Leveraging Human-Centered Explanations for Model Improvement and Evaluation

Thesis submitted in partial fulfillment
of the requirements for the degree of

Masters of Science
in
Computer Science and Engineering by Research

by

Avani Gupta
2019121004
avani.gupta@research.iiit.ac.in

International Institute of Information Technology
Hyderabad - 500 032, INDIA
November 2023

International Institute of Information Technology
Hyderabad, India

CERTIFICATE

It is certified that the work contained in this thesis, titled ‘Concept-based model improvement and evaluation by Avani Gupta, has been carried out under my supervision and is not submitted elsewhere for a degree.

Date

Adviser: Prof. P J Narayanan

To my Mother Meena Gupta

Acknowledgments

I would like to thank my advisor Prof. P J Narayanan who guided and believed in me. The idea of using ML interpretability-based method in a core graphics problem was out of the wild, but Prof. P J Narayanan allowed me to pursue it. The second idea of making a loss function of the concept based approaches was again out of the box too which happened to click while discussion meeting on first work which he encouraged to pursue. The freedom he gave me and the learning I had in these three years of working with him is precious. He was always inpiring and supportive as an advisor.

This dissertation would not have been possible without the constant support of Saurabh Saini who guided me throughout my work in Masters. I had discussions starting from reading papers to idea formulation, experimentation planning, and paper writing with him and my advisor Prof. P J Narayanan. I learned a lot about research from him. I would thank Pulkit who guided me in the procedures for everything, who had already been through the same stage as I was, being my senior. He also reviewed my paper on CSM and provided valuable feedback which helped me learn the importance of simple things like captions of figures in a research paper.

I would also like to thank Prof. Avinash Sharma and Sai Sagar Jinka with whom I worked on different research thread on 3D computer vision during my second year of this course (2019). That gave me a fair idea of the research over here in IIIT Hyderabad.

I would thank Dhawal, Astitva, and Chandradeep for their suggestions in 3D rendering while creating data for IID experimentation. I thank Dhruv whose support in technical errors helped a lot. I also thank Aaditya for the discussions and suggestions on paper writing. I thank Naren and BVK with whom I had many non-technical and technical discussions on various topics. I also thank Siddhant for reviewing my thesis and providing valuable suggestions.

I thank my family specially my parents, my brother Amber and my cousin Ankita who understood me and always stood with me. Finally, I thank my friends Jalees, Bhuvanesh, and Saanika who were always there in any situation.

Abstract

Neural Networks are known to be black box models which are explained by interpretability based approaches. ML Interpretability methods to explain the complex Neural Networks reasoning in human understandable form. Humans think in abstract concepts like color, texture, shapes, etc. Explaining black box models in these simple concepts aids human understanding of models leading to more transparency, reliability and proactive identification of risks/biases in the models. The recent interpretability based methods have started using concepts for explaining complex models in simple human understandable terms. In computer vision, concepts are defined as a set of images having a human-understandable meaning associated with them (eg. striped images form *stripiness* concept). Current concept based interpretability methods use the encoding of concepts in an intermediate layer of the model to define concept representations. They further use these representations by perturbing model activations in the intermediate layer and gauging its sensitivity to the perturbations. We use these CAV based sensitivity calculations for evaluation of desired properties by using the very definitions from the problem as concepts. We further extend the use of CAVs from post-hoc analysis to ante-hoc training via our novel concept loss and distillation paradigm. Concretely, we present a concept sensitivity based method to measure disentanglement in an ill-posed ambiguous problem of Intrinsic Image Decomposition (IID) followed by a novel method for concept based training of DNNs which utilizes conceptual knowledge from large pre-trained models in a distillation paradigm.

IID involves decomposing an image into its constituent Reflectance and Shading components, which are illumination-invariant and albedo-invariant, respectively. We use this definition of IID to define our concepts and propose to use the sensitivity scores to directly measure the alignment of models predictions with the definition. For this, we measure the illumination invariance of Reflectance prediction and albedo invariance of Shading prediction by gazing model's sensitivity to relevant concepts. We thus define the evaluation of IID in abstract human centered concepts. We define Concept Sensitivity Metric (CSM) to measure the disentanglement of Reflectance Shading in the models predictions for evaluating IID methods. We evaluate and interpret three recent IID methods on our synthetic benchmark of controlled albedo and illumination invariance sets. We also compare our metric with existing IID evaluation metrics on both natural and synthetic scenes and report our observations. Our metric not only surpasses several limitations of the existing metrics but is also consistent in both synthetic and real-world datasets. The concept based interpretability methods are post-hoc (after training) and can be used for analyzing the models. We aim to use the feedback provided by these post-hoc methods to train the model for further improvements in an ante-hoc manner. For this we propose a novel concept loss based on the CAV sensitivity. We argue that CAV learning in same model is not efficient and propose to use a separate model having knowledge of concepts in a knowledge distillation paradigm. We present Concept Distillation: a novel method for concept sensitive training of Deep Neural Networks. Concept Distillation can be used to sensitize or desensitize the student model towards user desired concepts. We show applications of our concept-sensitivity based training in debiasing in classification problems and prior induction in IID. We also introduce the TextureMNIST dataset to evaluate the presence of complex texture biases. Our concept-sensitive training can improve model interpretability, reduce biases, and induce prior knowledge.

Contents

List of Figures

Figure Page

List of Tables

Table Page

Chapter 1

Introduction

With the success of Deep Learning, Deep Neural Networks (DNNs) are being used in various fields like Computer Vision, Natural Language Processing, Speech Recognition, etc. [44]. They had surpassed human-level performance in several tasks like playing Go (alphaGo [164]), image recognition (Resnet50 [70]), and speech recognition (Microsoft's Deep Speech 2 [8]) long time back. While DNNs have been successful in extracting patterns from data and learning, they are considered as black-box due to the complexity/ambiguity of the functions approximated by them [123]. The complexity of DNNs makes it difficult to understand how they arrive at their predictions or decisions, making them appear black-box or incomprehensible to human understanding. The focus of recent research has shifted from models merely trained to optimize accuracy to *Human Interpretable* models. Explainable Artificial Intelligence (XAI) [66] techniques open the *black-box* of model decisions by providing insights into ”why” the model made a particular prediction. The field of XAI has grown tremendously in the past 4-5 years with a multitude of XAI techniques [47, 112, 155, 184, 199]. There are various spurious correlations, clever-hans, and biases induced in the model due to data or training procedures. This hampers human trust in these models and their applicability in real-world deployment. There is a need for accountable and transparent models which are not only accurate but fair and unbiased. ML interpretability provides insights into the model's hidden biases and unintended correlations. These insights can be used to improve the model by debiasing (or biasing) the model for an intended cause. With the shift from black box models to explainable models, a shift from explainable models to *explanation driven models* is needed [55, 189].

Out of many XAI-based approaches, concept based approaches are the most human-interpretable. Concepts are linked to their origin in Neuro-science literature, where they are defined as ideas derived or inferred from facts [45], and they hold the human's mental world together [127]. It has been observed that humans learn by creating concept models of things around them, helping them gather generalized knowledge [93]. In computer science, *Concepts* are defined as higher level human understandable units [78] inherent to an entire class or the task itself, e.g., image ‘stripe-ness’ is a vital concept for all samples in the zebra classification task. Beyond textures and colors, concept-based explanations are also quite useful in interpreting hard-to-formulate class attributes like gender, age, etc.

Methods using concepts represent them in the model's activation space [16, 51, 80]. Concept Activation Vector (CAVs) [80] are the most popular concept based representation method. CAV is typically learned in model activation space as a decision boundary separating the concept of interest and its negative counterpart (or a neutral concept). By analyzing the importance given by the model to various concepts, they are used to get insights into what was learned and used for prediction increasing transparency. We now provide a brief overview of the terms throughout the thesis.

1.1 ML Interpretability

Machine Learning (ML) interpretability methods aim to open the black box of the model by explaining their predictions in a human-interpretable way [46]. Interpretability methods provide valuable insights into the inner workings of ML models and can be used to identify biases or errors in the model. They can also help to build trust and transparency with end-users, such as doctors, patients, or customers.

1.1.1 How ML interpretability differs from explainability

ML interpretability and explainability both aim to provide insights into the decision-making process of ML models, but they approach this from different perspectives. Interpretability focuses on ”human understandability” while explainability focuses on mere ”justification or rationale behind a prediction” [46, 122]. In other words, interpretability is concerned with how easily a human can understand the model's predictions, while explainability focuses on explaining and justifying predictions regardless of human understandability [46]. Despite their subtle differences, interpretability and explainability are often used interchangeably and are commonly referred to as eXplanable Artificial Intelligence (XAI).

In this thesis, we restrict our discussion to interpretability methods in Computer Vision.

1.1.2 Local vs. Global Interpretability

Based on the scope of explanation, interpretability methods can be divided into two categories: Global and Local. Global interpretability methods focus on model interpretations that are true for the entire class. In contrast, Local methods focus on explanations of a single data point that are true for the specific sample and its neighbors [80]. Existing methods can fall into either or both of the above categorizations. Local methods include pixel attribution methods like [1, 143, 157, 165] while global methods include [60, 80], etc. Some methods like SHapley Additive exPlanations (SHAP) [118], LIME [137], and Integrated Gradients [174] can provide both local and global interpretations.

1.1.3 Major ML interpretability Methods

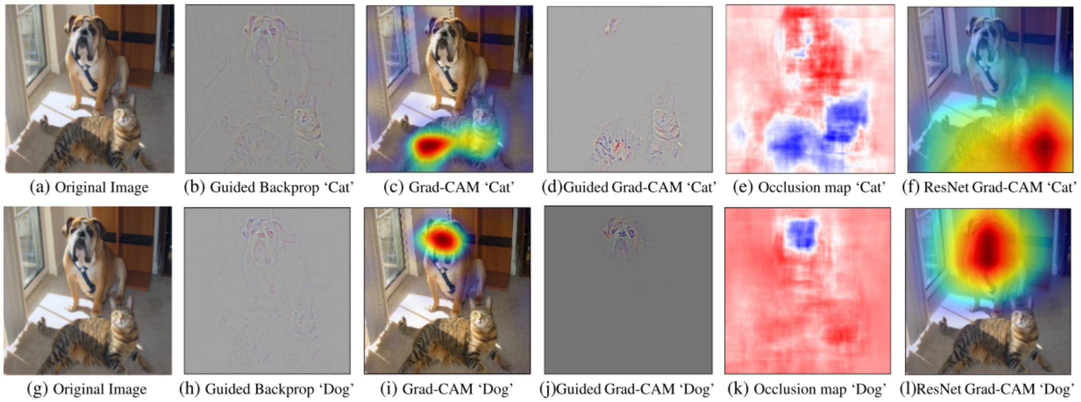
Interpretability methods in Computer Vision can be broadly classified as methods visualizing the importance given to pixels in models input space [1, 143, 157] or quantifying the importance given to concepts [13, 26, 32, 80, 151] (also discusses this categorization [156]).

1.1.3.1 Pixel attribution methods

Pixel attribution methods attribute importance scores given by the model to individual pixels in the given input image [1, 143, 157, 165]. These attributions can then be understood by humans and looked upon to check which regions the model focuses on. GradCAM, or Gradient-weighted Class Activation Mapping [157], is a popular pixel attribution method that generates a heat map indicating which regions of an image contributed most to the model's prediction. It works by calculating the gradients of the output class score with respect to the feature maps of the last convolutional layer of the model. These gradients are then used to weight the feature maps, and the resulting weighted feature maps are averaged to generate the heat map. In addition to GradCAM, there are several other pixel attribution methods, such as Guided Backpropagation [168], Integrated Gradients [174], and SmoothGrad [165], each with its own strengths and weaknesses. Choosing the right pixel attribution method depends on the specific task and the model's characteristics. An example of some pixel attribution method visualizations is shown in Fig. 1.1

1.1.3.2 Concept attribution methods

We discuss a few concept attribution methods and refer readers to [67] for details on more existing Concept attribution methods.



(a) Original Image (b) Guided Backprop 'Cat' (c) Grad-CAM 'Cat' (d)Guided Grad-CAM 'Cat' (e) Occlusion map 'Cat' (f) ResNet Grad-CAM 'Cat'

(g) Original Image (h) Guided Backprop 'Dog' (i) Grad-CAM 'Dog' (j)Guided Grad-CAM 'Dog' (k) Occlusion map 'Dog' (l)ResNet Grad-CAM 'Dog'

Figure 1.1: Comparison of Pixel attribution methods. Source: [157]

Hidden Units Concept images alignment based Network Dissection [16] uses user-provided concept sets and checks their alignment with individual hidden units at each layer of CNN. Since DNNs are expected to learn partially non-local representations in denser layers, concepts can align with a combination of several hidden units. However, to assess disentanglement, they focus on measuring the alignment of concepts with single units. This method hypothesizes that the interpretability of units is equivalent to their random linear combination. They evaluate every individual convolutional unit in CNN as a solution to binary segmentation tasks for each visual concept. They pass in all concept sets from the model and determine the distribution of concept activations a_k for each convolutional unit k. They then determine the top quantile level T_k for each unit k such that $P(a_k > T_k) = 0.0005$ over every spatial location of the activation map in the concept set. This is followed by a selection of regions exceeding the threshold T_k and evaluating segmentations with every concept c in the concept set by computing the intersection over union ($IoU_{k,c}$) of the above-selected regions with input concept annotation masks. A concept detector is reported if $IoU_{k,c}$ is greater than a certain threshold. The interpretability of a layer is quantified by the number of unique concepts aligned with its units (*unique detectors*). Since IoU is an objective measure (not relative), it enables comparison of interpretability across networks.

Concept Embeddings Vectors Based Fong and Vedaldi [51] proposed Net2Vec, which aligns the concepts with CNN filters. They first collect the pre-trained model's activations for a concept dataset (probe) and then learn the weights to recognize the concept in various semantic tasks. These weights are interpreted as concept embeddings and analyzed to gain insights into how concepts are encoded in the network.

Concept Activation Vectors (CAV) CAVs were first proposed by Kim et al. [80] as a Global interpretability technique for model interpretation. CAVs estimate the sensitivity of a neural network to a given concept. A binary linear classifier is used to separate concept examples from random examples in a specific layer's activation space. The CAV is then estimated as the normal to the linear decision boundary separating concepts. The working of CAVs is explained in Fig. 1.2.

For a given concept C in specific layer l, CAV is denoted by $\boldsymbol{v_C^l}$. A *sensitivity* score $S_{C,l}$ of model towards concept C is calculated as the directional derivative of the loss term ∇L (or logit) of the class sample's activation $f_l(\boldsymbol{x})$ in the direction of $\boldsymbol{v_C^l}$:

$$S_{C,l}(\boldsymbol{x}) = \nabla L\left(f_l(\boldsymbol{x})\right) \cdot \boldsymbol{v}_C^l$$

The sensitivity scores are averaged across all class samples to measure class sensitivity and across all classes for overall model sensitivity.

In summary, CAVs provide a useful tool for understanding the impact of concepts on neural network models, which can aid in addressing biases and improving performance.

