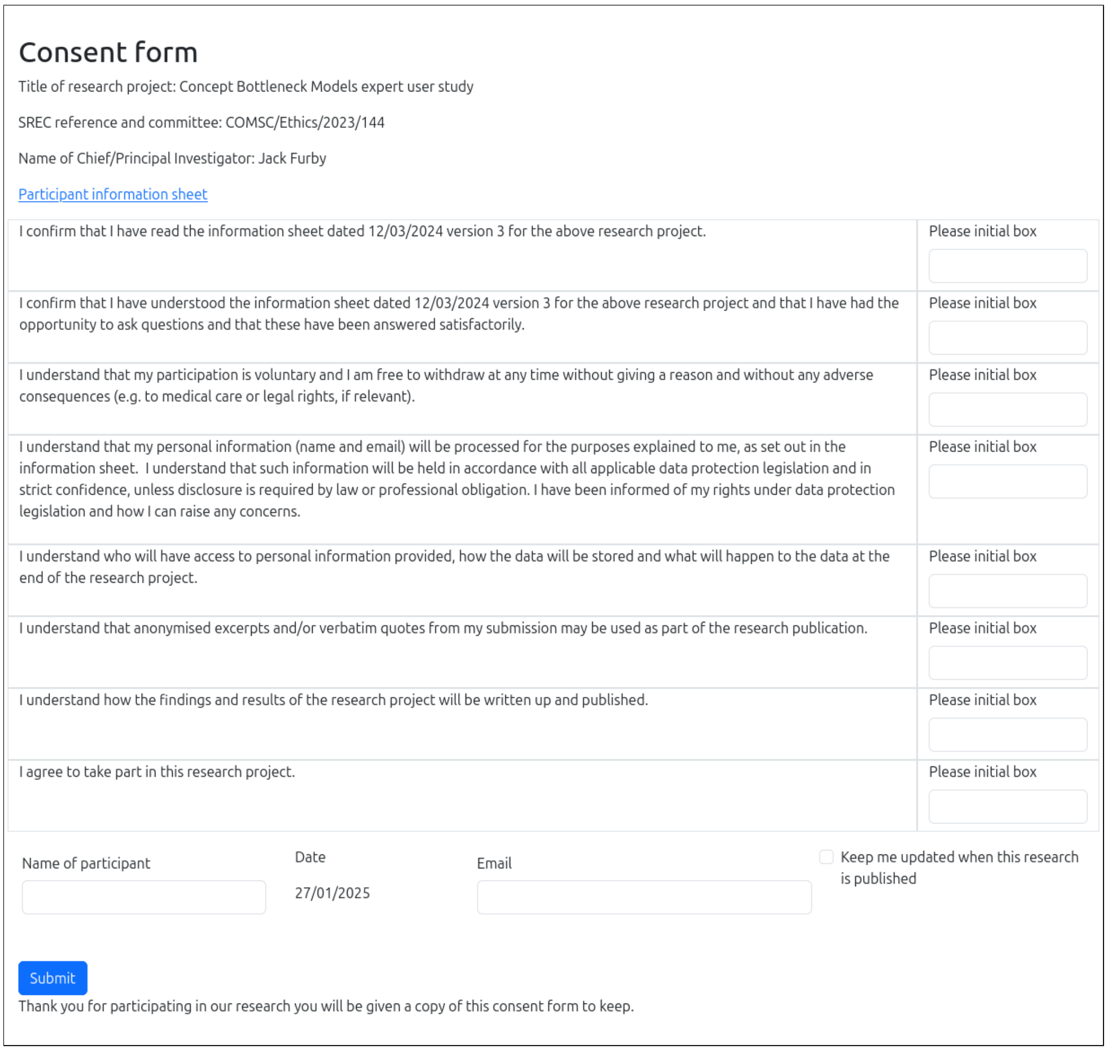
This study is designed and run by Jack Furby, A PhD student in the School of Computer Science and Informatics at Cardiff University under the supervision of Prof Alun Preece.

This study has been tested with Chrome and Firefox, although it should function with most browsers. We require cookies and Javascript to be enabled (these are enabled by default with most browsers). In addition, please disable ad blockers as they may interfere with some of this website's functionality. Please use a laptop or desktop computer to take part in this study.

During the study, we will track button presses and page elements visible on your screen. Tracking is restricted to just the study platform and will not include any activity outside of the study.

Some pages may take upwards of 15 seconds to load. Please be patient. This should only occur on a handful of pages at the start of the study.

Figure B3: Expert study interface welcome page.



Consent form

Title of research project: Concept Bottleneck Models expert user study

SREC reference and committee: COMSC/Ethics/2023/144

Name of Chief/Principal Investigator: Jack Furby

Participant information sheet

I confirm that I have read the information sheet dated 12/03/2024 version 3 for the above research project.	Please initial box
I confirm that I have understood the information sheet dated 12/03/2024 version 3 for the above research project and that I have had the opportunity to ask questions and that these have been answered satisfactorily.	Please initial box
I understand that my participation is voluntary and I am free to withdraw at any time without giving a reason and without any adverse consequences (e.g. to medical care or legal rights, if relevant).	Please initial box
I understand that my personal information (name and email) will be processed for the purposes explained to me, as set out in the information sheet. I understand that such information will be held in accordance with all applicable data protection legislation and in strict confidence, unless disclosure is required by law or professional obligation. I have been informed of my rights under data protection legislation and how I can raise any concerns.	Please initial box
I understand who will have access to personal information provided, how the data will be stored and what will happen to the data at the end of the research project.	Please initial box
I understand that anonymised excerpts and/or verbatim quotes from my submission may be used as part of the research publication.	Please initial box
I understand how the findings and results of the research project will be written up and published.	Please initial box
I agree to take part in this research project.	Please initial box

Name of participant

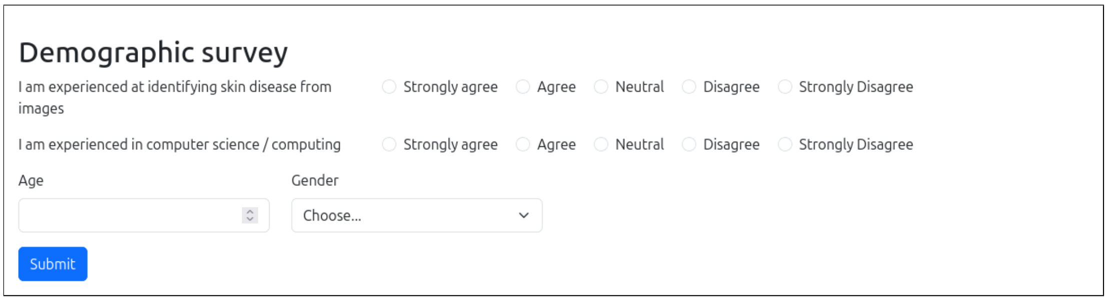
Date
27/01/2025

Email

□ Keep me updated when this research is published

Thank you for participating in our research you will be given a copy of this consent form to keep.

29[Expert study interface consent page]Expert study interface consent page.



Demographic survey

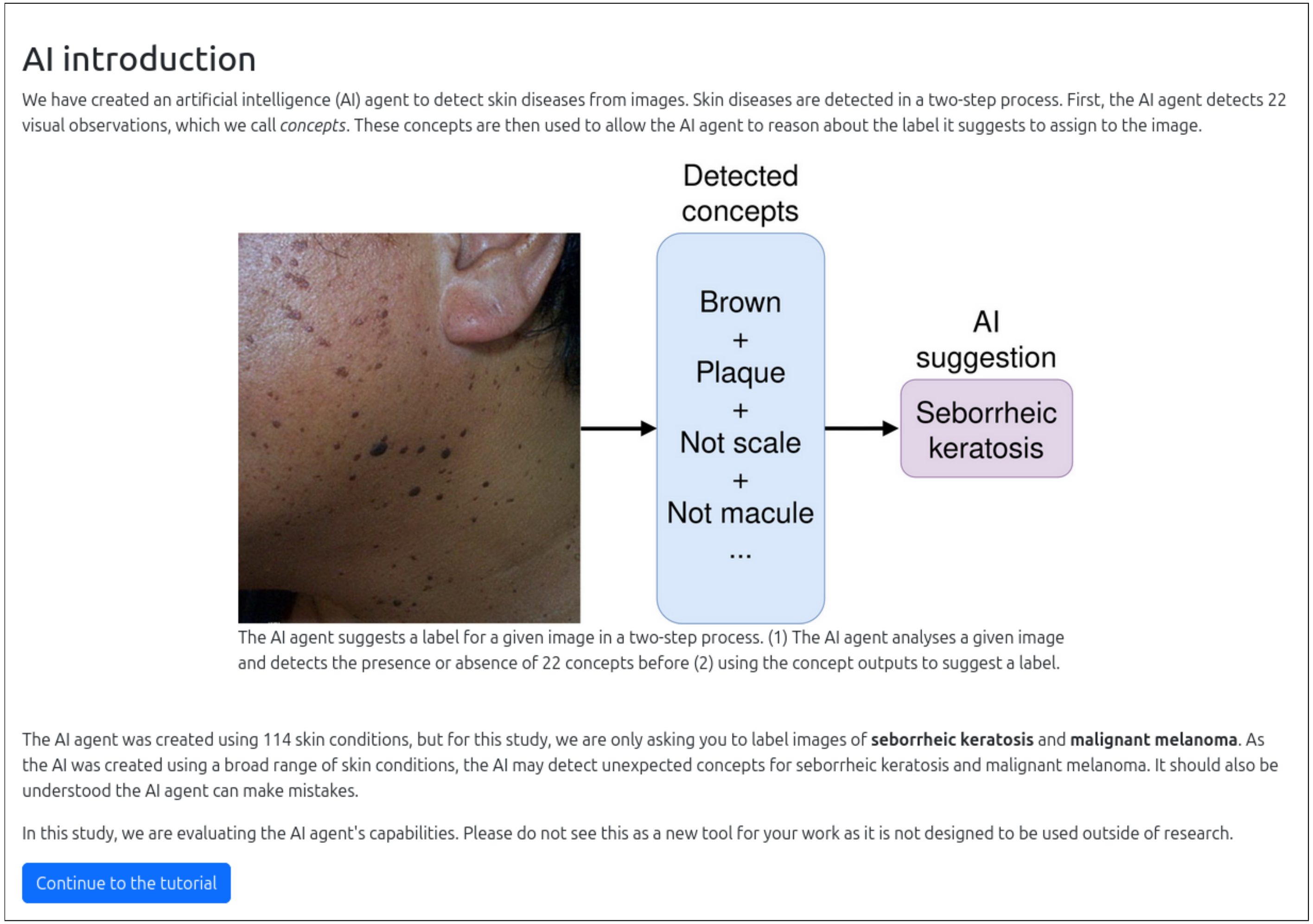
I am experienced at identifying skin disease from images ○ Strongly agree ○ Agree ○ Neutral ○ Disagree ○ Strongly Disagree

I am experienced in computer science / computing ○ Strongly agree ○ Agree ○ Neutral ○ Disagree ○ Strongly Disagree

Age

Gender
Choose...

Figure B4: Expert study interface demographic page.



AI introduction

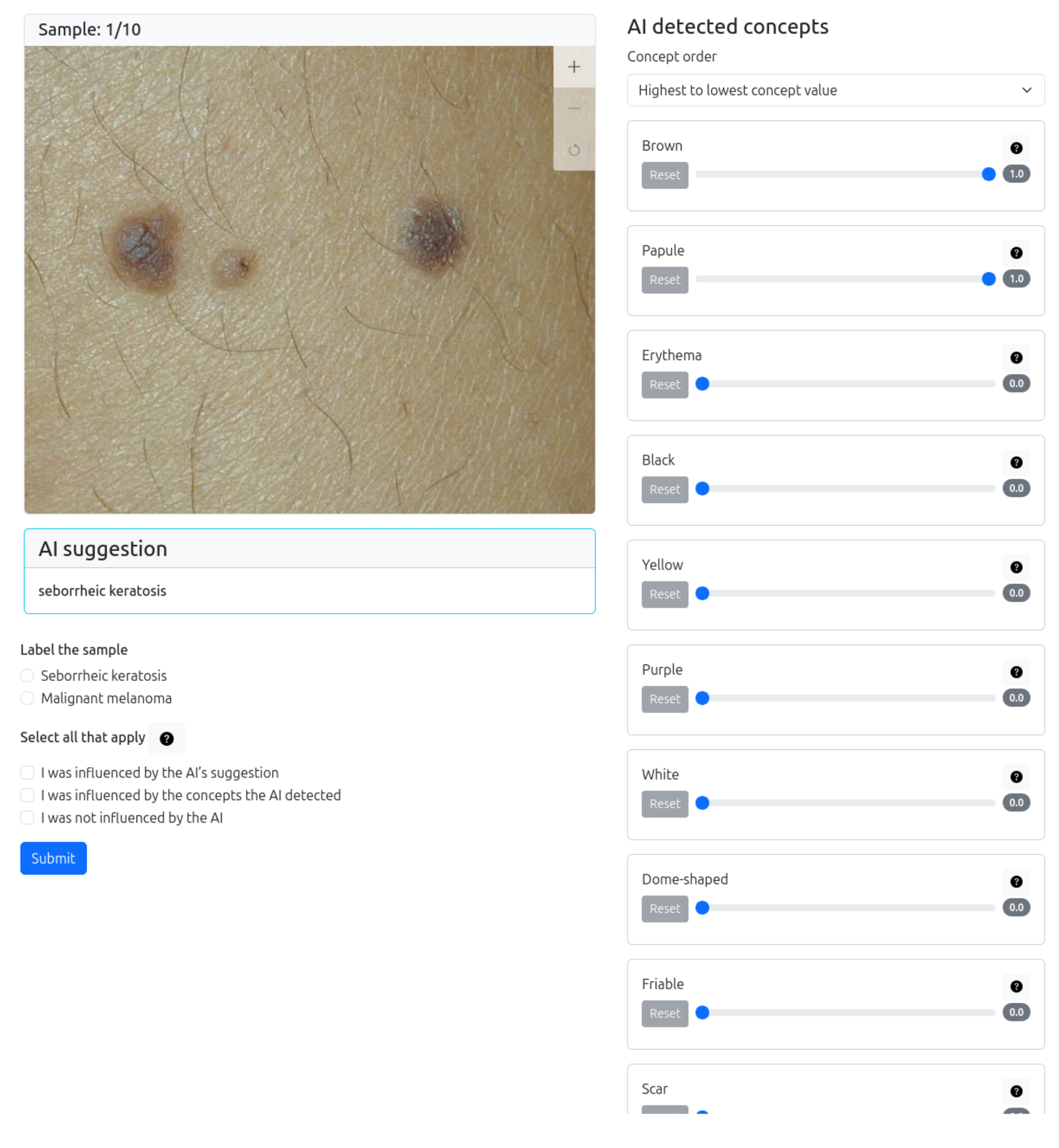
We have created an artificial intelligence (AI) agent to detect skin diseases from images. Skin diseases are detected in a two-step process. First, the AI agent detects 22 visual observations, which we call *concepts*. These concepts are then used to allow the AI agent to reason about the label it suggests to assign to the image.