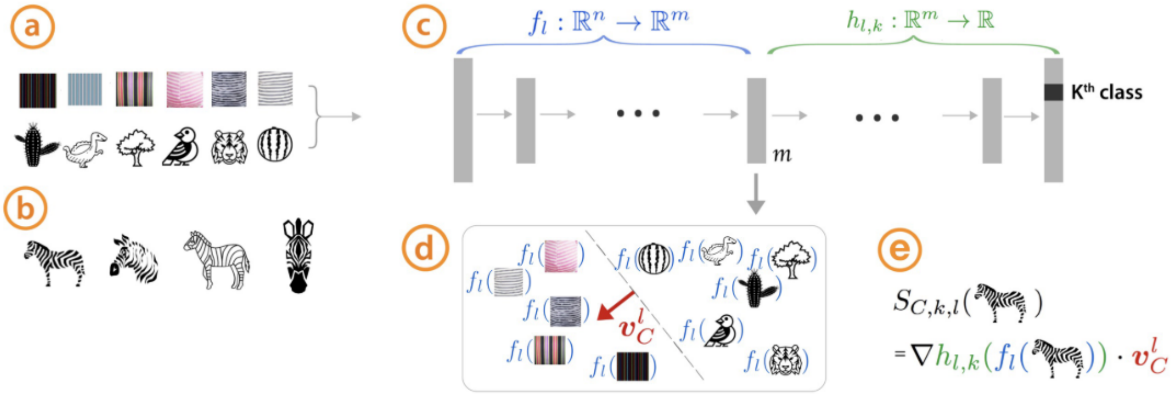

a
b
c
$f_l : \mathbb{R}^n \rightarrow \mathbb{R}^m$
$h_{l,k} : \mathbb{R}^m \rightarrow \mathbb{R}$
Kth class
m
d
v_C^l
e
$S_{C,k,l}$
$= \nabla h_{l,k}(f_l$ $)) \cdot v_C^l$

Figure 1.2: CAV working: Given user-defined Concept set C images (striped here), random images C' (a), images of class of interest (zebras) (b), and a trained network (c); Example images of C and C' are passed from a pre-trained model to get its activations in layer l. A linear classifier is trained to distinguish between activations of C, $f_l(C_x)$ and C' $f_l(C_y)$. CAV vector v_C^l is taken as the normal to the hyperplane separating the two concepts' activations. For the class of interest (zebras), Kim et al. [80] use the directional derivative $S_{C,l(x)}$ to quantify conceptual sensitivity. Source: [80]

1.2 Bias in ML models

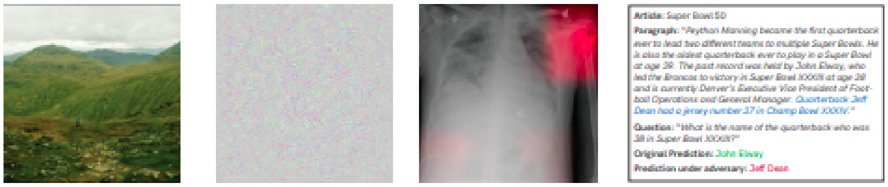
Bias in ML refers to a systematic error or deviation in a model's predictions that can arise due to the way the model was trained [121]. This can occur, for example, if the training data is not representative of the population that the model will be applied to or if the model is designed to favor certain outcomes over others. Bias can manifest itself in various ways, such as the under-representation of certain groups or the overestimation of certain features. DNNs can learn various spurious correlations which are wrong. They also show clever-hans and shortcut learning phenomena [58].

1.2.1 Clever Hans

Clever Hans was a horse that was claimed to perform complex intellectual tasks as well as arithmetic calculations. A formal investigation was done over it by psychologist Oskar Pfungst. It was found that the horse was not performing the arithmetic or intellectual tasks but rather watching his trainer's reactions (body language, etc.), who was unaware of providing such cues. This phenomenon was named as *Clever Hans effect* [43]. DNNs can also exhibit such phenomena, where they use unintended cues for prediction while achieving good test performance [96].

1.2.2 Shortcut Learning in Neural Networks

The DNNs learn various shortcuts for prediction [58], which are not intended to be learned (Fig. 1.3). These shortcuts work well on a specific data distribution (similar to a training set) but fail on a different



Article: Super Bowl 50

Paragraph: *"Peyton Manning became the first quarterback ever to lead two different teams to multiple Super Bowls. He is also the oldest quarterback ever to play in a Super Bowl at age 39. The past record was held by John Elway, who led the Broncos to victory in Super Bowl XXXIII at age 38 and is currently Denver's Executive Vice President of Football Operations and General Manager. Quarterback Jeff Dean had a jersey number 37 in Champ Bowl XXXIV."*

Question: *"What is the name of the quarterback who was 38 in Super Bowl XXXIII?"*

Original Prediction: John Elway

Prediction under adversary: Jeff Dean

Task for DNN	Caption image	Recognise object	Recognise pneumonia	Answer question
Problem	Describes green hillside as grazing sheep	Hallucinates teapot if certain patterns are present	Fails on scans from new hospitals	Changes answer if irrelevant information is added
Shortcut	Uses background to recognise primary object	Uses features irrecognisable to humans	Looks at hospital token, not lung	Only looks at last sentence and ignores context

Figure 1.3: Shortcut Learning Examples: DL models learn a lot of shortcuts as shown in the above four tasks. Source:[58]

distribution. These include learning under covariate shift, anti-causal learning, dataset bias, tank legend, and clever-hans effect. Shortcuts can arise from biases in the dataset (the training set has a particular distribution that is biased) or biased decision rules (spurious correlations learned by the model (like clever-hans)) [58]. *Spurious Correlations* are the faulty correlations learned by ML models that are incorrect and shall not be used.

Distribution (ood) datasets can reveal shortcut learning in models by providing samples with different distributions from the training set. In contrast, Identically Distributed (iid) datasets have identical distributions to the training set and are typically used as validation sets. While iid datasets can detect overfitting, they may not detect shortcut learning issues. For instance, consider the star-moon classification dataset shown in Fig. 1.4.

1.2.3 Debiasing Models

Debiasing methods are used to address biases in machine learning models, which can lead to unfair or discriminatory outcomes. These methods typically involve using various techniques, such as adversarial training [98], counterfactuals [21], bias swapping [81], etc., to remove different types of biases from the model. Adversarial training involves training the model on adversarial examples, which are designed to be similar to the original data but with slight perturbations that cause the model to make incorrect predictions. Counterfactuals involve generating hypothetical scenarios that could have led to different outcomes and using them to train the model to be more robust to biases. Bias swapping involves swapping data samples from different subgroups to ensure that the model does not learn to associate certain features with certain outcomes [81, 110].

To evaluate the effectiveness of debiasing methods, models are typically trained and validated on independent and identically distributed (iid) datasets and tested on out-of-distribution (ood) datasets [58]. This helps ensure that the model is not just memorizing the training data and is able to generalize to new and unseen data.

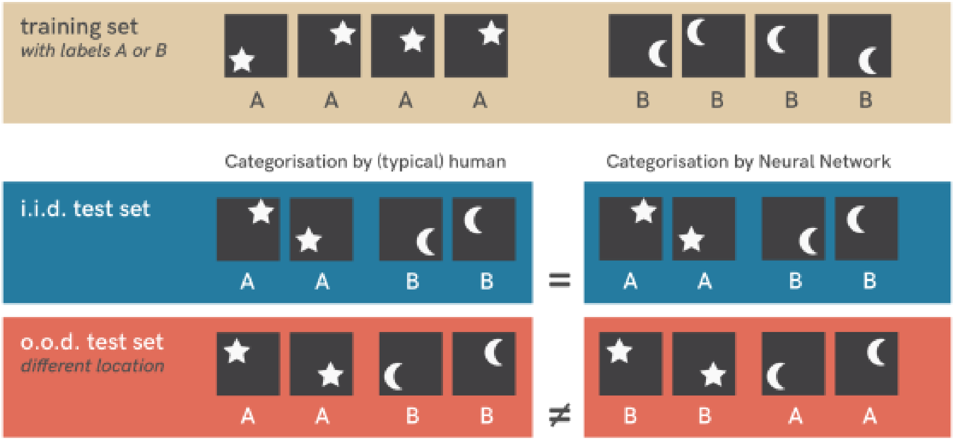


Figure 1.4: iid vs ood distributions: Star vs Moon classification example. The training set has stars in the top right and bottom left corners while the moon is in the top left and bottom right corners. The model takes a shortcut and predicts based on position not learning the actual shape of the object. This phenomenon of shortcut learning cannot be detected in an indentically distributed (i.i.d) dataset (set with identical distribution to training set) wherein the model and human will have the same categorization but can be detected in Out of Distribution (o.o.d) set wherein the model will perform poorly due to shortcut learning and its predictions won't match that of a human. Source:[58]

1.3 Various Learning paradigms using XAI

1.3.1 Explanation Guided Learning (EGL)

Explanation Guided Learning (EGL) [55] is learning using Explanations to train or improvise the model. EGL can have one or both of the following aims: (i) improving the interpretability, (ii) improving the model's performance. The primary goal of EGL is to learn a model that can make accurate predictions while generating meaningful explanations for its predictions. EGL typically involves jointly optimizing the model prediction and the explanation by incorporating three key terms in the objective function: task supervision, explanation supervision, and explanation regularization. Its objective function can be written as follows:

$$\min \underbrace{\mathcal{L}_{\text{Pred}}(f(X),Y)}_{\text{task supervision}} + \underbrace{\alpha\mathcal{L}_{\text{Exp}}(g(f,\langle X,Y\rangle),\hat{M})}_{\text{explanation supervision}} + \underbrace{\beta\Omega(g(f,\langle X,Y\rangle))}_{\text{explanation regularization}}$$

Where $\hat{M}$ incorporates the 'right' explanation provided via human annotation. The task supervision term guides the model to learn task-specific information, while the explanation supervision term supervises the model explanation to ensure consistency with ground truth. The explanation regularization term helps avoid overfitting and encourages the model to generate interpretable and meaningful expla-

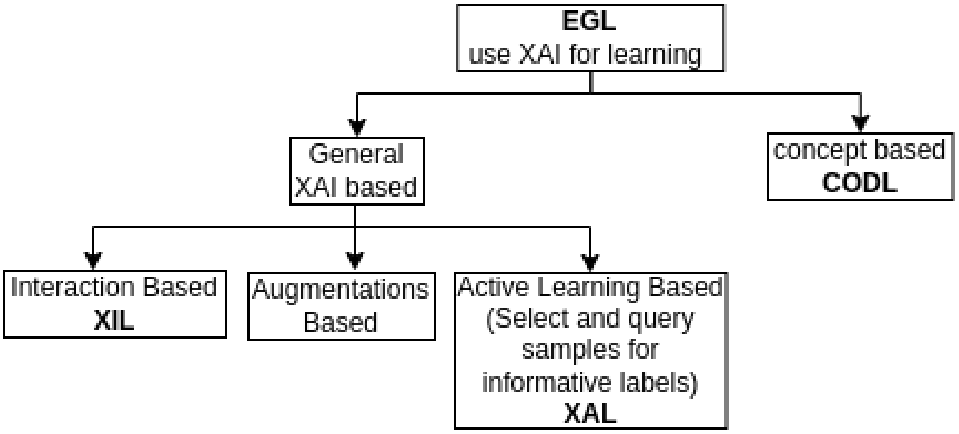


Figure 1.5: Broad Categorization of Explanation Guided Learning (EGL) methods.

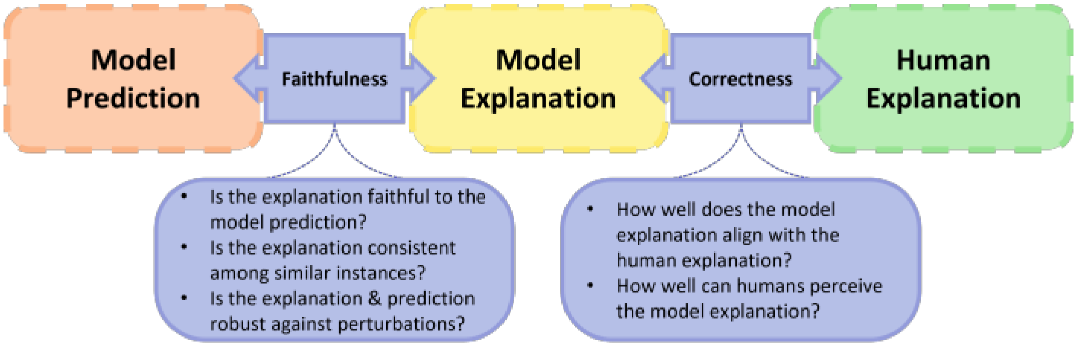

Model
Prediction
Faithfulness
Model
Explanation
Correctness
Human
Explanation
• Is the explanation faithful to the model prediction?
• Is the explanation consistent among similar instances?
• Is the explanation & prediction robust against perturbations?
• How well does the model explanation align with the human explanation?
• How well can humans perceive the model explanation?

Figure 1.6: EGL performance evaluation. Source: [55]

nations. We divide EGL methods into General XAI based, which includes methods not using concepts and methods using concepts. Furthermore, General XAI-based methods can be divided into methods that use human interactions and feedback for improving the model (Explanatory Interactive Learning [178]), methods that augment data, model, loss function, etc. (Augmentation based []) for providing feedback to the model, and methods involving smart querying for labeling (Explanatory Active Learning (XAL)). EGL is an umbrella that includes Explanatory Interactive Learning (XIL) [178], Explanatory Active Learning (XAL) [59], and Concept Oriented Deep Learning (CODL) [31] approaches as shown in Fig. 1.5. We describe each of the categories given in Fig. 1.5 below. Since we focus on concept-based methods, we describe EGL as methods not using concepts and methods using concepts.

1.3.2 General XAI based (non-concept based)

The methods which use generic XAI but not concepts can be categorized in terms of using interactions with an expert (Interaction Based) for guidance, augmenting some aspects of data/model/loss function (Augmentations Based), or using active learning (Active Learning Based).

```
$f \leftarrow \textsc{Fit}(\mathcal{A})$
repeat
    $X \leftarrow \textsc{Select}(f, \mathcal{N})$
    $\hat{y} \leftarrow f(X)$
    $\hat{E} \leftarrow \textsc{Explain}(f, X, \hat{y})$
    Present $X$, $\hat{y}$, and $\hat{E}$ to the user
    $\overline{y}, \overline{C} \leftarrow \textsc{Obtain}(X, \hat{y}, \hat{E})$
    $\mathcal{A} \leftarrow \mathcal{A} \cup \{(X, \overline{y}, \overline{C})\}$
    $f \leftarrow \textsc{Revise}(\mathcal{A})$
    $\mathcal{N} \leftarrow \mathcal{N} \setminus \{X\}$
until budget $T$ is exhausted or $f$ is good enough
return $f$
```

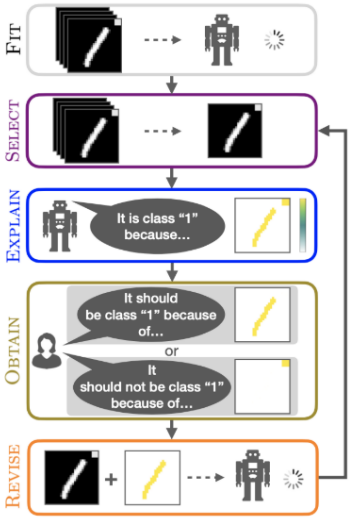

FIT
SELECT
EXPLAIN
It is class "1" because...
OBTAIN
It should be class "1" because of...
or
It should not be class "1" because of...
REVISE

Figure 1.7: Keys Steps of XIL approaches: A typical XIL method consists of four steps comprising Select, Explain, Obtain, and Revise. Source: [53]

Interactions based: Explanatory Interactive Learning (XIL) Explanatory and Interactive Learning (XIL) [178] is a framework for machine learning models to be inspected, interacted with, and revised to ensure that their learned knowledge aligns with human knowledge. XIL methods aim to mitigate learning shortcuts and provide explanations for the model's decision-making process, enabling users to interact with the model and provide feedback to improve its performance. Friedrich et al. [53] provides a typology of existing XIL methods and a generalized XIL algorithm consisting of four essential steps of Select, Explain, Obtain, and Revise as shown in Fig. 1.7. The XIL algorithm takes in a set of annotated examples (A), a set of non-annotated examples (N), and an iteration budget (T). The Select module is responsible for selecting samples from N to present to the teacher. The Explain module provides insights to the teacher about the model's reasoning process. The Obtain module allows the teacher to observe whether the model's prediction is correct or incorrect and to provide corrective feedback. Finally, the Revise module uses the corrective feedback to update the model's behavior towards the user.

Augmentation Based Weber et al. [189] classify XAI-based model improvement based on augmentation made, which can be over data, loss, gradient, model, etc., as shown in Fig. 1.8.

- **Augmenting the data** It is one of the most common methods of XAI. It involves using XAI to find the spurious or undesirable effects in the model and removing them by augmenting the training data used for the model.
- **Augmenting the Intermediate Features** Explanations offer valuable insights into the importance of individual features in a machine-learning model. This information can be used to modify the intermediate features of the model by scaling, masking, or transforming them.

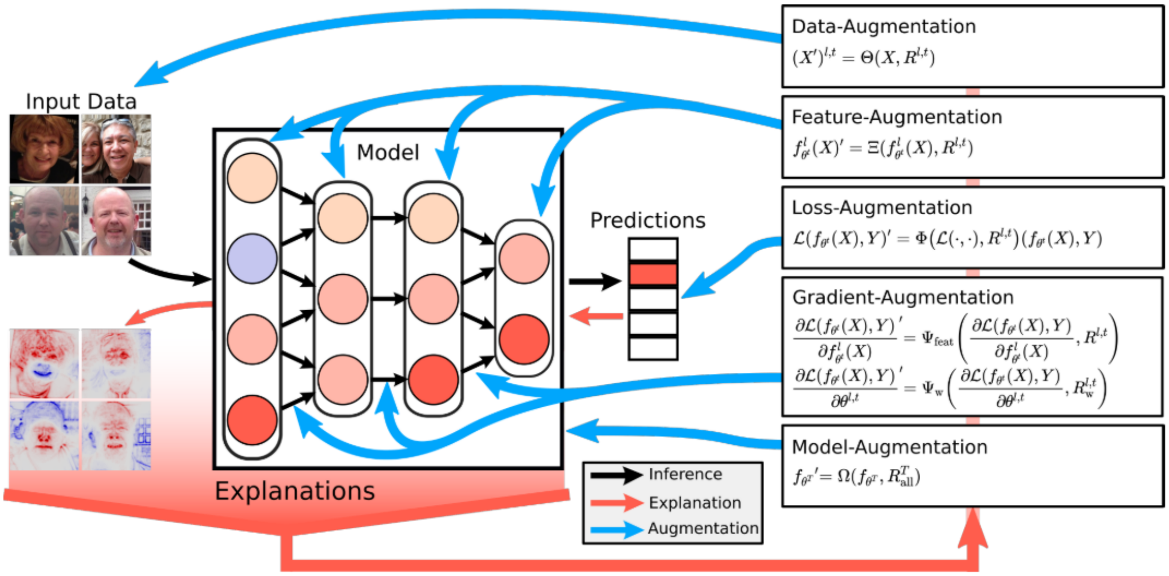

Input Data
Model
Predictions
Explanations
Data-Augmentation
$(X')^{l,t} = \Theta(X, R^{l,t})$
Feature-Augmentation
$f^l_{\theta^t}(X)' = \Xi(f^l_{\theta^t}(X), R^{l,t})$
Loss-Augmentation
$\mathcal{L}(f_{\theta^t}(X), Y)' = \Phi\big(\mathcal{L}(\cdot,\cdot), R^{l,t}\big)(f_{\theta^t}(X), Y)$
Gradient-Augmentation
$\frac{\partial \mathcal{L}(f_{\theta^t}(X), Y)'}{\partial f^l_{\theta^t}(X)} = \Psi_{\mathrm{feat}}\left(\frac{\partial \mathcal{L}(f_{\theta^t}(X), Y)}{\partial f^l_{\theta^t}(X)}, R^{l,t}\right)$
$\frac{\partial \mathcal{L}(f_{\theta^t}(X), Y)'}{\partial \theta^{l,t}} = \Psi_{\mathrm{w}}\left(\frac{\partial \mathcal{L}(f_{\theta^t}(X), Y)}{\partial \theta^{l,t}}, R^{l,t}_{\mathrm{w}}\right)$
Model-Augmentation
$f_{\theta^T}{}' = \Omega(f_{\theta^T}, R^T_{\mathrm{all}})$
Inference
Explanation
Augmentation

Figure 1.8: Categorisation of EGL based on Augmentation method used. Source: [189]

- **Augmenting the Loss Function** The behavior of a machine learning model is determined by its loss function. Therefore, by augmenting the loss function with information from explanations, it is possible to guide the model toward desired behaviors. Explanations can be used as feedback to specify the desired behavior and improve the model's overall performance.
- **Augmenting the gradients** Explanations can also be used to augment gradients during the backward pass of the model. This is because the information about feature importance provided by explanations can be applied to the calculation of parameter updates using the chain rule.
- **Augmenting the model** XAI can be used to prune or quantify the model to reduce the amount of storage space required by the parameters.

Typically, these XAI-based augmentation techniques are applied one at a time. However, in theory, multiple augmentations targeting the same model property could be applied at the same time, altering different components of the training process. We introduce the categorization above while discussing the works for each of these categorizations in subsection 2.2.1.

Active Learning based: Explanatory Active Learning (XAL) Explanatory Active Learning (XAL) is an emerging learning paradigm that combines active learning with explanations [59]. Active Learning (AL) is a learning paradigm that allows a learning algorithm to intelligently select instances to be labeled, which can lead to high performance with much less training data compared to traditional supervised learning approaches [136]. AL has become increasingly important in modern machine learning, where labeled data can be expensive and time-consuming to obtain.

Active learning reduces labeling workload by intelligently selecting instances to query a machine teacher for labels. However, the human-AI interface remains minimal and opaque, hindering the development of teacher-friendly interfaces for AL algorithms. XAL aims to introduce techniques from the field of XAI into AL settings to make AI explanations a core element of the human-AI interface for teaching machines. In this paradigm, the teacher should be able to understand the reasoning underlying the model's mistakes during the learning process. With further training of students, the teacher should be able to recognize, trust and feel confident about their teaching outcome. The teacher here can be a human or a large model.

1.3.3 Concept Oriented Deep Learning (CODL)

Concept Oriented Deep Learning (CODL) [31] uses concept-level supervision for models to improve model interpretability and performance. The major aspects of CODL, as introduced by Chang [31], include concept graphs, concept representations, concept exemplars, and concept representation learning systems supporting incremental and continual learning. CODL leverages a common or background knowledge base, such as Microsoft Concept Graph, for the framework of conceptual understanding. By focusing on learning and using concept representations and exemplars, CODL is able to address the major limitations of deep learning, including interpretability, transferability, contextual adaptation, and the requirement for a large amount of labeled training data. Our proposed Concept Distillation approach is a type of CODL approach. Apart from de-biasing the models for spurious correlations or ensuring fairness and transparency, XAI can be used for prior knowledge induction in the models in cases where the dataset is limited or some prior knowledge is known apriori, which is to be respected by the model.

1.4 Prior Knowledge Induction

Prior knowledge refers to knowledge about a task that is known beforehand and can be utilized to perform that task [18]. This knowledge may come from domain experts or established facts/rules relevant to the task at hand. Leveraging this knowledge in a model can be facilitated by incorporating explanations and ground truths provided by domain experts [17, 18].

In their review, [17, 18] examine methods that utilize prior knowledge to either increase explainability or integrate it into a model via explainability techniques. These methods can be particularly useful for tasks requiring domain-specific knowledge for accurate predictions or decision-making.

1.5 Knowledge Distillation

Knowledge Distillation [71] involves the transfer of knowledge from a large and complex model, known as the teacher model, to a smaller and simpler model, known as the student model. This is done by training the student model to mimic the output of the teacher model. The teacher model provides ”soft labels,” which are probability distributions over the possible outputs rather than just the single most likely output. This allows the student model to learn from the teacher model's knowledge and

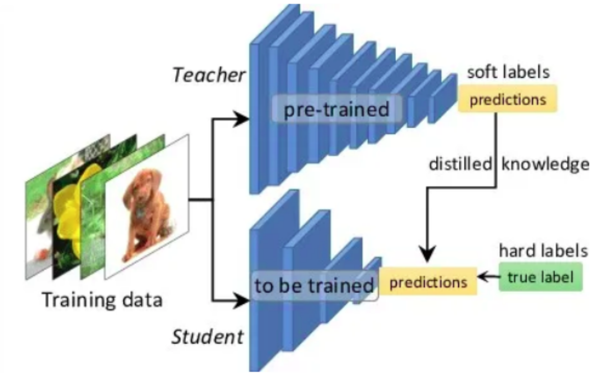


Figure 1.9: Traditional Knowledge Distillation: involves a soft loss coming from teacher's predictions apart from conventional GT loss coming from hard labels. Source: [2]

gain a better understanding of the problem being solved. Figure 1.9 illustrates the process of traditional knowledge distillation, where the student model is trained to predict both the soft labels from the teacher model and the true labels of the dataset. The objective of such training is to minimize the difference between the output of the teacher model and the student model while also minimizing the difference between the student model's output and the true labels of the dataset.

1.6 Interpretability Transfer between models

Interpretability can be transferred from an interpretable model to another model that lacks interpretability. This process can be particularly useful in cases where we have a pre-trained model that is interpretable, and we want to transfer its interpretability to a new model that is not interpretable. By transferring the interpretability from an interpretable model to a non-interpretable model, we can improve the transparency and understanding of the non-interpretable model, making it easier to explain its behavior and outcomes [79].

Generally, Prototypes refer to representative examples or instances used to define or classify a particular class or category. For example, in the context of clustering algorithms, prototypes are representative points used to define the boundaries of the different clusters. These algorithms aim to group similar data points together and assign them to the same cluster, with each cluster represented by a prototype.

Proto2Proto is a recent method proposed by Keswani et al. [79] that uses knowledge distillation to transfer interpretability from a teacher model to a student model. Unlike traditional knowledge distillation, Proto2Proto focuses on aligning class prototypes and feature spaces between the teacher and student models, as shown in Fig. 1.10. The class prototypes in Proto2Proto are the average representations of the examples in each class, and they are used to interpret the model's decision-making process.

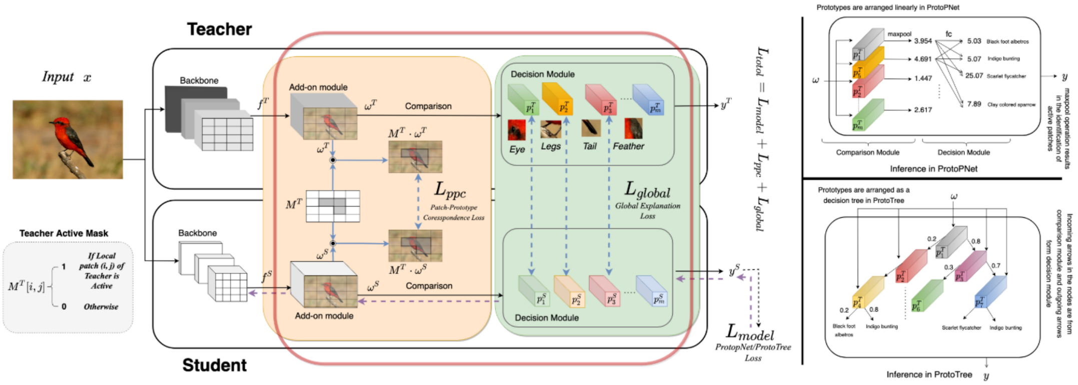


Figure 1.10: Proto2proto: Prototype based Interpretability tranfer from a large interpretabile model to a small model via Knowledge Distillation. Source: [79]

1.7 Contributions

We leverage human centered concepts for post-hoc model evaluation and ante-hoc training. We use the derive abstract concepts from the very definition of problem and show how those concepts can be used to evaluate the problem and even improve it further. Specifically, we propose a new evaluation strategy for Intrinsic Image Decomposition by measuring the *quality of disentanglement* between the decomposed components R and S. We use the core IID concepts of illumination-invariance of R and albedo-invariance of S to measure disentanglement without specifically relying on synthetic images or relative quality metrics computed on fixed sparsely annotated datasets. We choose an ML interpretability technique based upon Concept Activation Vectors (CAV) [80] for this. We use as concepts two core characteristics derived from the very definition of IID, i.e., illumination-invariance of R and albedo-invariance of S. We assess disentanglement between them by measuring the model's sensitivity to these concepts in the form of *Concept Sensitivity Metrics* (CSM) (Fig. 3.1). The CSM provides a generic framework applicable to problems other than IID using relevant concepts. We also release a new configurable dataset of images and corresponding generation scripts with controlled illumination and albedo variation. CSM is the first metric to evaluate IID via a disentanglement of R-S by measuring models' sensitivity to albedo and illumination. Our demonstrated results show that the existing IID evaluation metrics fail to capture the quality of R-S disentanglement, while CSM can capture it well.

We extend the use of human concept based interpretability methods specifically Concept Activation Vectors from post-hoc analysis to ante-hoc training by usage of our novel Concept Loss. Through our concept loss we can sensitize or desensitize a model towards user desired concepts. For example, a concept set comprising various face shapes vs. skin-tone patches can be used to learn the concept of 'face skin-tone invariance,' which can be used to improve a biased face detector model. We employ a novel teacher student distillation [71] paradigm for this concept sensitive training by learning CAVs in teacher and using them to (de)sensitize student. We additionally enhance the CAV sensitivity calculation giving it a more global sense with the proposed use of prototypes. We can encorporate the benefits of both Local and Global interpretability techniques by our proposed Concept loss and additional existing local losses. We show applications of our method in debiasing in classification problems having severe biases and prior induction in a reconstruction problem (IID). By employing multiple concept sets and corresponding CAVs, we can ameliorate several existing biases together or decompose a complex spurious correlation into smaller debiasing steps. We substantiate our claims by showing results on multiple toy and real-world classification tasks. We introduce a more challenging debiasing dataset: TextureMNIST, for a more comprehensive analysis. Additionally, we also show how our method can be used beyond classification in other challenging Computer Vision problems. Intrinsic Image Decomposition (IID) [90] to induce priors into the model, improving the performance of current SOTA solutions.

The main contributions of this thesis are as follows:

- A novel method CAV based sensitivity method for measuring disentanglement in neural networks. Specifically, we measure R-S disentanglement in IID and evaluate it via our proposed Concept Sensitivity Metric (CSM) in Chapter 3.
- A novel Concept Distillation method for concept sensitive training of Deep Neural Networks described in Chapter 4. Our proposed Concept Distillation method enables *concept sensitive training* of models by our novel sensitivity[80] inspired concept loss in a student teacher paradigm.

1.8 Thesis Roadmap

The thesis comprises five main chapters. Chapter 1 presents the briefing on ML interpretability, specifically concept-based methods and the common terms used across thesis. In it, we discuss the main approaches to ML interpretability followed by various biases in the model. We also discuss how ML interpretability can be used for model improvement, specifically bias removal. We provide an introduction to the problem of Intrinsic Image Decomposition and its evaluation methods.

Chapter 2 describes the related works to our thesis specifically discussing various concept based XAI methods used for model improvement. Various IID evaluation methods and their limitations are provided.