The AI agent suggests a label for a given image in a two-step process. (1) The AI agent analyses a given image and detects the presence or absence of 22 concepts before (2) using the concept outputs to suggest a label.

The AI agent was created using 114 skin conditions, but for this study, we are only asking you to label images of **seborrheic keratosis** and **malignant melanoma**. As the AI was created using a broad range of skin conditions, the AI may detect unexpected concepts for seborrheic keratosis and malignant melanoma. It should also be understood the AI agent can make mistakes.

In this study, we are evaluating the AI agent's capabilities. Please do not see this as a new tool for your work as it is not designed to be used outside of research.

Figure B5: Expert study interface introduction page.



Sample: 1/10

AI suggestion

seborrheic keratosis

Label the sample

○ Seborrheic keratosis
○ Malignant melanoma

Select all that apply

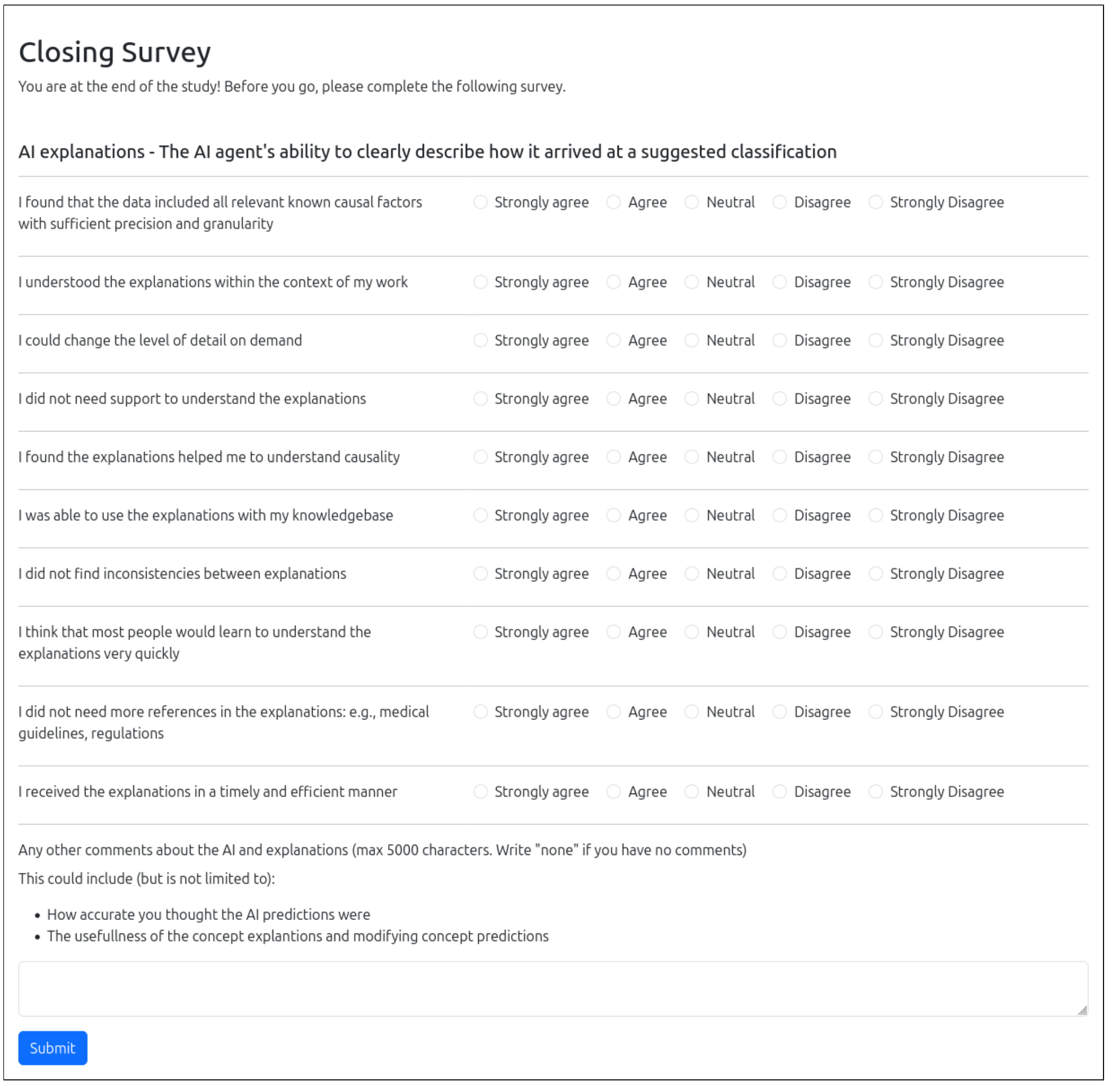
□ I was influenced by the AI's suggestion
□ I was influenced by the concepts the AI detected
□ I was not influenced by the AI

AI detected concepts

Concept order: Highest to lowest concept value

Brown	1.0
Papule	1.0
Erythema	0.0
Black	0.0
Yellow	0.0
Purple	0.0
White	0.0
Dome-shaped	0.0
Friable	0.0
Scar	

Figure B6: Expert study interface main page.



Closing Survey

You are at the end of the study! Before you go, please complete the following survey.

AI explanations - The AI agent's ability to clearly describe how it arrived at a suggested classification

I found that the data included all relevant known causal factors with sufficient precision and granularity	○ Strongly agree	○ Agree	○ Neutral	○ Disagree	○ Strongly Disagree
I understood the explanations within the context of my work	○ Strongly agree	○ Agree	○ Neutral	○ Disagree	○ Strongly Disagree
I could change the level of detail on demand	○ Strongly agree	○ Agree	○ Neutral	○ Disagree	○ Strongly Disagree
I did not need support to understand the explanations	○ Strongly agree	○ Agree	○ Neutral	○ Disagree	○ Strongly Disagree
I found the explanations helped me to understand causality	○ Strongly agree	○ Agree	○ Neutral	○ Disagree	○ Strongly Disagree
I was able to use the explanations with my knowledgebase	○ Strongly agree	○ Agree	○ Neutral	○ Disagree	○ Strongly Disagree
I did not find inconsistencies between explanations	○ Strongly agree	○ Agree	○ Neutral	○ Disagree	○ Strongly Disagree
I think that most people would learn to understand the explanations very quickly	○ Strongly agree	○ Agree	○ Neutral	○ Disagree	○ Strongly Disagree
I did not need more references in the explanations: e.g., medical guidelines, regulations	○ Strongly agree	○ Agree	○ Neutral	○ Disagree	○ Strongly Disagree
I received the explanations in a timely and efficient manner	○ Strongly agree	○ Agree	○ Neutral	○ Disagree	○ Strongly Disagree

Any other comments about the AI and explanations (max 5000 characters. Write "none" if you have no comments)

This could include (but is not limited to):

- How accurate you thought the AI predictions were
- The usefullness of the concept explantions and modifying concept predictions

Figure B7: Expert study interface survey page.