Chapter 3 presents a metric based on CAV sensitivity scores: Concept Sensitivity Metric, which measures the quality of disentanglement in IID.

Chapter 4 presents our proposed Concept Distillation framework for Concept Sensitive training of Deep Neural Networks via our proposed simple and effective concept loss in a distillation paradigm.

Chapter 2

Related Works

We divide the related works into existing XAI approaches, specifically Concept Based Approaches in subsection 2.1.2 followed by XAI-based model improvement methods (EGL) in section 2.2. To place the existing XAI methods in more context we also discuss some taxonomies over them as well. Since we show the applications of our concept sensitive training in debiasing and prior knowledge induction in IID we additionally discuss their existing works in section 2.3 and section 2.4.

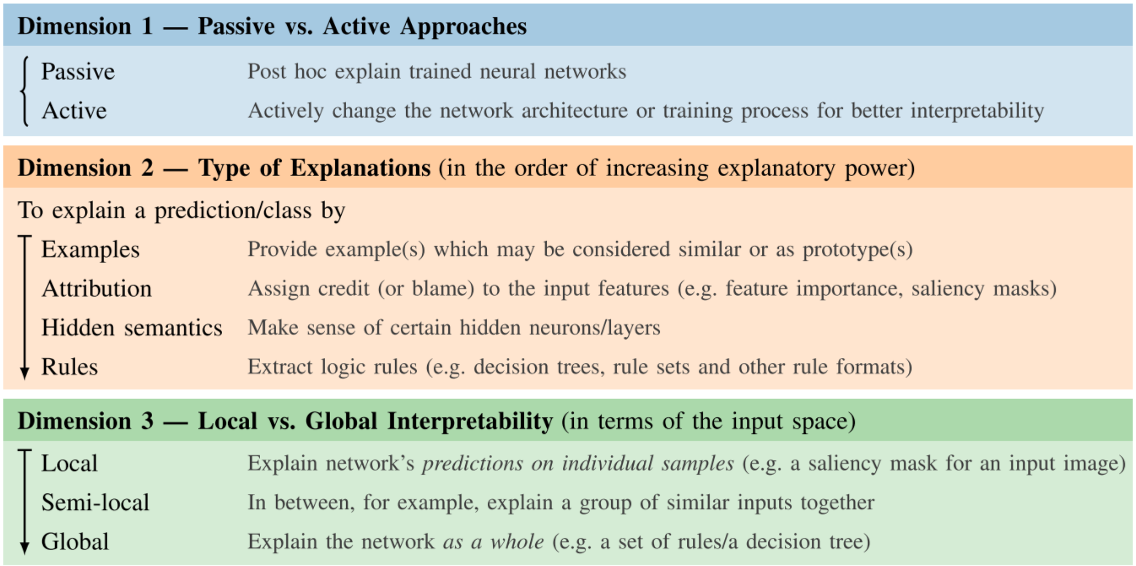
2.1 Explainable AI (XAI)

The goal of Neural Network Interpretation research is to go beyond the mere *black box* usage or accuracy-based interpretation of Deep Learning architectures and develop an understanding of the internal workings of the learned model. Several techniques have been used for this purpose like activation maps visualization [157], saliency estimation [147], model simplification [190], model perturbation [52], adversarial exemplar analysis [63], etc. For more details on XAI techniques, we point the reador to the recent surveys [3, 47, 112, 184, 199] which categorize XAI methods into various meaningful hierarchies. Apart from XAI methods, the number of XAI method surveys has also outgrown and one must not be shocked to see a survey on surveys Schwalbe and Finzel [155] too!

To understand various types of XAI methods we discuss the taxonomy by Zhang et al. [199] in Fig. 2.1. In dimension 1, the methods are divided based on wherther they do post-hoc or ante-hoc model explanations. In dimension 2, the methods are categorised on type of explanations as example-based, attribution-based, hidden semantic-based, and rule-based. While in dimension 3, local vs. global explainability seggregation is used (discussed in subsection 1.1.2). In this thesis, we focus on a specific category of post-hoc model interpretation techniques based on the analysis of concepts.

2.1.1 General XAI approaches

Some of the XAI approaces include activation maps visualization [157], saliency estimation [147], model simplification [190][137], model perturbation [52], adversarial exemplar analysis [63], *etc*.. Acti-



Dimension 1 — Passive vs. Active Approaches

Passive	Post hoc explain trained neural networks
Active	Actively change the network architecture or training process for better interpretability

Dimension 2 — Type of Explanations (in the order of increasing explanatory power)

To explain a prediction/class by

Examples	Provide example(s) which may be considered similar or as prototype(s)
Attribution	Assign credit (or blame) to the input features (e.g. feature importance, saliency masks)
Hidden semantics	Make sense of certain hidden neurons/layers
Rules	Extract logic rules (e.g. decision trees, rule sets and other rule formats)

Dimension 3 — Local vs. Global Interpretability (in terms of the input space)

Local	Explain network's *predictions on individual samples* (e.g. a saliency mask for an input image)
Semi-local	In between, for example, explain a group of similar inputs together
Global	Explain the network *as a whole* (e.g. a set of rules/a decision tree)

Figure 2.1: 3D taxonomy on XAI. Source: [199]

vation maps visualization [157] allows us to visualize which regions of an input image are most relevant for a particular model prediction. Saliency estimation [147] is another technique used to highlight important regions of an input image, but it works by assigning a relevance score to each pixel based on how much it contributes to the output prediction. Model simplification methods [190][137] aim to reduce the complexity of a model while maintaining its performance, which can make it easier to understand and interpret. This can be achieved through techniques such as pruning, which removes unimportant connections between neurons, or distillation, which trains a smaller, simpler model to mimic the behavior of a larger, more complex one. Model perturbation [52] involves introducing small changes to the input data and observing how the model's output changes in response, which can help identify which features are most important for the model's predictions. Adversarial exemplar analysis [63] focuses on generating examples that cause a model to make a specific prediction, which can help identify weaknesses or vulnerabilities in the model. This technique has been used to expose biases in machine learning models and test the model's robustness to various forms of attack. For a more comprehensive overview of state-of-the-art interpretability methods, the reader is referred to the work of Linardatos et al. [113], which provides an extensive survey of existing techniques and their applications.

2.1.2 Concept based approaches

Since CNNs are known to extract higher level features, initial concept based approaches extracted concept representations by analysing CNNs filters [16, 51]. For instance, Network Dissection [16] uses user-provided concept sets and checks their alignment with individual hidden units at each layer of CNN. Net2Vec Fong and Vedaldi [51] used a probe-dataset to collect CNNs features and then learn the weights

Table 2.1: Concept Representations of existing ML interpretability literature

Concept type	**Concept Representation Method**	**Papers**
Non-hierarchial	Hidden Units Concept images alignment based	Network Dissection [16]
	Concept Embeddings Vectos based	CAV [80], A-CAV [167], CG [13], ICB [201], ICE [198], CaCE Goyal et al. [64], ICS [153], ConceptSHAP [193], [37]
	Proto-types	Bontempelli et al. [26, 26], Chen et al. [35], Li et al. [103]
	Neuro-Symbolic	RRC [170], COOL [120], RRR [170], NeuroSC Marconato et al. [119],
	Others	Multi Agent Debate [84]
Hierarchial	Neuron Attribution	HINT [185]
	Weight Attribution	Concept Graphs: Kori et al. [85], Zhang et al. [196]; Decision Trees: POEM [41], CHAIN [187]
	Symbolic features	Concept trees: Santhirasekaram et al. [148], CNN2DT [75], TreeICE [128], ACDTE [160]

Table 2.2: Concept Discovery categorization

Type	**Method used**	**Papers**
Post-hoc	Super-pixel	ACE [61], CocoX [4]
	Coorperative Attribution (shapeley)	ConceptSHAP [194]
	Autoencoders based	PACE [76]
	Causality based	MCE [183]
	Using Probe Dataset	ACDTE [160]
	Using GANs	Alipour et al. [5]
Ante-hoc	Saliency based	HINT [185], Kori et al. [85]
	Using Slot-attention [117]	Stammer et al. [170]
	Latent Space Disentanglement based	GAN based: EPE [162], StylEx [95], Dissect [60] img2tab[166], Charachon et al. [33], Augustin et al. [10], Tran et al. [180], Causality enforcing O'Shaughnessy et al. [133]
	XIL based	iCSNs [171], RRC [170]

to recognize the concept in various semantic tasks. Kim et al. [80] quantified concepts via a linear decision boundary separating the concept of interest and randoms learning Concept Activation Vectors (CAV). This work became widely popular due to their simple but effective concept representation in terms of concept vectors.

Mathematical details of our base method: CAV We now discuss the mathematical formulation of CAVs. For input image x from the test set, layer l's activation is given by $f_l(x)$. Given a concept of interest C, a set of images of that concept are taken as positive samples, vs. a set of random images is taken as a negative concept set (C'). A binary linear classifier (linear regression or SVM) is used to distinguish between l's activations for C and C' (*i.e.*, between sets $f_l(x) : x \in C$ and $f_l(y) : y \in C'$). The vector representing the classifier hyperplane is stored as the CAV, v_C^l. Model's concept sensitivity $S_{C,l} \in [0,1]$ for a given class sample x is estimated by calculating the alignment between the learned CAV v_C^l and the gradient $\nabla L_l\left(f_l(\boldsymbol{x})\right)$ of loss with respect to activation for that layer (computed via backpropagation) as:

$$S_{C,l}(\boldsymbol{x}) = \nabla L_l\left(f_l(\boldsymbol{x})\right) \cdot \boldsymbol{v}_C^l, \quad \text{where} \quad \nabla L_l\left(f_l(\boldsymbol{x})\right) = \frac{\partial L(\boldsymbol{x})}{\partial \boldsymbol{x}_{a,b}}. \tag{2.1}$$

TCAV has been used to analyze classifiers using a Cross-entropy (CE) loss between the predicted logit and GT class label. We use TCAV to evaluate the disentanglement of image decomposition by introducing a pixel-wise loss instead of CE. For pixel location (a, b), L is calculated as MSE between the predicted $\hat{R}$, $\hat{S}$, and their respective GT values. Finally, the complete TCAV sensitivity score for concept C is computed as the fraction of inputs (x's) in the complete concept set X, which were positively aligned with v_C^l.

$$\mathrm{TCAV}_{C,l} = \frac{|\{\boldsymbol{x} \in X : S_{C,l}(\boldsymbol{x}) < 0\}|}{|X|} \tag{2.2}$$

Note that the sign is negative here because the gradient is taken with respect to loss instead of logit values. Thus when C has a positive influence, the loss is minimised[1].

Other Concept Representation Methods [153, 167, 198] refined CAV representations further. Goyal et al. [64] combined*Causal* modelling with CAVs and qunatified Causal Concept Effect (CaCE). For this they generate counterfactual examples of the concept of interest and check CaCE scores of them. Another important measure for checking usefullness of concepts is via checking their Completeness: how sufficient a particular set of concepts is for explaining the model's prediction. ConceptSHAP [193] measures the Completeness of concepts by using Shapeley values associated with cooperative game theory. On the other hand, Concept Whitening [37] disentangles concepts by decorrelating the latent space and projecting concepts such that they align in different directions. This ensures that concepts are well decomposed and distributed in latent space. Adversarial TCAV Soni et al. [167] and Concept Gradients Bai et al. [13] extends the CAVs to model non-linearity. While we are focused on CAVs in this thesis we refer readers to recent surveys for more details on other Concept Based Representations and advancements Gupta and Narayana [67], Hitzler and Sarker [73], Holmberg et al. [74, 74], Schwalbe [154]. Hitzler and Sarker [73] present a survey on CAV [80] based representations whileSchwalbe [154] survey various visual Concept Embeddings giving a taxonomy of Concept Analysis approaches. Since many of the methods require user provided concepts there are many widely used concept datasets [60, 111]. We categorize various existing concept representations as given in Tab. 2.1.

Automatic Concept Discovery Methods Since the current approaches require user given concept sets, many concept discovery algorithms were propose to automate this. We provide the taxonomy of varioous concept discovery algorithm in Tab. 2.2. ACE [61] and CocoX [4] use super-pixels followed by saliency maps to discover concepts. ACE uses varying levels of super-pixels to generate both lower (color, texture) and higher-level features (objects) and clusters their activations in the model's activation plane to get concepts. These clusters need to be identified with human interpretable concepts, which need a human manually going through them. ACE automates this by using ImageNet-trained CNN features as a guide. It compares the perceptual similarity of identified cluster segments with the guide to label them. It also removes outliers to make the concepts *coherent*. It uses TCAV [80] scores to gauze concept *importance* for prediction. On the other hand, CocoX [4] uses Grad-CAM [157] followed by pooling to determine importance of super-pixels. It then selects the top p super-pixels for each class based on the importance weights and clusters the actual image regions corresponding to important super-pixels for getting concepts. ConceptSHAP [194] uses Shapeley values that attribute importance

[1] https://github.com/tensorflow/tcav

to concepts when combined to discover a complete set of concepts. For the scope of the thesis, we do not include more details on these methods in this chapter, but refer readers to our survey paper on the same [67].

2.2 Using XAI to Improvise models: Explanation Guided Learning (EGL)

While XAI approaches offer post-hoc analysis of models and provide feedback regarding models decision process, it is importance to use this feedback for improving the models. XAI-based model improvement, also referred to as Explanation Guided Learning (EGL), uses this feedback given by XAI approaches to guide the models towards desired explanations. [18, 55, 69, 189] survey various XAI approaches for model improvement.

Note that (i) includes (ii), as using explanations for better generalization will automatically improve the interpretability of models. In the following subsections, we discuss the categorization of generic XAI methods.

2.2.1 EGL categorization based on augmentation

Weber et al. [189] classify based on augmentation made for improvement, which can be over training data, features, loss, gradient, model *etc.*(Fig. 1.8). The methods which augment data do it by removing spurious correlations from training data, using class-prototypes [57, 79]. It is to be noted that Weber et al. [189] include XIL-based methods into methods augmenting data while we categorize it differently because of the very different nature of these methods requiring interactive feedback from methods that augment training data directly. Among methods that augment the features, Attention Branch Network [54] masks features based on class activation mapping [200] based interpretations to make the model focus on correct attention masks. ClArC [9] augments intermediate activations (features) based on their orientation with CAV. While RRR [139], CDEP [138], Selvaraju et al. [158], Chen et al. [36] including our proposed method of concept distillation augment the loss function for enforcing explanation alignment penalization on the model. ha Lee et al. [68], XAI-GAN [129] and augment the gradient in different ways for enhancing learning via XAI. Among methods augmenting model, Yeom et al. [195] prune the model according to LRP [22] while Sabih et al. [140] quantize the model using DeepLIFT explanations [101].