B.3.2 Lay-person Study

Participant Recruitment: Participants for the lay-person study were recruited via emails sent to mailing lists, advertisements within Cardiff University, and social media posts.

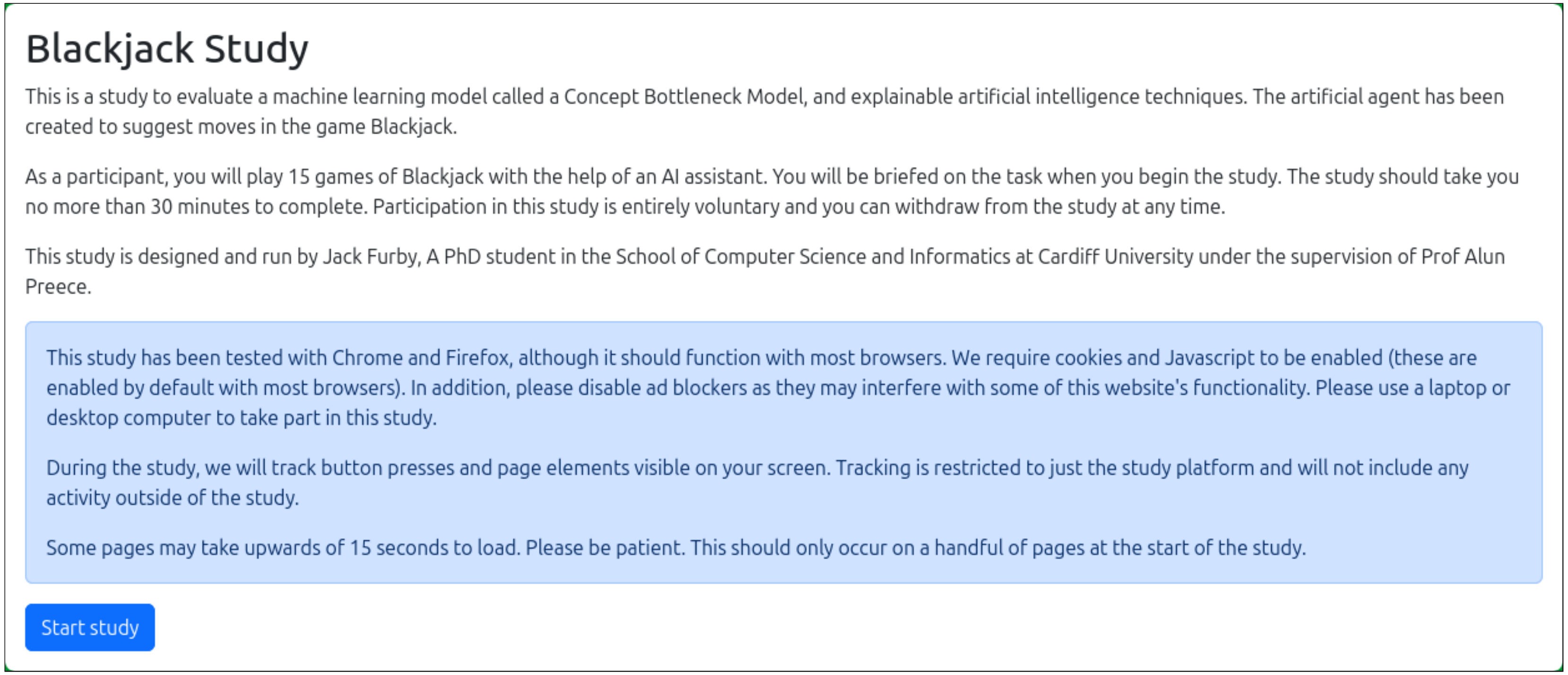
Inclusion and Exclusion Criteria:

- **Inclusion Criteria**: Participants must be over the age of 18 and fluent in English.
- **Exclusion Criteria**: Individuals with visual impairments were excluded, as participants needed the ability to see input images and differentiate between the colours red and blue.

Personal Data Collected: During the study we collected the following information from participants:

- Self declared experience level playing the card game Blackjack
- Self declared experience in Computer Science
- Age
- Gender

Below are screenshots of the expert study interface that were shown to participants:



Blackjack Study

This is a study to evaluate a machine learning model called a Concept Bottleneck Model, and explainable artificial intelligence techniques. The artificial agent has been created to suggest moves in the game Blackjack.

As a participant, you will play 15 games of Blackjack with the help of an AI assistant. You will be briefed on the task when you begin the study. The study should take you no more than 30 minutes to complete. Participation in this study is entirely voluntary and you can withdraw from the study at any time.

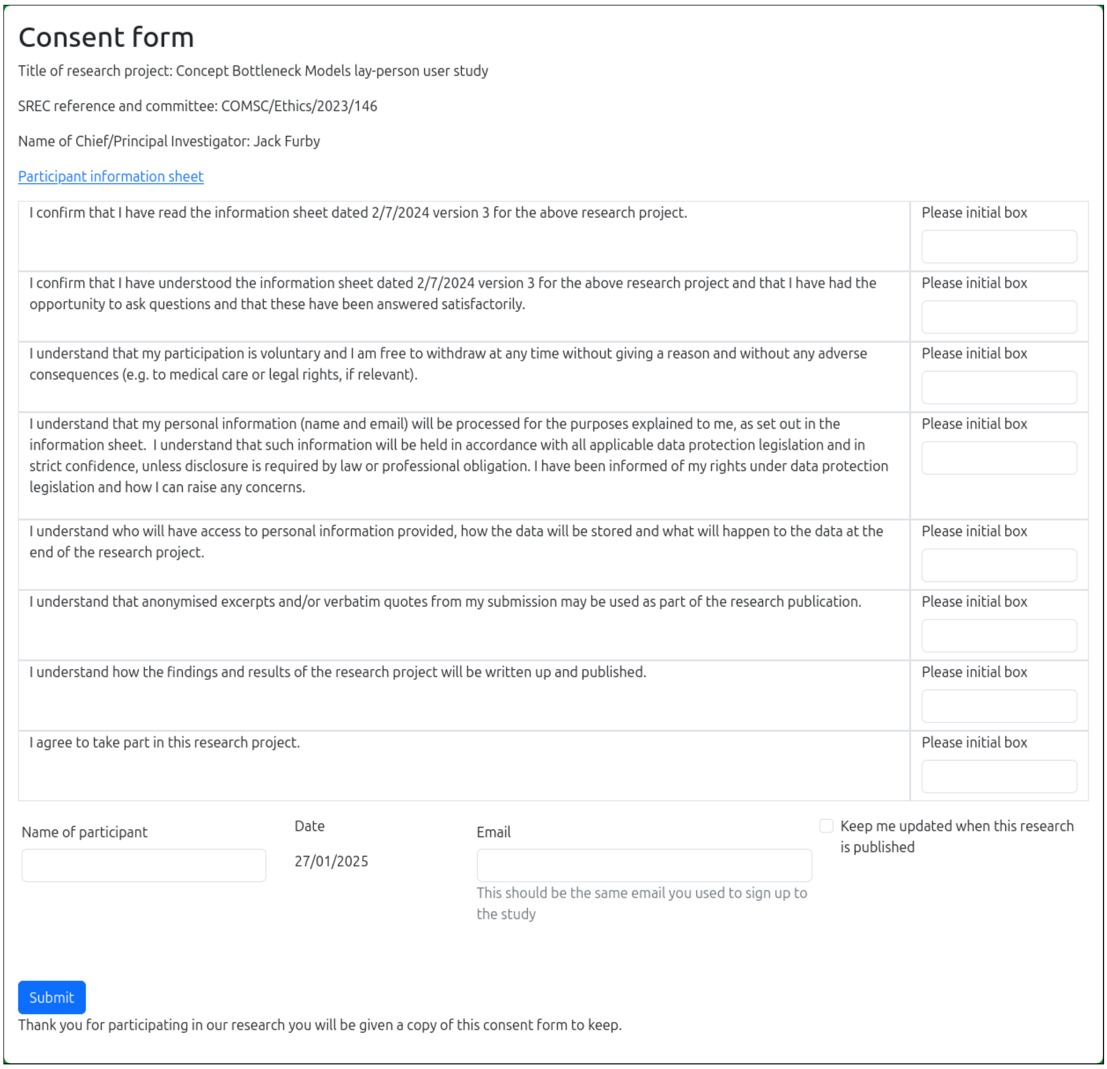
This study is designed and run by Jack Furby, A PhD student in the School of Computer Science and Informatics at Cardiff University under the supervision of Prof Alun Preece.

This study has been tested with Chrome and Firefox, although it should function with most browsers. We require cookies and Javascript to be enabled (these are enabled by default with most browsers). In addition, please disable ad blockers as they may interfere with some of this website's functionality. Please use a laptop or desktop computer to take part in this study.

During the study, we will track button presses and page elements visible on your screen. Tracking is restricted to just the study platform and will not include any activity outside of the study.

Some pages may take upwards of 15 seconds to load. Please be patient. This should only occur on a handful of pages at the start of the study.

Figure B8: Lay-person study interface welcome page.



Consent form

Title of research project: Concept Bottleneck Models lay-person user study

SREC reference and committee: COMSC/Ethics/2023/146

Name of Chief/Principal Investigator: Jack Furby

Participant information sheet

I confirm that I have read the information sheet dated 2/7/2024 version 3 for the above research project.	Please initial box
I confirm that I have understood the information sheet dated 2/7/2024 version 3 for the above research project and that I have had the opportunity to ask questions and that these have been answered satisfactorily.	Please initial box
I understand that my participation is voluntary and I am free to withdraw at any time without giving a reason and without any adverse consequences (e.g. to medical care or legal rights, if relevant).	Please initial box
I understand that my personal information (name and email) will be processed for the purposes explained to me, as set out in the information sheet. I understand that such information will be held in accordance with all applicable data protection legislation and in strict confidence, unless disclosure is required by law or professional obligation. I have been informed of my rights under data protection legislation and how I can raise any concerns.	Please initial box
I understand who will have access to personal information provided, how the data will be stored and what will happen to the data at the end of the research project.	Please initial box
I understand that anonymised excerpts and/or verbatim quotes from my submission may be used as part of the research publication.	Please initial box
I understand how the findings and results of the research project will be written up and published.	Please initial box
I agree to take part in this research project.	Please initial box

Name of participant

Date 27/01/2025

Email

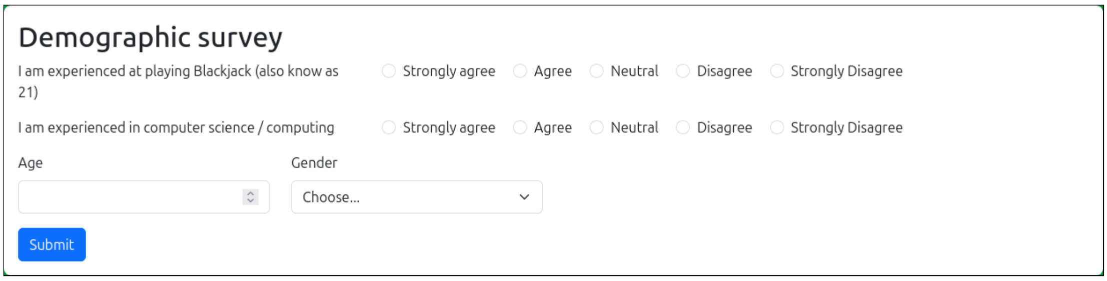
This should be the same email you used to sign up to the study

□ Keep me updated when this research is published

Submit

Thank you for participating in our research you will be given a copy of this consent form to keep.

Figure B9: Lay-person study interface consent page.



Demographic survey

I am experienced at playing Blackjack (also know as 21) ○ Strongly agree ○ Agree ○ Neutral ○ Disagree ○ Strongly Disagree