2.2.2 EGL categorization based on goal

We categorize XAI-based model improvement in terms of the end goal as aiming *(i)* enhanced interpretability and *(ii)* better generalization. Most of the existing literature focuses on the former [7, 53, 77, 79, 83, 92, 141, 150, 151, 170, 171, 177, 179, 186, 191] with some works addressing the latter [9, 12, 25, 26, 53, 88, 152, 159, 163, 166, 170]. Such division concurs with Holmberg et al. [74] that found that explanations with the goals of better understanding have a significant difference

Table 2.3: Taxonomy of Concept based model improvement methods

Model Improvement Goal	**Method**	**Sub Method: Papers**
Better Interpretability	Concept Conditioned Prediction Based	Supervised: CBM [83], CME [77]
		Partially-Supervised: CBM-AUC [151]
		Unsupervised: SENN [7], XAL based: CALI [177], C-SENN [150], Sarkar et al. [149]
	Concept Reasoning based	Neuro-Symbolic: RRC [170]
	Interaction based	Human Interaction: Lage and Doshi-Velez [92], NesyXIL [170]
		Proto-type based: proto2proto [79], Xue et al. [191], Wang et al. [186], Sacha et al. [141]
Better Generalization	CAV based	few shot: ClArC [9]
		zero shot: Concept Distillation chapter 3
	Causality Based	Bahadori and Heckerman [12]
	Latent Space Disentanglement based	Neuro-Symbolic Reasoning Based: NeSyXIL [170]
		GAN based: Img2Tab [166]
		XAI based: ProtoPDebug [26], CAIPI [152], Interactive CBM's [34], Bontempelli et al. [25]
	Probability Distribution based	Kronenberger and Haselhoff [88]

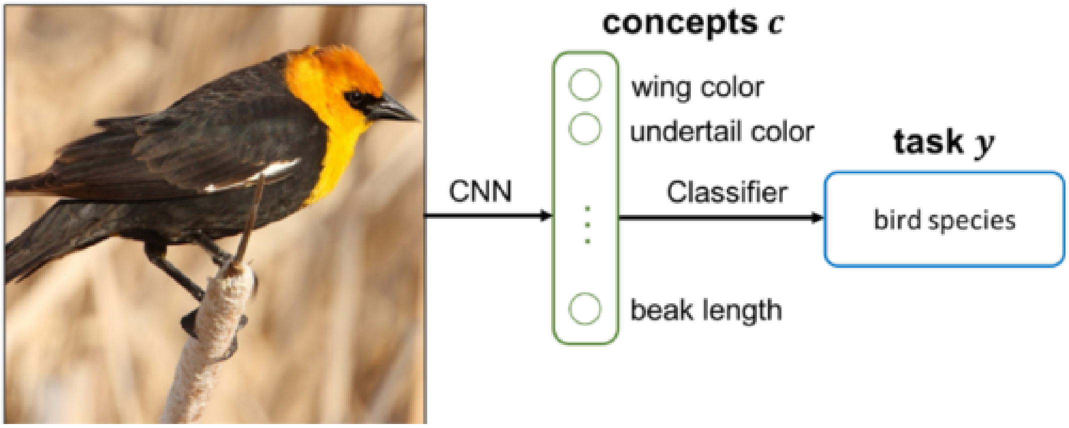


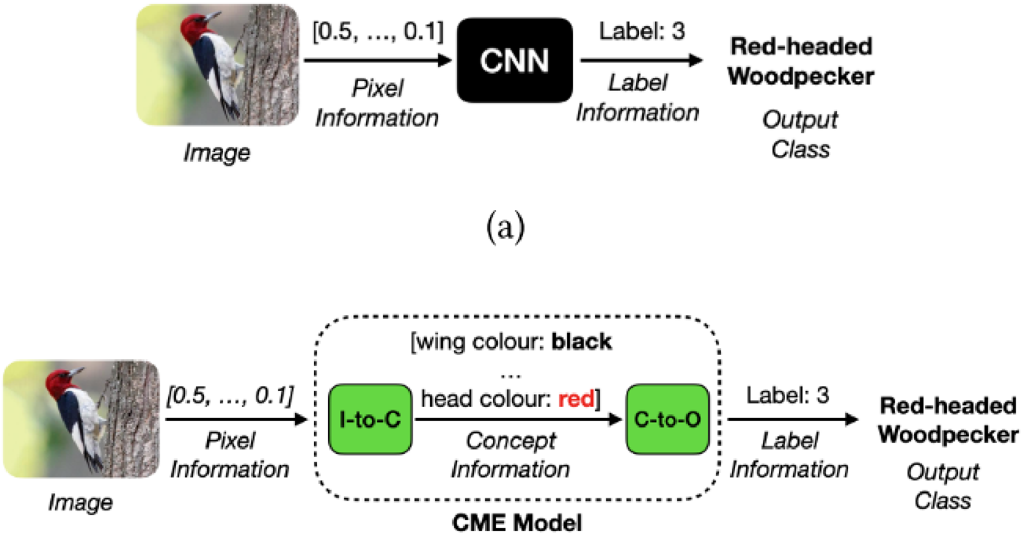
Figure 2.2: CBMs: Concept Bottleneck Models predict a set of concepts c, then use c to predict the final output y. In bird identification the concepts like wing color etc are predicted first and this is further used to predict bird label. Source: [83]

from explanations to find actionable decisions. Note *(i)* includes *(ii)* because using explanations for better Generalization will automatically improve the Interpretability of models. We restrict ourselves to Concept-Based Model Improvement Methods in this section.

2.2.2.1 Aiming better interpretability

These methods aim for no class accuracy impairment along with enhanced Interpretability of the model and can be categorized broadly based on the type of interpretability methods used as shown in Tab. 2.3.

- **Concept Conditioned Prediction Based**
  - **Supervised** Concept Bottleneck Models (CBMs) [83] use concept mapping as an intermediate step in the model's final prediction. Specifically, they map the input x to a concept c using a concept prediction module $g : x \to c$, which is then used to predict the target class y using a classification module $f : c \to y$. This conditioning of the model's prediction over concepts improves the interpretability and generalization of the model. The authors tested CBMs using three different training paradigms: independent training of g and f (with f trained on ground truth concepts), sequential training (where f is given g's predicted concept $\hat{c}$), and joint training (where both f and g are optimized over a joint objective). They found that all three training paradigms produced similar classification accuracies for both concept prediction and target class prediction. However, independently trained models had better test-time interventions. They intervene not on the actual value of concept c but on the model's concepts predictions $\hat{c}$. Thus they do not check the induction of concept c unlike our work: Concept Distillation discussed in chapter 3.



(a)

(b)

Figure 2.3: Concept-based Model Extraction (CME): The CNN model processes input images in a black-box fashion using pixel information to predict the class label. On the other hand, the CME extracted model computes concept information, such as bird wing or head color values, from the input image using an Input-to-Concept function. Then, a Concept-to-Output function is used to generate the output class label based on this concept information. Source: [77]

- **Partially-Supervised** Concept-based Model Extraction (CME) [77] approximates DNNs using simpler interpretable models like linear, logistic regression, and decision trees. While CBMs require binary concepts, CME can handle multi-valued concepts and can derive concepts combining multiple layers (unlike TCAV, CBM, *etc*., which require one layer to be chosen). Additionally, CME can train in a partially supervised manner with few labeled and other unlabelled samples. Further, it assumes k different concepts forming concept representation $\mathcal{C} \subset \mathbb{R}^k$ such that every basis vector in $\mathcal{C}$ spans all possible values of a particular concept. They define two functions for mapping input to concept space ($p : \mathcal{X} \to \mathcal{C}$) and concepts to predicted class space ($q : \mathcal{C} \to \mathcal{Y}$). CME approximates f with $\hat{f}$ given as $\hat{f}(\mathbf{x}) = \hat{q}(\hat{p}(\mathbf{x}))$ where $\hat{q}$ and $\hat{p}$ are extracted by CME. For this, they define a function $g^l : \mathcal{H}^{\updownarrow} \to \mathcal{C}$ $\hat{g}$ and extract it by SSMTL [114] using an approximation of k separate tasks (one for each concept). For every concept, i [77] finds the best layer for predicting the concept by minimizing a loss function l $l^i = \underset{l \in L}{\arg\min} \ell\left(g_i^l, i\right)$ and finally approximating $\hat{p}$ as

$$\hat{p}(\mathbf{x}) = \left(g_1^{l^1} \circ f^{l^1}(\mathbf{x}), \ldots, g_k^{l^k} \circ f^{l^k}(\mathbf{x})\right).$$

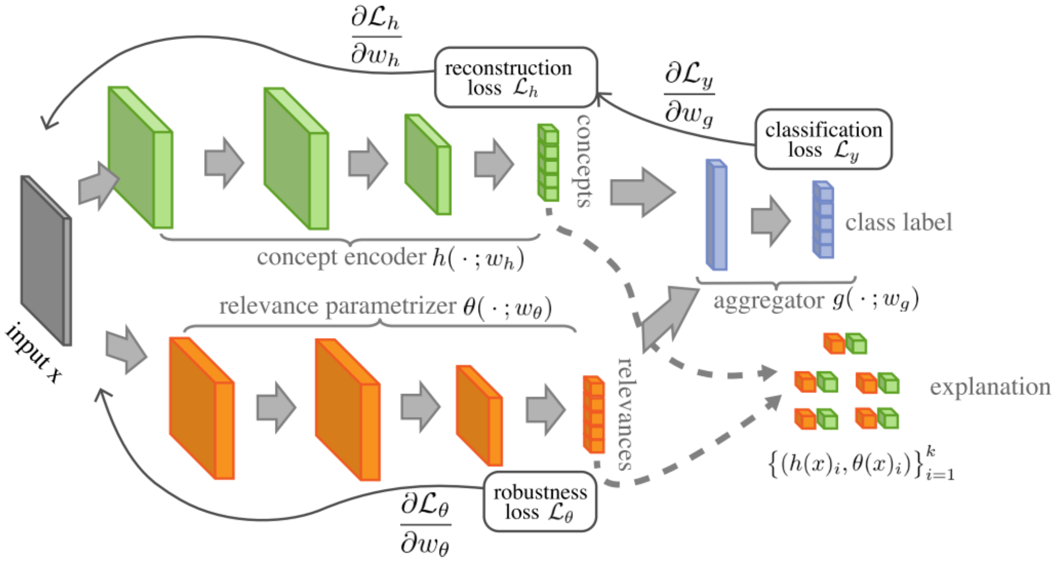

$\frac{\partial \mathcal{L}_h}{\partial w_h}$
reconstruction loss $\mathcal{L}_h$
$\frac{\partial \mathcal{L}_y}{\partial w_g}$
classification loss $\mathcal{L}_y$
concepts
class label
concept encoder $h(\cdot\,;w_h)$
aggregator $g(\cdot\,;w_g)$
relevance parametrizer $\theta(\cdot\,;w_\theta)$
input x
relevances
explanation
$\{(h(x)_i,\theta(x)_i)\}_{i=1}^{k}$
robustness loss $\mathcal{L}_\theta$
$\frac{\partial \mathcal{L}_\theta}{\partial w_\theta}$

Figure 2.4: Select Explaining Neural Networks (SENNs) consist of a concept encoder (green), an input-dependent parametrizer (orange) which generals relevance scores, and an aggregator which gives final prediction. Source: [7]

$\hat{q}$ is approximated in a supervised manner using a mapped set of concept labels and target labels (class labels)as a training set and fitting using Decision trees or logistic regression. Concept Bottleneck Model with Additional Unsupervised Concepts (CBM-AUC) [151] combines CBMs with SENNs to extend CBMs to additional unsupervised concepts along with supervised concepts.

- **Unsupervised** Self-Explaining Neural Networks (SENN) [7] is a type of Unsupervised method which learns interpretable basis concepts by approximating a model with a linear classifier. They focus on explicitness, faithfulness, and stability via regularizing models. For a linear classifier $f(x) = \theta(x)^T h(x)$ where x is input, $h(x)$ is function mapping input to concepts, and θ represents model parameters, they enforce the following constraints: *(i)* The output of f is approximately same for two close inputs that is $\nabla_x f(x) \approx \theta(x_0)$ for all x in a neighborhood of x_0; *(ii)* model is linear in terms of concepts; *(iii)* Aggregation function for features shall be generic enough such that it is permutation invariant, isolate multiplicative interactions between concepts and preserve sign and relative magnitude of relevance values $\theta(x)_i$. Finally, they define self-explaining prediction model as: $f(x) = g(\theta_1(x)h_1(x), \ldots, \theta_k(x)h_k(x))$ with some additional constraints [7]. When the above formulation is applied over a neural network, it becomes SENN (the condition being g is continuous over concepts and model weights). SENN is capable of using user-provided concepts as well as learning new concepts which satisfy **Fidelity**: concepts persevere relative information and **Diversity**: concepts for a particular input are non-overlapping. For SENN, they learn an autoencoder h and ensure sparsity constraint to increase diversity while using proto-types for interpretations. They use a concept encoder that transforms inputs to concepts, an input-dependent parameterizer for generating relevance scores, and an aggregation function that combines them for the prediction of class labels. The concepts and their relevance predictions form an explanation. SENN has reduced interpretability in real-world tasks like autonomous driving, which is overcome by C-SENN [150]. C-SENN combines Contrastive learning with concept learning of SENN to improve the discovered concepts and task accuracy.

Sarkar et al. [149] use a base encoder (which is the same as the second last layer of the classifier) followed by two branches: one for concept and one for classification. The classification branch resembles the last layer of the classifier and gives the final class label prediction while the concept head has a concept decoder that reconstructs the image given the intermediate representation (features by the base encoder). Thus the intermediate representations act as concepts. They additionally impose an image reconstruction loss apart from fidelity and classification losses.

- **Concept Reasoning Based (Neuro-Symbolic)** Right for the Right Concept [170] uses a concept embedding module that learns concepts via slot attention Interaction based[117] and a reasoning module that reasons for the concepts. The concept embedding module creates a decomposed representation of input space which can be mapped to concepts, while the reasoning module makes predictions based on the above-obtained concepts.
- **Interaction based** Interaction based methods aiming for better interpretability need interaction with an expert which is eigther done by a human expert or by using class prototypes. Henceforth, they can be classified into broadly two categories as follows.
  - **Human interaction based** Lage and Doshi-Velez [92] learn interpretable models via user feedback on concepts(which ones are similar and which are not) and also which concepts should(not) affect. Interactive Concept Swapping Networks iCSN's [171] bind concepts to prototypes by swapping the latent representations of paired images. They use prototypes for interactively learning concepts with user feedback. A Human can query iCSN prototypes and update them for concepts.

    NesyXIL [170] adds a user interactive layer over Nesy [170] discussed above.

    For a detailed survey on methods that use explanations in interactive ML please refer [53, 179].
  - **Proto-type based** Proto2Proto [79] uses knowledge distillation to transfer interpretability from (interpretable) teacher to a student model.

    Xue et al. [191] use global and local prototypes for enhanced interpretability in Visual Transformers (ViT). Wang et al. [186] use a macro (broader) and micro proto-types (more specific) for interpretable models learning from mistakes. A similar idea of macro and micro prototypes is used by Sacha et al. [141] which leverages support prototypes capturing macro-level features and trivial proto-types capturing specific micro features. The support (or macro) proto-types provide a global overview of the concepts (or features) but cannot capture some class-specific trivial features (thus captured by trivial (or micro) proto-types.

2.2.2.2 Aiming better generalization

These models typically remove bias or confounding factors and show accuracy improvement on poisoned datasets. [53] gives a topology exploring mitigation of shortcut behavior in models which involves steps of Select, Explain, Obtain feedback, and Revise model. We divide these methods as follows.