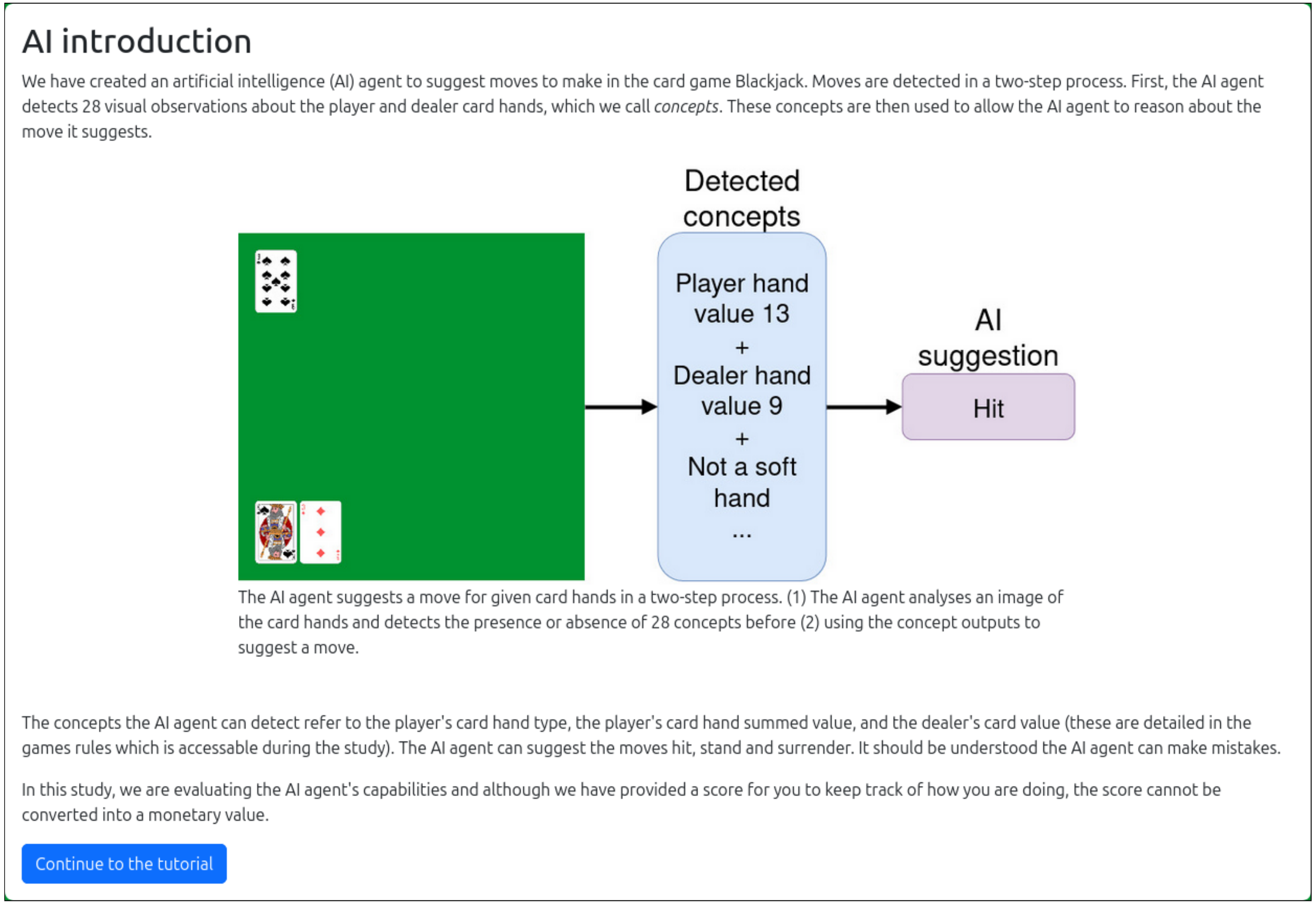
I am experienced in computer science / computing ○ Strongly agree ○ Agree ○ Neutral ○ Disagree ○ Strongly Disagree

Age

Gender Choice...

Submit

Figure B10: Lay-person study interface demographic page.



AI introduction

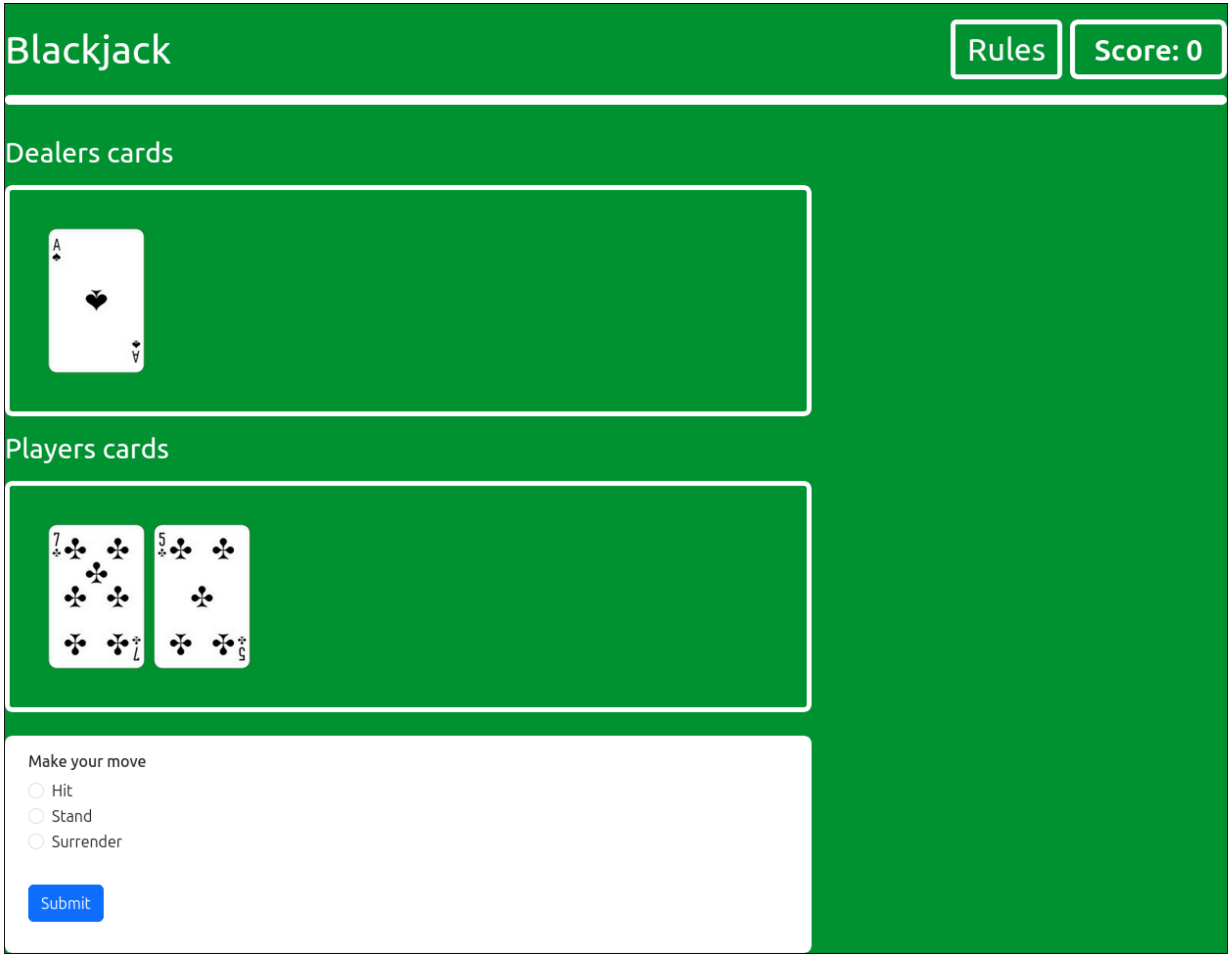
We have created an artificial intelligence (AI) agent to suggest moves to make in the card game Blackjack. Moves are detected in a two-step process. First, the AI agent detects 28 visual observations about the player and dealer card hands, which we call *concepts*. These concepts are then used to allow the AI agent to reason about the move it suggests.

The AI agent suggests a move for given card hands in a two-step process. (1) The AI agent analyses an image of the card hands and detects the presence or absence of 28 concepts before (2) using the concept outputs to suggest a move.

The concepts the AI agent can detect refer to the player's card hand type, the player's card hand summed value, and the dealer's card value (these are detailed in the games rules which is accessable during the study). The AI agent can suggest the moves hit, stand and surrender. It should be understood the AI agent can make mistakes.

In this study, we are evaluating the AI agent's capabilities and although we have provided a score for you to keep track of how you are doing, the score cannot be converted into a monetary value.