- **CAV based** They use CAVs [80] for the representation of concepts and move activations to make a model sensitive or insensitive to the concept.
  - **Few Shot** Anders et al. [9] does artifact removal from models by moving activations of images according to the CAV's learned for class images containing artifact vs. non-artifact class images. It is a few-shot method since it requires artifact and non-artifact images. It uses two methods for the removal of artifacts: Augmentative and Projective. Augmentative Class Artifact Compensation (AClArC) augments the class samples trying to remove the artifact by moving them according to the CAV direction and retraining the original model. Projective Class Artifact Compensation (PClArC) on the other hand moves the class activations to a concept-neutral direction in the model's activation space by a simple linear transformation.
  - **Zero Shot** There is no existing zero-shot method to the best of our knowledge, and we provided the first zero-shot concept-based model improvement method: **Concept Distillation**, which uses pre-trained model's conceptual knowledge to train a student model via concept and proto-typical losses. Our proposed method is discussed in chapter 3 in detail.
- **Causality based** Causal modeling involves checking for causes of a particular effect (here model's particular predictions). Bahadori and Heckerman [12] use causal graphs for debiasing CBM's removing confounding factors or clever-hans. It models the impacts of unobserved variables using causal graphs and removes them with a two-stage regression technique aided by instrumental variables.
- **Latent Space Disentanglement Based** They disentangle latent space to represent similar concepts in similar spatial regions. Latent space disentanglement is typically achieved through the use of generative models, such as autoencoders, variational autoencoders (VAEs) and Generative Adversarial Networks.
  - **GAN based** Img2Tab [166] as described above, allows user interventions on the model for concept based debugging by identifying class-relevant concepts from W_k metric or classifier and using those to filter the semantics our of classifier P by masking all unwanted features across training samples.

  - **Neuro-Symbolic Reasoning Based** NeSyXIL [170] allows for user corrections to its concept embedding module explanations or reasoning modules explanations.
  - **User Interaction for input based** ProtoPDebug [26] debugs part-prototype networks using a concept-level debugger with human supervision regarding what part-prototype is forgotten or kept. Right for the Right Latent factors [159] provides a debiasing approach for generative models via disentanglement of latent space with human feedback. They enforce disentanglement via ELBO loss and a match pairing loss [163]. CAIPI [178] propose an XIL framework where DNN's query the user while the user explains the queries and corrects the explanation. Right for the Right Scientific Reasons RRSR Schramowski et al. [152] use CAIPI or RRR [139] depending on the task and demonstrate results removing clever-hans phenomena. Interactive CBMs [34] extend CBMs to XIL using an interaction policy that selects which concept labels to be queried from the user to improve prediction maximally. Their policy combines concept prediction uncertainty and the influence of concept on model prediction. [25] introduce a debugging technique for CBM's debugging using human supervision. They can intervene on both concept-level and input-level bugs. CALI [177] extends SENN to XAL setting learning SENNs from class labels and explanation guidance by users.
- **Probability Distribution based** [88] uses explanations to verify (accept or reject) the prediction. They show results over GTSRB dataset [169], which consists of 43 different classes showing german traffic light signs. They further use three attributions of varying complexity as concepts: *Simple:* Color *Medium:* Primitive Shapes like squares, circles, triangles, octagonal, etc. *Complex:* Numbers (0-9) and symbols (truck, animal, car, children, bicycle, etc) The examples above are included as synthetic data.

2.3 Various Debiasing Methods

To restate, ante-hoc model improvement methods utilize explanations from the pre-trained model to remove biases that degrade the performance on Out-Of-Distribution (OOD) samples in a corrupted test set. We can subdivide them into three categories based on the amount of corruption information available during retraining:

2.3.1 Multi-shot Scenario

If we have sufficient OOD samples available beforehand, we can pose them as target distribution and directly use domain generalization techniques like feature augmentation [104, 124], adversarial learning [99, 100]. Such methods debias the model by generalizing the features to appropriately represent corrupted information by directly learning on the OOD samples. Hence such methods can directly penalize biases using data without explicitly requiring their identification and modeling.

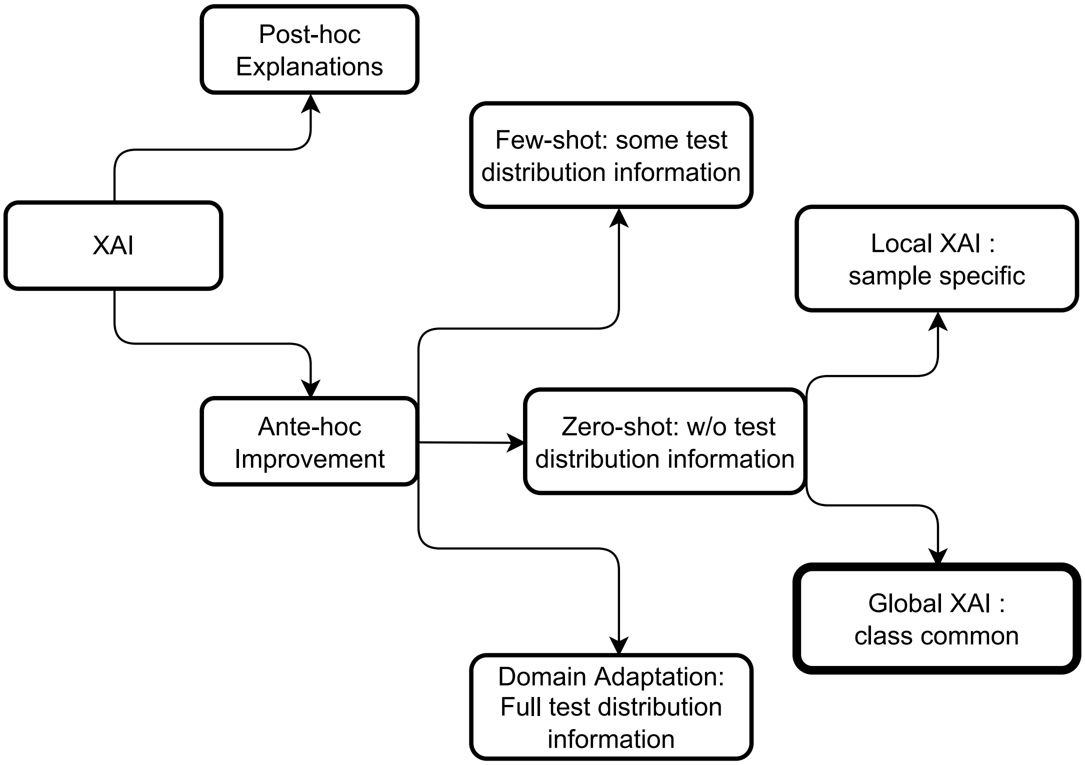

Post-hoc Explanations
XAI
Few-shot: some test distribution information
Local XAI : sample specific
Ante-hoc Improvement
Zero-shot: w/o test distribution information
Global XAI : class common
Domain Adaptation: Full test distribution information

Figure 2.5: Categorization of debiasing methods with ours highlighted.

2.3.2 Few-shot Scenario

If only a limited number of OOD samples are available, then they can be used to model the underlying bias in a few-shot manner and then utilized for generalization. DFA Lee et al. [98] use a small percentage ($< 1\%$) of the corrupted test set as adversarial samples during training. They design separate encoders for learning disentangled features for intrinsic and biased attributes, which they swap to improve feature diversity during debiasing. Due to the limited OOD sample dependence, the extracted interpretations might not be generic enough for complete debiasing and would require retraining in case of small corruption parameters perturbation. A contemporary work Debiasing Alternate Networks (DebiAN) [110] uses a bias discoverer coupled with a classifier that uses predicted class-softmax probabilities differences across various samples to detect and mitigate biases.

2.3.3 Zero-shot Scenario

The last category is when we do not have any OOD samples available beforehand. Such methods identify existing biases by interpreting the pre-trained model and induce this as prior knowledge to generalize the learning to OOD cases in a zero-shot setting by augmenting loss functions [49, 98, 138, 139]. We split them into two types based on the scope of their interpretability:

Local Methods: These methods use local explanations (LXAI) for bias assessment *i.e.*they identify bias on a per-sample basis and then use it for ante-hoc zero-shot improvement. Note that the methods do not require OOD samples for bias assessment and hence still fall under our zero-shot categorization. RRR by Ross et al. [139] uses a penalization term if it does not prefer the Right answers for Right Reasons by using explanations as constraints. Erion et al. [49] augment learning gradients with a penalizing term to match with user-provided binary annotations. Sun et al. [172] use LRP [11] in a cross-domain setting and later extend the idea [173] to weigh feature maps for fine-tuning image captioning models by modulating feature importance scores. Most relevant to our work is CDEP by Rieger et al. [138], which introduces an explanation penalization based augmentation to the loss term computed using the model's current and the desired explanations. Their main idea is to use contextual decomposition [125] to estimate feature interactions in addition to pixel importance. Thus they require additional user-provided feature importance scores for each input sample. For ColorMNIST, they minimize the contribution of pixels in isolation (which represents color), forcing the model to focus on the interaction between pixels (thereby learning shape). On the other hand, we can directly specify concepts like color and shape via our concept set-based definition. Furthermore, it is not straightforward how to encode complex concepts like age, gender, race, etc, using their method.

Global Methods: Contrary to the sample-specific explanations based on local methods, recently, some global methods have come up which detect biases for an entire class or all the classes (*i.e.*complete domain). They explicitly define and penalize the concepts behind the class/domain biasing attributes. This is based on the understanding that we humans also construct class common concepts while learning [93] instead of mere sample specific interpretations. There are only a handful of methods in this category. A couple of methods focus on improving the reasoning of models via neuro-symbolic concept representations [170, 171]. RRC [170] maps the concepts to a better disentangled embedding space and requires interactive user feedback for improving the model. Anders et al. [9] uses CAVs to perturb input features towards or away from the concept. They leverage CAVs like us for concept identification, but their concept definition is restrictive as they learn CAVs directly in the base model's activation space. We learn CAVs from a large teacher transformer which provides more unbiased conceptual knowledge. Similarly, Proto2Proto by Keswani et al. [79] also uses knowledge distillation but focuses alignment of classes proto-types and feature spaces instead of concepts.

Due to individual sample-specific modeling, several rich semantics emerging out of cross-sample/inter-class relationships and abstract intrinsic class attributes are not explicitly encoded in the local methods. This limits their utility in complex real-world problems. Our method falls in the global category model class-wise instead of sample-wise, not directly requiring any OOD samples for debiasing. We directly encode the class intrinsic and biasing attributes via concepts. Furthermore, our method can be easily extended as a hybrid approach by supplementing it with local techniques and few-shot information for additional performance gains.

2.4 Intrinsic Image Decompostion

2.4.1 IID Methods

IID, as modeled in Eq. 3.1, was first proposed by Land and McCann [94]. Earlier IID solutions were mostly unsupervised optimization-based approaches constrained by strong assumptions and specific auxiliary inputs like time-lapse video [109], multi-view images [48], IID using stereo images [91], IID on RGBD data [14], focal stacks [146], etc. Single image IID methods depend upon complex cost functions and optimization algorithms like IID by chromatic clustering [56], convex energy minimization [62], hierarchical priors [144, 145], *etc*.. With the advent of Deep Learning, various supervised, semi-supervised, and unsupervised Neural Network based solutions have been proposed in the literature. Initially, Bell et al. [20] and Zhou et al. [202] proposed a hybrid DL and optimization-based framework for IID. Narihira et al. [130] proposed a direct R and S regression framework trained on synthetic Sintel dataset [27]. Li and Brown [106] introduced a relative loss function for reflectance estimation. Li and Snavely [109] learned an unsupervised IID model using time-lapse videos and consistency loss. Finally, Li and Snavely [108] and Fan et al. [50] trained multiple sequential modules supervised by hybrid synthetic, sparse, and dense datasets with appropriately designed loss functions. A fully unsupervised DL approach has been proposed [116], which poses IID as a style-transfer problem. PIE-Net [42] has a hybrid-CNN approach for addressing shading-reflectance leakages in strong illumination conditions whereas Baslamisli et al. [15] use photometric invariance and some other physics-based priors in encoder-decoder architecture.

As earlier optimization-based approaches are interpretable by design, we focus on recent state-of-the-art DL-based IID models. Specifically, we focus on 3 IID solutions:

– Intrinsic Images by Watching the World (**IIWW**) [109] trained in a partially-supervised manner on their self-introduced Bigtime dataset consisting of time-lapse videos of natural indoor and outdoor scenes.

– CGIntrinsics (**CGIID**) [108] which does supervised training on their new synthetic dataset containing physically based renderings with GT R and S, as well as natural scenes from IIW [20] and SAW [87] datasets.

– Unsupervised Single Image Intrinsic Image Decomposition (**USI3D**) [116] which first disentangles content from style features, then utilizes adversarial learning to separately learn R and S style domains and performs content preserving image translation with consistency losses for IID in an unsupervised training regime.

2.4.2 Evaluation Strategies

Since there is a lack of dense real-world GT annotations for R and S, all the above models are evaluated on synthetic images (ARAP [23], Sintel [27]), small single object scenes (MIT Intrinsics [65]) or sparse manual annotations (IIW [20], SAW [87]). Synthetic GT-based evaluation is affected by synthetic-natural domain shift, whereas single object images do not capture the complexity of everyday natural scenes. Sparse manually annotated GT from IIW and SAW either provide only relative assessments or use classification accuracies on fixed shading categories. Additionally, Bonneel et al. [24] acknowledge the issue of IID evaluation and propose to evaluate IID quality by estimating the performance in downstream image editing applications (logo removal, shadow removal, texture replacement, and wrinkles attenuation) using the decomposed R and S components. None of these IID evaluation strategies specifically capture the disentanglement quality of the decomposed R and S, and implicitly assume that the small set of curated GT annotations/cases represents all the possible test scenarios.

2.4.3 Metrics

On densely annotated synthetic GT images, IID quality is measured using pixel-to-pixel comparisons and metrics like Local Mean Square Error (LMSE), Mean Squared Error (MSE), and Difference in Structural Similarity Index (DSSIM). These metrics are not robust to the ambiguous nature of the IID problem ($I = R \odot S = \lambda R \odot \frac{S}{\lambda,} \quad \forall \lambda \in \mathbb{R}^+$). For sparse human annotated GTs like IIW and SAW, Weighted Human Disagreement Ratio [20] (WHDR) measures the percentage of disagreement between human assessment and model prediction weighted by the confidence of each annotation. The SAW AP$\%$ [87] is calculated based upon average precision on varying recall percentages over the classification of pixels into smooth vs. non-smooth shading regions. Both of these metrics only assess a sparse set of pixels and ignore specific cases like multiple shadows, colored highlights, material transmissivity, *etc*.. Also, they are specific to the dataset, which is comprised mostly of indoor scenes. Current limitations of IID metrics motivate the search for a more comprehensive and fundamental evaluation strategy which we attempt to address through our proposed approach in this thesis.

Chapter 3

Interpreting Intrinsic Image Decomposition using Concept Activations

3.1 Introduction

Image-Based Inverse-Rendering (IBIR) problems like image stylization, image harmonization, illumination estimation, palette extraction, *etc*., are often under-constrained and ill-posed in nature. They are under-constrained as we need to estimate more output parameters than available inputs. For example, style-content decomposition for single-image stylization lacks constraints. These problems are also frequently ill-posed due to the underlying optical model approximations and assumptions like diffuse surfaces, monochromatic illumination, point light source, *etc*.. As a result, performance evaluation of their solutions becomes a challenging task. The issue is exacerbated due to the lack of a proper ground truth dataset and evaluation metric. To address this issue, we propose a novel *evaluation by interpretation* technique in this paper thereby introducing a Concept Sensitivity Metric (CSM). We focus on one such problem in this thesis. *i.e*., Intrinsic Image Decomposition (IID) [90].

IID is an IBIR task that involves decomposing a given image into its constituent illumination-invariant Reflectance (R) and albedo-invariant Shading (S) components. The decomposition finds direct use in many applications, such as shadow removal [89], image colorization [115], material manipulation [24], relighting [48], and retexturing [24]. Current IID methods assume a simple Lambertian reflectance model on diffuse surfaces:

$$I = R \odot S, \tag{3.1}$$

where $\odot$ denotes element-wise multiplication. Due to the under-constrained nature, existing IID methods either depend on hand-crafted [20, 106, 161] or deep learned [42, 108, 109, 116] priors.

Performance evaluation of IID is carried out on a small number of natural images like MIT Intrinsics [65], sparse human annotation datasets like Intrinsic Images in the Wild (IIW) [20], Shading Annotation in the Wild (SAW) [87], *etc*., or synthetic datasets like Sintel [27], As-Realistic-As-Possible (ARAP) [23], *etc*.. For densely annotated GT images, evaluation is carried using dense per pixel error estimation or using quality score involving metrics like Peak Signal-to-Noise Ratio (PSNR), Local Mean Square

[1] https://avani17101.github.io/Concept-Sensitivity-Metric/

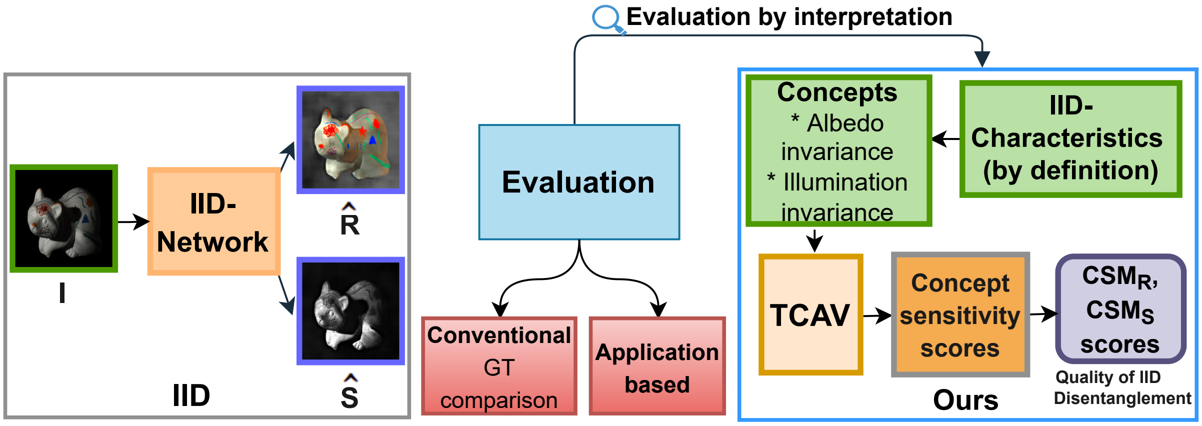

Evaluation by interpretation
I
IID-Network
R̂
Ŝ
IID
Evaluation
Conventional GT comparison
Application based
Concepts
* Albedo invariance
* Illumination invariance
IID-Characteristics (by definition)
TCAV
Concept sensitivity scores
CSM_R, CSM_S scores
Quality of IID Disentanglement
Ours

Figure 3.1: Evaluation-via-interpretation: Given a pre-trained IID network, existing evaluation techniques rely on comparison with ground truth or performance in a downstream application. We propose a novel *evaluation-via-interpretation* strategy based on learned Concept Activation Vectors (CAV) [80]. We estimate concept sensitivity scores and evaluate IID performance by measuring the quality of albedo-illumination disentanglement via our proposed Concept Sensitivity Metric (CSM).

Error (LMSE), Difference in Structural Similarity Index (DSSIM), *etc*.. For sparsely annotated GT datasets, IID-specific metrics like Weighted Human Disagreement Ratio (WHDR) [20] and Average Precision (AP%) of classified shading pixel regions [87] have been proposed. Yet another way to evaluate IID solutions is via the effectiveness of the decomposed components in a downstream application. Bonneel et al. [23] propose hardcoded application scenarios like logo removal, shadow removal, texture replacement, and wrinkles attenuation on a fixed set of hand-picked 21 images to benchmark IID solutions. These evaluation strategies either require dense GT annotations which are available only for synthetic scenes [23, 27] (with exception of a few single object images from Grosse et al. [65]) or are dataset specific with sparse human annotations [20] [87]. Since multiple R and S pairs can result in the same image, even the "ground truth" is only one possible solution and measures on that is inadequate to evaluate the method.

To address these issues, we propose a new evaluation strategy for ill-posed problems like IID by measuring the *quality of disentanglement* between the decomposed components R and S. We use the core IID concepts of illumination-invariance of R and albedo-invariance of S to measure disentanglement, without specifically relying on synthetic images or relative quality metrics computed on fixed sparsely annotated datasets. We choose an ML interpretability technique based upon Testing with Concept Activation Vectors (TCAV) [80] for this. Originally introduced for classifiers, TCAV is a post-hoc Concept-based Model Extraction (CME) [77] method that interprets a Neural Network using human-understandable concepts. Specifically, TCAV quantifies the importance of a user-defined concept in the model's prediction by extracting activation vectors from a provided *concept set*. For example, for a zebra classifier one may be interested in interpreting concepts like ‘striped-ness’ *vs.*‘dotted-ness’, which are defined by learning *Concept Activation Vectors* w.r.t.the model from user provided sets with striped and dotted textures respectively. We use as concepts two core characteristics derived from the very definition of IID, *i.e.*, illumination-invariance of R and albedo-invariance of S. We assess disentanglement between them by measuring the model’s sensitivity to these concepts in the form of *Concept Sensitivity Metrics* (CSM) (Fig. 3.1). The CSM provides a generic framework applicable to problems other than IID using concepts relevant to them. To summarise, the main contributions our work are:

- A novel method for using ML interpretability algorithms like TCAV to measure disentanglement.
- A novel IID performance evaluation metric: Concept Sensitivity Metric (CSM) and benchmarked results on three state-of-the-art IID solutions.
- A new configurable dataset of images and corresponding generation scripts with controlled illumination and albedo variation.

3.2 Concept Sensitivity Metric: IID Evaluation by Interpretation

In this section, we first provide a quick primer on the background of Testing with CAV [80], followed by our definition of IID *concept sets* for R illumination-invariance and S albedo-invariance and proposed IID *evaluation strategy* using our novel Concept Sensitivity Metric.

Concepts used for IID: For IID evaluation-via-interpretation, we define two concepts: albedo-invariance (C_{Δ_i}) and illumination-invariance (C_{Δ_a}). For definition of associated concept sets (Δ_i and Δ_a), we render synthetic images of objects by varying one concept while fixing the other. We use random textures, albedo maps and different illumination settings for this. For negative concept set, we randomly select images from a large dataset (unrelated to IID).

R and S Sensitivity: By IID definition, in an ideal case Δ_a should only affect $\hat{R}$ and Δ_i should just affect $\hat{S}$. In other words, the sensitivity R_{Δ_a} of $\hat{R}$ towards concept C_{Δ_a} must be high and the sensitivity R_{Δ_i} towards concept C_{Δ_i} must be low. Conversely, for $\hat{S}$, the sensitivity S_{Δ_a} towards concept C_{Δ_a} must be low and the sensitivity S_{Δ_i} towards concept C_{Δ_i} must be high. In the ideal case of complete R-S disentanglement, R_{Δ_a}, S_{Δ_i} should be 1 and R_{Δ_i}, S_{Δ_a} should be 0. Due to several inherent assumptions in the IID definition (diffuse surfaces, linear optics, *etc.*), total disentanglement is impossible and a measure of disentanglement will be useful.

Concept Sensitivity Metric (CSM) We evaluate the model’s disentanglement quality by combining the above sensitivity scores to gauge the model’s performance in the two experiments separately to give Concept Sensitivity Metric (CSM) scores. CSM scores give a quantitative measure to gauge the quality of $\hat{R}$ *vs.*$\hat{S}$ disentanglement. We introduce two CSM scores: CSM_S which measures $\hat{S}$ albedo invariance

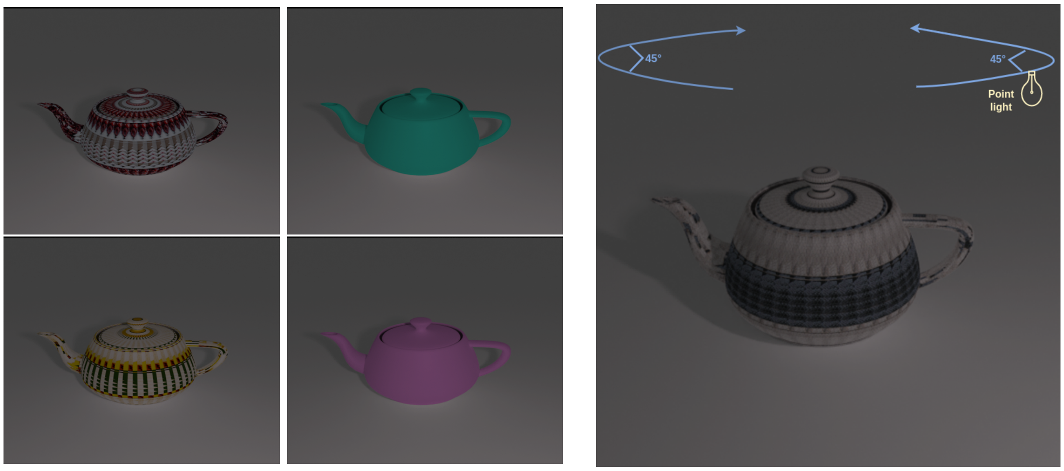


Figure 3.2: IID Concept sets: Left grid of images shows four samples from our Δ_a concept set with varying textures and base colors. Image on the right illustrates light source variation setting (±22 in either direction) for rendering Δ_i concept set images.

and CSM_R which measures $\hat{R}$ illumination invariance:

$$CSM_S = \frac{R_{\Delta_a}}{S_{\Delta_a}} \quad and \quad CSM_R = \frac{S_{\Delta_i}}{R_{\Delta_i}}. \tag{3.2}$$

Higher value of CSM_S indicates less leakage of albedo information in $\hat{S}$. Similarly, higher value of CSM_R indicates less illumination leakage in $\hat{R}$. We verify the same experimentally in section 3.4.

3.3 Experiments

3.3.1 Datasets

Concept Sets: For the two IID concepts, the respective concept sets (Δ_a and Δ_i) are rendered in Blender [40] in a controlled environment. We generate two types of scenes: *simple scenes* with a single object and *complex scenes* with multiple (3) objects. We setup the scene by randomly choosing from a set of frequently used standard 3D meshes[1] and placing them on a white table-top. We place a point light source of white color, 1000W power, 0.5m radius with fixed temperature $T \in \{2500, 4500, 6500\}$. Specifically, for Δ_a, we render scenes with fixed viewpoint and illumination but randomly vary the base color and surface texture. For Δ_i, we vary illumination by rotating the light source every 2° in ±45 left and right directions as shown in Fig. 3.2 but keep the camera viewpoint and base color/surface texture constant (thus getting 44 images: 22 in the left and 22 in right direction). For defining the negative concept set C', we take random images from Adobe-5k-dataset [28].

[1] https://github.com/alecjacobson/common-3d-test-models

We also compare the CAVs learned from our synthetically rendered concept sets against available natural scene datasets. For Δ_a concept set, we found no suitable large enough dataset with natural albedo variations, so we report on synthetic concept set only. For Δ_i we use two publicly available datasets which have natural illumination changes: *(i)* Multi-Illumination Images in the Wild dataset (**MIW**) [126]: test split which consists of 30 scenes under 25 different illuminations. *(ii)* Photometric Stereo dataset (**PS**) [6] consists of 3 objects (apple, gourd1, gourd2) under several illumination settings in all directions (approximately 100 per object). Though MIT Intrinsics dataset [65] also has scenes in varying illuminations (11), it cannot be used as concept set because at least 20 images are needed per concept for a stable CAV estimation as recommended by Kim et al. [80].

Testsets: We take ARAP [23] dataset images as our input x for IID networks. ARAP contains realistic synthetic renderings of both indoor and outdoor scenes. We remove the single object scenes ('Katie', 'redhead', 'skin', 'strawberries', 'toad', 'revolution') to maintain inter-scene consistency taking the remaining complex scenes to get a total of 42 scenes which have 3-4 varying illuminations.

3.3.2 Implementation details

We use our PyTorch [134] implementation of TCAV. We use the pre-trained models and the inference codes for IIWW [109], USI3D [116], and CGIID [108] from their official repositories and use their respective hyper-parameter settings. Our sensitivity computation requires only activation values of the pre-trained model and does not require full training. The hardware requirement of our framework depends on the model being analysed for CAV estimation. We tested our framework on 2 Nvidia GTX1080Ti GPUs, which were required by the largest model we analyzed ($USI3D$).

We perform multiple iterations (100) of CAV estimation experiments for robust concept definition. We decide upon the number of iterations through exhaustive experimentation as reported in section 3.4. In each iteration, we use 100 rendered images per set with varying albedo for C_{Δ_a} and 44 images with varying illumination for C_{Δ_i}. All images are resized to 256 × 256 dimensions. With each iteration, we perform hypothesis significance testing (double-sided t-test with $p = 0.01$ kept same as [80]) and average over the passing significant CAVs.

3.3.3 Experimental Details

We analyze our framework for different scenarios by enumerating over a combination of various experimental conditions:

- **Layer Selection:** Original CAV sensitivity can be evaluated for any layer of the model. In our IID adaptation, we restrict to the last layer sensitivities. There are two reasons for this design choice. First, the IID concepts which we are trying to capture are high-level abstractions which are better represented by the deeper layers. Second, different IID methods have different number of layers making the choice of comparable layers difficult across architectures *e.g.*IIWW and CGIID both have

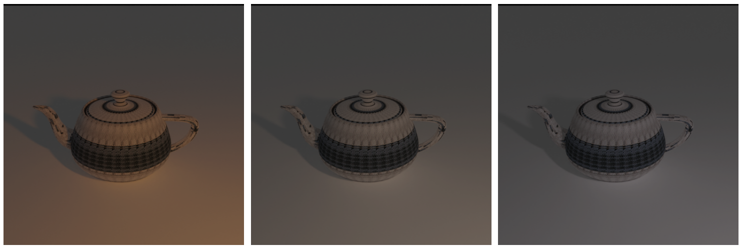


Figure 3.3: Illumination temperatures: We use three temperatures for illumination which are 2500, 4500, 6500 as shown in from left to right.

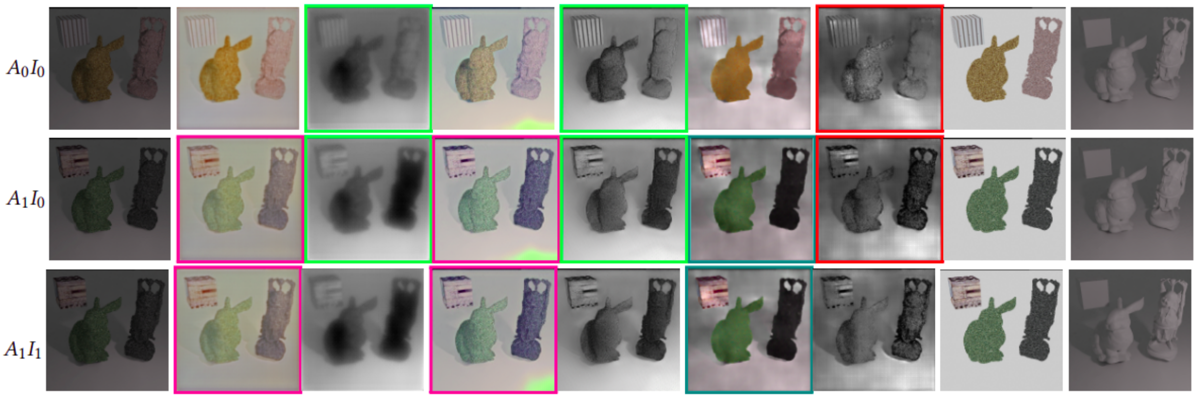
separate branches of R and S while USI3D has two generators of R and S styles plus a content encoder. Computing CAV sensitivities on the last layer makes our method more stable and architecture agnostic.