Figure B11: Lay-person study interface introduction page.



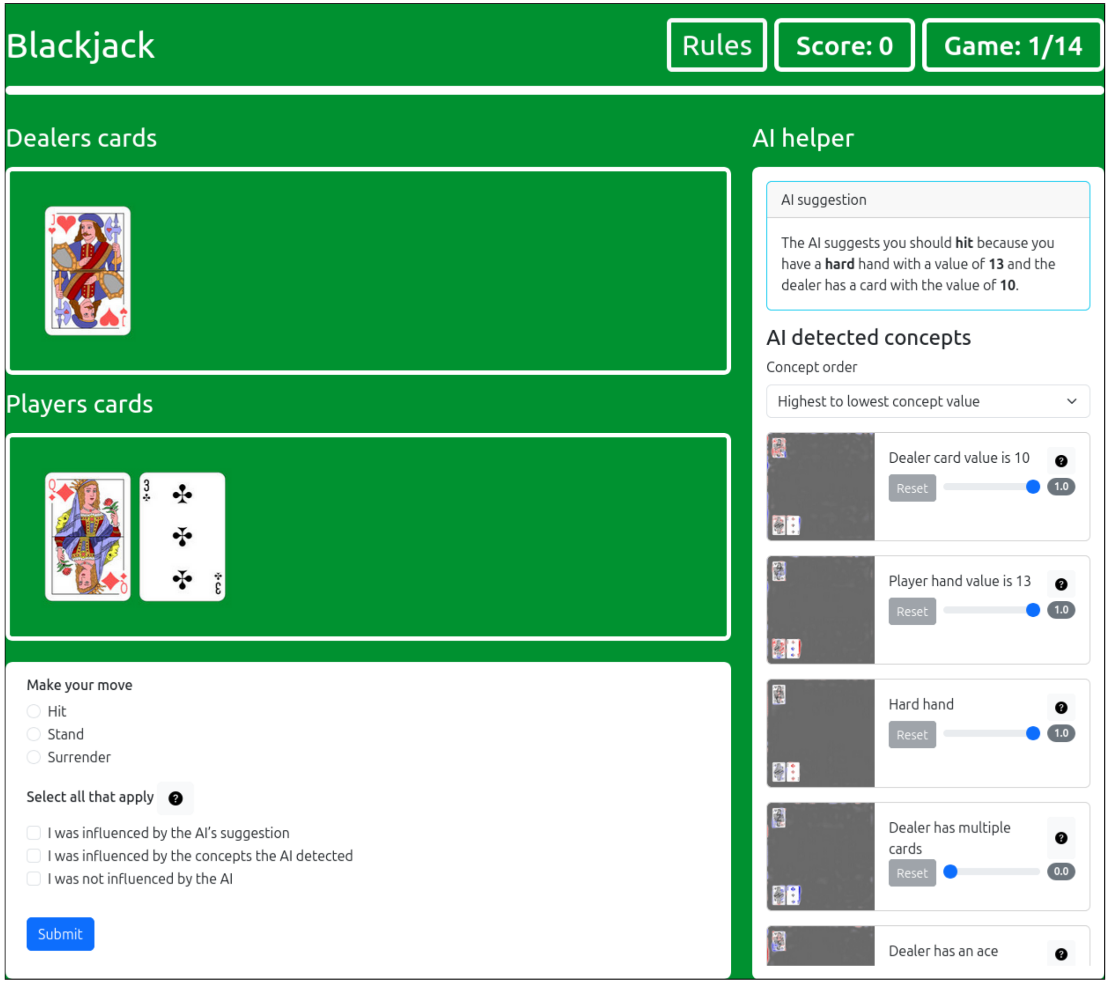
Make your move

○ Hit

○ Stand

○ Surrender

Figure B12: Lay-person study interface main page with AI disabled.



Blackjack

Rules

Score: 0

Game: 1/14

Dealers cards

Players cards

Make your move

○ Hit

○ Stand

○ Surrender

Select all that apply

□ I was influenced by the AI's suggestion

□ I was influenced by the concepts the AI detected

□ I was not influenced by the AI

Submit

AI helper

AI suggestion

The AI suggests you should **hit** because you have a **hard** hand with a value of **13** and the dealer has a card with the value of **10**.

AI detected concepts

Concept order

Highest to lowest concept value

Dealer card value is 10 — Reset — 1.0

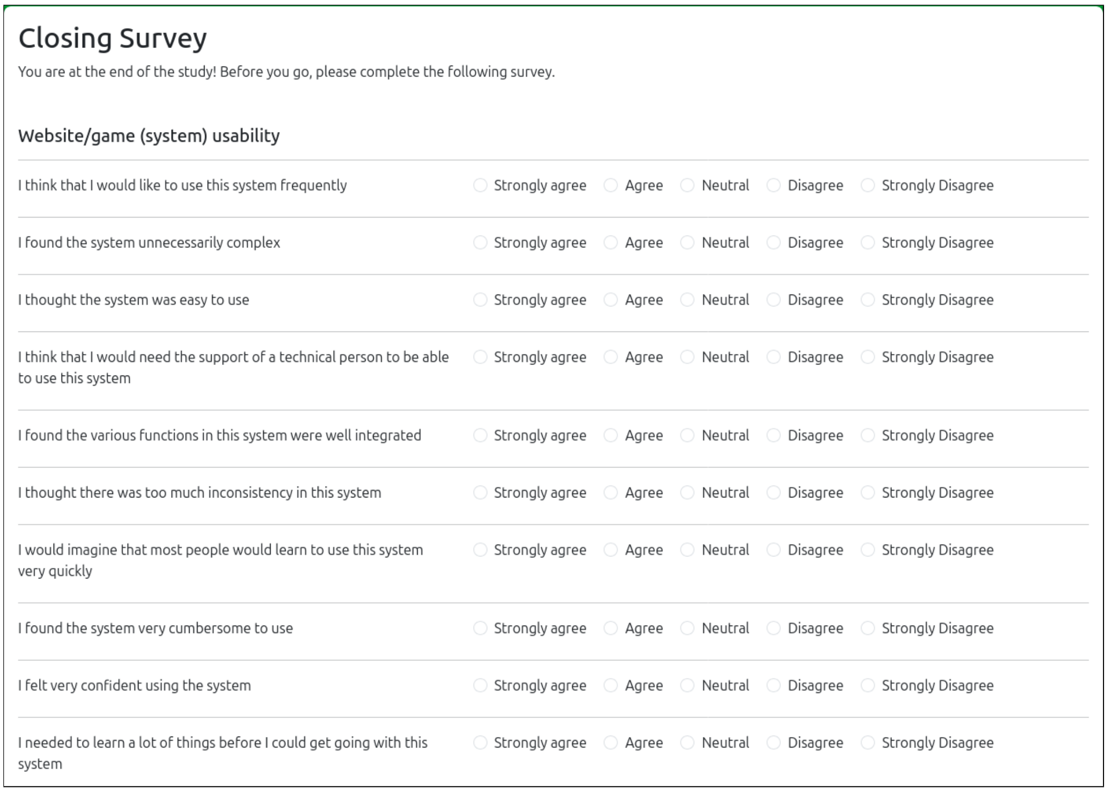
Player hand value is 13 — Reset — 1.0

Hard hand — Reset — 1.0

Dealer has multiple cards — Reset — 0.0

Dealer has an ace

Figure B13: Lay-person study interface main page with AI enabled.



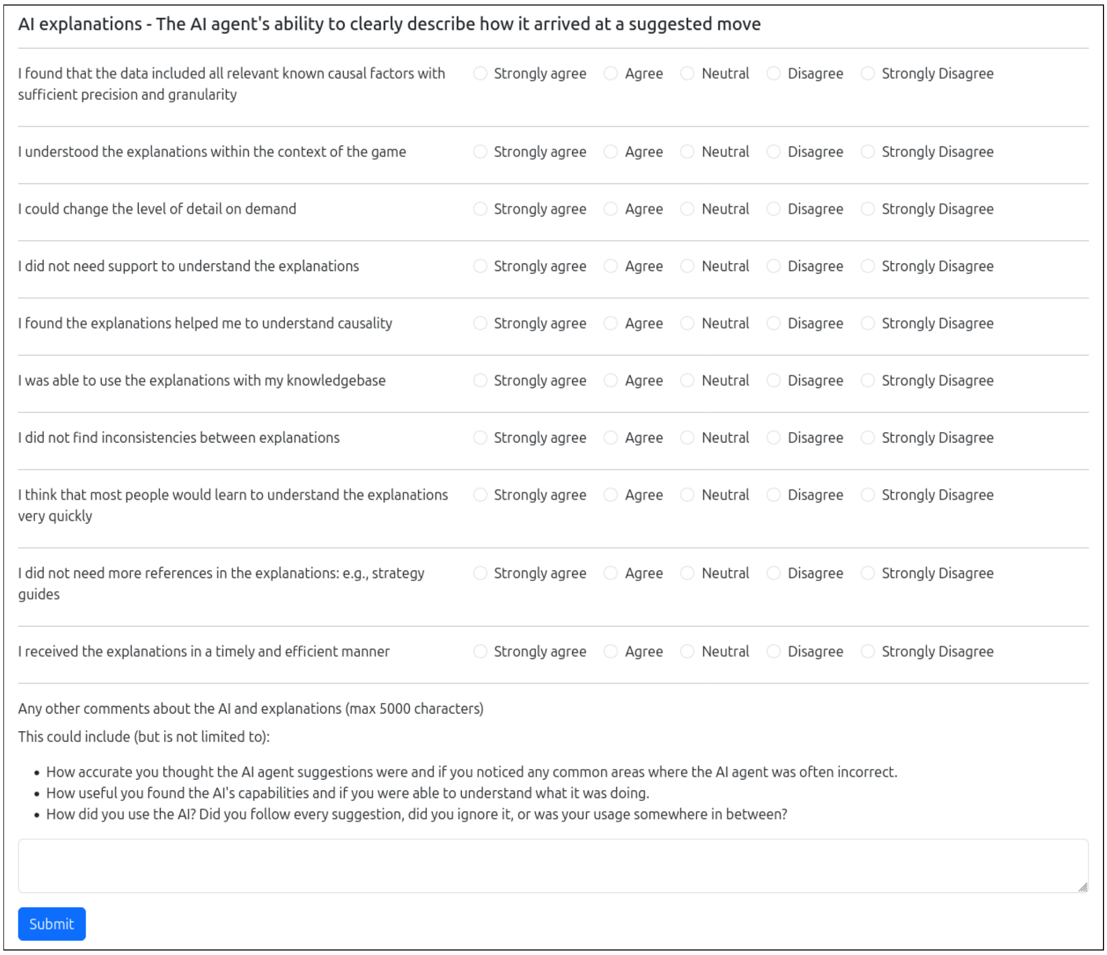
Closing Survey

You are at the end of the study! Before you go, please complete the following survey.

Website/game (system) usability

I think that I would like to use this system frequently	○ Strongly agree	○ Agree	○ Neutral	○ Disagree	○ Strongly Disagree
I found the system unnecessarily complex	○ Strongly agree	○ Agree	○ Neutral	○ Disagree	○ Strongly Disagree
I thought the system was easy to use	○ Strongly agree	○ Agree	○ Neutral	○ Disagree	○ Strongly Disagree
I think that I would need the support of a technical person to be able to use this system	○ Strongly agree	○ Agree	○ Neutral	○ Disagree	○ Strongly Disagree
I found the various functions in this system were well integrated	○ Strongly agree	○ Agree	○ Neutral	○ Disagree	○ Strongly Disagree
I thought there was too much inconsistency in this system	○ Strongly agree	○ Agree	○ Neutral	○ Disagree	○ Strongly Disagree
I would imagine that most people would learn to use this system very quickly	○ Strongly agree	○ Agree	○ Neutral	○ Disagree	○ Strongly Disagree
I found the system very cumbersome to use	○ Strongly agree	○ Agree	○ Neutral	○ Disagree	○ Strongly Disagree
I felt very confident using the system	○ Strongly agree	○ Agree	○ Neutral	○ Disagree	○ Strongly Disagree
I needed to learn a lot of things before I could get going with this system	○ Strongly agree	○ Agree	○ Neutral	○ Disagree	○ Strongly Disagree

Figure B14: First half of lay-person study interface survey page.



AI explanations - The AI agent's ability to clearly describe how it arrived at a suggested move

I found that the data included all relevant known causal factors with sufficient precision and granularity	○ Strongly agree	○ Agree	○ Neutral	○ Disagree	○ Strongly Disagree
I understood the explanations within the context of the game	○ Strongly agree	○ Agree	○ Neutral	○ Disagree	○ Strongly Disagree
I could change the level of detail on demand	○ Strongly agree	○ Agree	○ Neutral	○ Disagree	○ Strongly Disagree
I did not need support to understand the explanations	○ Strongly agree	○ Agree	○ Neutral	○ Disagree	○ Strongly Disagree
I found the explanations helped me to understand causality	○ Strongly agree	○ Agree	○ Neutral	○ Disagree	○ Strongly Disagree
I was able to use the explanations with my knowledgebase	○ Strongly agree	○ Agree	○ Neutral	○ Disagree	○ Strongly Disagree
I did not find inconsistencies between explanations	○ Strongly agree	○ Agree	○ Neutral	○ Disagree	○ Strongly Disagree
I think that most people would learn to understand the explanations very quickly	○ Strongly agree	○ Agree	○ Neutral	○ Disagree	○ Strongly Disagree
I did not need more references in the explanations: e.g., strategy guides	○ Strongly agree	○ Agree	○ Neutral	○ Disagree	○ Strongly Disagree
I received the explanations in a timely and efficient manner	○ Strongly agree	○ Agree	○ Neutral	○ Disagree	○ Strongly Disagree

Any other comments about the AI and explanations (max 5000 characters)

This could include (but is not limited to):

- How accurate you thought the AI agent suggestions were and if you noticed any common areas where the AI agent was often incorrect.
- How useful you found the AI's capabilities and if you were able to understand what it was doing.
- How did you use the AI? Did you follow every suggestion, did you ignore it, or was your usage somewhere in between?

Figure B15: First half of lay-person study interface survey page.