- **Scene Domain:** We analyze our technique under both synthetic and natural domains by choosing appropriate concept sets.
- **Concept Scene Complexity:** We estimate CAVs for both simple and complex scene settings with single object and multiple objects respectively.
- **Concept Albedo Complexity:** We render Δ_a concept sets with two albedo complexity settings: Simple RGB base color change and texture variation. In the first case, each object has one randomly assigned solid RGB color. In the second case, we apply a random texture map from DTD dataset [38] on each object.

Overall, we have four experimental settings for each concept: *Textured-Simple*, *Textured-Complex*, *RGB-Simple* and *RGB-Complex*. We perform comprehensive experiments by forming multiple concept sets under each of above categories. Specifically, for C_{Δ_a} concept we have 10 scenes rendered under 4 different illumination directions with 3 illumination temperatures $t \in \{2500, 4500, 6500\}$ (as shown in Fig. 3.3), hence $10 \times 3 \times 4 = 120$ concept sets each containing 100 albedo/texture variation images. Similarly for C_{Δ_i} we have 10 scenes $\times\, 3$ temperatures $\times\, 3$ albedos $=$ 90 concept sets, each with 44 (22 left $+$ 22 right) illumination direction variation images.

3.4 Results and Analysis

We report our CSM disentanglement scores (CSM_S, CSM_R) for C_{Δ_a} and C_{Δ_i} in Tab. 3.1 after averaging over all the corresponding concepts sets. From Tab. 3.1, we find that USI3D does best disentanglement of albedo from $\hat{S}$ (best $\hat{S}$ albedo invariance thus highest CSM_S) while CGIID does best disentanglement of illumination information from $\hat{R}$ (highest CSM_R) amongst the three models. We also show illustrative qualitative results in Fig. 3.4 which show predicted $\hat{R}$ and $\hat{S}$ from the three methods for the same scene under 2 different albedo (A_0 and A_1) and illumination (I_0 and I_1) settings.



(a) Input (b) IIWW $\hat{R}$ (c) IIWW $\hat{S}$ (d) USI3D $\hat{R}$ (e) USI3D $\hat{S}$ (f) CGIID $\hat{R}$ (g) CGIID $\hat{S}$ (h) GT R (i) GT S

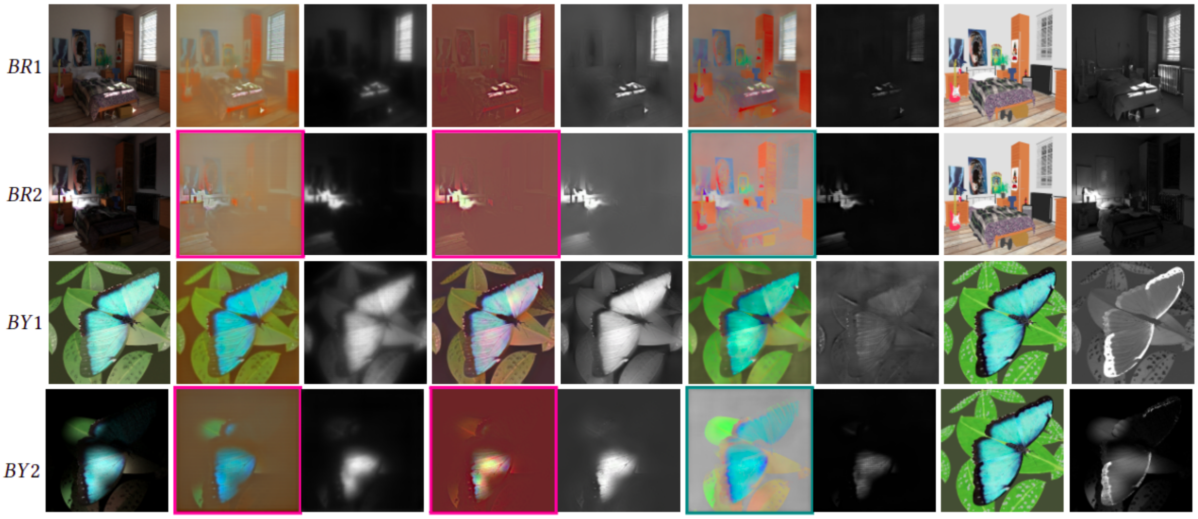
Figure 3.4: Illustrative qualitative results for albedo variation (first and second rows) and illumination variation (second and third row) experiments: A_iI_j represent scene in albedo i and illumination j. For albedo variation ($A_0 \rightarrow A_1$) , USI3D [116] followed by IIWW [109] observes least changes in $\hat{S}$ (green) and thus is best in disentangling albedo information from $\hat{S}$ while CGIID [108] (magenta) is worst. For illumination variation ($I_0 \rightarrow I_1$) CGIID observes least changes in $\hat{R}$ (teal) having least leakage of shadows in $\hat{R}$ for all three rows while IIWW and USI3D have comparatively more shadow leakage (magenta). This is reflected in our CSM_S and CSM_R scores in Tab. 3.2 unlike in existing metrics (Tab. 3.4). (Best viewed in color).

Table 3.1: Disentanglement quality: Quality of disentanglement for albedo variation (as measured by CSM_S) and illumination variation (as measured by CSM_R). USI3D performs best in CSM_S metric and CGIID the worst. The trend is reversed for the CSM_R metric. Note: Best is **bold**, second best is underlined.

	$CSM_S \uparrow$					$CSM_R \uparrow$					$WHDR \downarrow$	$SAW AP\% \uparrow$
Model	**Textured**		**RGB**			**Textured**		**RGB**				
	Simple	**Complex**	**Simple**	**Complex**	$\overline{CSM_S} \uparrow$	**Simple**	**Complex**	**Simple**	**Complex**	$\overline{CSM_R} \uparrow$		
IIWW [109]	**2.049**	1.286	0.939	**1.732**	1.524	0.857	0.979	0.936	0.741	0.878	20.3	91.87
USI3D [116]	2	**1.943**	**2.806**	1.669	**2.139**	0.648	0.504	0.38	0.674	0.552	18.69	78.69
CGIID [108]	0.753	1.328	0.606	0.93	0.909	**1.35**	**1.231**	**1.806**	**1.79**	**1.544**	**14.8**	**97.93**

Table 3.2: Disentanglement quality in different temperatures: On an average a similar trend is followed across temperatures.

Temp	Model	$CSM_S \uparrow$					$CSM_R \uparrow$				
		Textured		**RGB**		$\overline{CSM_S} \uparrow$	**Textured**		**RGB**		$\overline{CSM_R} \uparrow$
		Simple	**Complex**	**Simple**	**Complex**		**Simple**	**Complex**	**Simple**	**Complex**	
2500	IIWW [109]	1.735	1.112	0.894	1.426	1.292	1.021	1.098	0.994	0.757	0.968
	USI3D [116]	**4.138**	**2.802**	**4.11**	**2.643**	**3.423**	0.449	0.406	0.275	0.646	0.444
	CGIID [108]	0.901	1.379	0.554	1.051	0.971	**1.166**	**1.412**	**1.52**	**1.66**	**1.44**
4500	IIWW [109]	1.995	1.544	0.955	**1.877**	1.593	0.805	0.979	0.886	0.771	0.86
	USI3D [116]	**2.352**	**2.248**	**2.697**	1.54	**2.209**	0.646	0.497	0.385	0.645	0.543
	CGIID [108]	0.745	1.334	0.706	0.9	0.921	**1.347**	**1.282**	**1.858**	**1.862**	**1.587**
6500	IIWW [109]	**2.5**	1.248	0.965	**1.969**	**1.671**	0.763	0.879	0.93	0.693	0.816
	USI3D [116]	1.091	**1.285**	**2.063**	1.253	1.423	0.942	0.74	0.512	0.727	0.73
	CGIID [108]	0.611	1.278	0.558	0.837	0.821	**1.624**	**1.295**	**2.147**	**1.865**	**1.733**



(a) Input (b) IIWW $\hat{R}$ (c) IIWW $\hat{S}$ (d) USI3D $\hat{R}$ (e) USI3D $\hat{S}$ (f) CGIID $\hat{R}$ (g) CGIID $\hat{S}$ (h) GT R (i) GT S

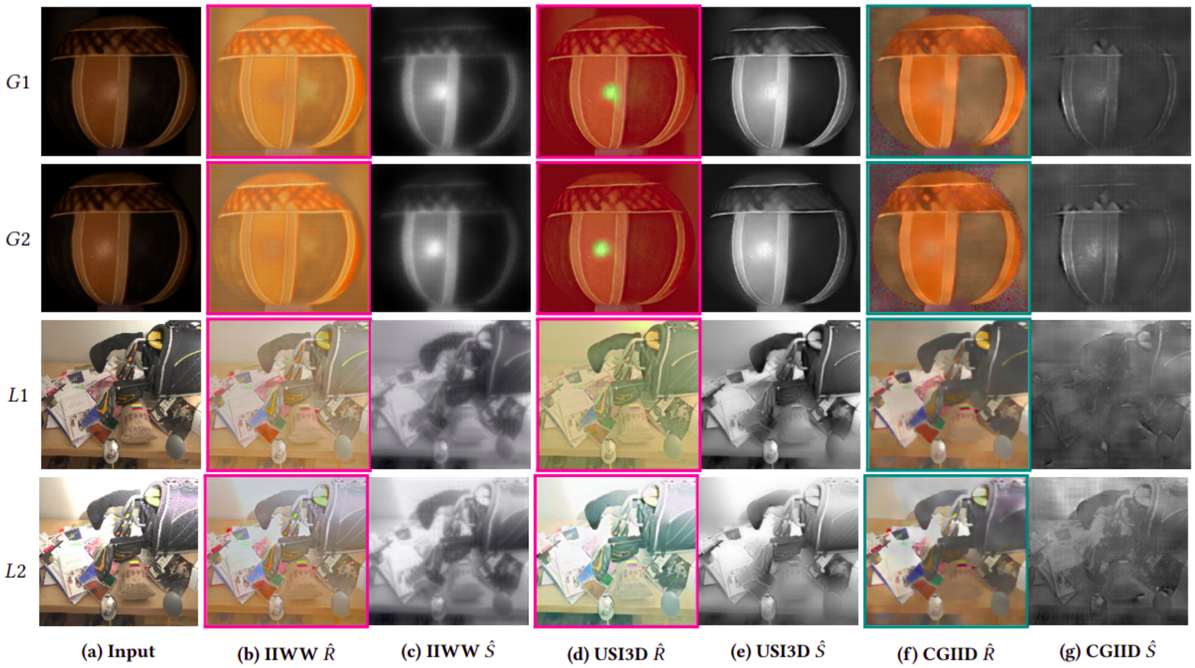
Figure 3.5: Illumination changes in ARAP dataset scenes: The first and second rows are bedroom scene in different illuminations (BR1, BR2) while third and fourth are butterfly in different illuminations (BY1, BY2). For both the scenes CGIDD has least leakage of illumination in $\hat{R}$. In BR1 and BR2, the shadow information is leaked more in $\hat{R}$ for both IIWW and USI3D (magenta), while it is leaked lesser for CGIDD (green). From $BY1 \rightarrow BY2$, butterfly's top wing has a drastic illumination change. CGIDD predicts this top wing's albedo correctly to huge extent(green) while IIWW and USI3D have that illumination directly leaked in $\hat{R}$ (magenta).

Table 3.3: Natural concept datasets: Similar to synthetic case in Tab. 3.1, CGIID significantly performs the best in CSM_R followed by IIWW and USI3D on both natural image datasets. This shows no significant shift between the two domains for CAV computation.

Model	**MIW [126]**			**PS [6]**			$\overline{CSM_R} \uparrow$
	$R_i \downarrow$	$S_i \uparrow$	$CSM_R \uparrow$	$R_i \downarrow$	$S_i \uparrow$	$CSM_R \uparrow$	
IIWW [109]	0.391	0.296	0.757	0.409	0.501	1.225	$\underline{0.991}$
USI3D [116]	0.173	0.133	0.751	0.75	0.612	0.816	0.784
CGIID [108]	0.373	0.462	**1.239**	0.061	0.587	**9.623**	**5.431**

Table 3.4: Limitation of standard IID metrics: For two exemplar images A_0I_0 and A_1I_0 in Fig. 3.4, USI3D exhibits best performance for $\hat{R}$ although it contains shadow leakages. Whereas for the $\hat{S}$ component, the metrics contradict each other.

Img	**Model**	**MSE↓**		**LMSE↓**		**D-SSIM↓**		**LPIPS↓**[197]	
		R	**S**	**R**	**S**	**R**	**S**	**R**	**S**
A_0I_0	IIWW [109]	0.025	0.021	0.006	0.004	0.254	**0.365**	0.336	0.429
	USI3D [116]	**0.013**	0.058	0.006	0.006	**0.188**	0.431	**0.185**	**0.396**
	CGIID [108]	0.021	**0.02**	0.006	**0.003**	0.28	0.421	0.364	**0.396**
A_1I_0	IIWW [109]	0.035	**0.041**	0.011	0.011	0.261	**0.456**	0.362	0.439
	USI3D [116]	**0.014**	0.07	0.006	**0.01**	**0.162**	0.495	**0.153**	**0.406**
	CGIID [108]	0.062	**0.041**	0.006	0.014	0.327	0.559	0.388	0.443



(a) Input (b) IIWW $\hat{R}$ (c) IIWW $\hat{S}$ (d) USI3D $\hat{R}$ (e) USI3D $\hat{S}$ (f) CGIID $\hat{R}$ (g) CGIID $\hat{S}$

Figure 3.6: Real-world illumination change results: First two rows are images from Gourd scene (G) which belongs to PS [6] dataset, last two rows are scene lobby (L) from MIW[126]. For G1 and G2, $\hat{R}$ illumination variance is most in USI3D (circular luminant) followed by IIWW while it is least by CGIID. Though CGIID is unable to predict good $\hat{R}$ for the dark background of G1 and G2, its foreground's $\hat{R}$ is good. In lobby scene (L1 and L2) USI3D has global intensity changes in $\hat{R}$ and IIWW also has a more changes compared to CGIID which has lesser changes (as seen in black bag on right). Thus CGIID's $\hat{R}$ is less sensitive to Δ_i followed by IIWW and then USI3D (which performs worst).

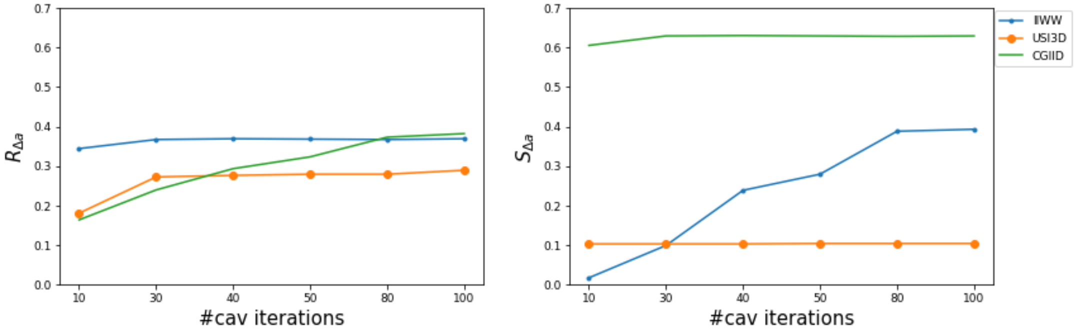
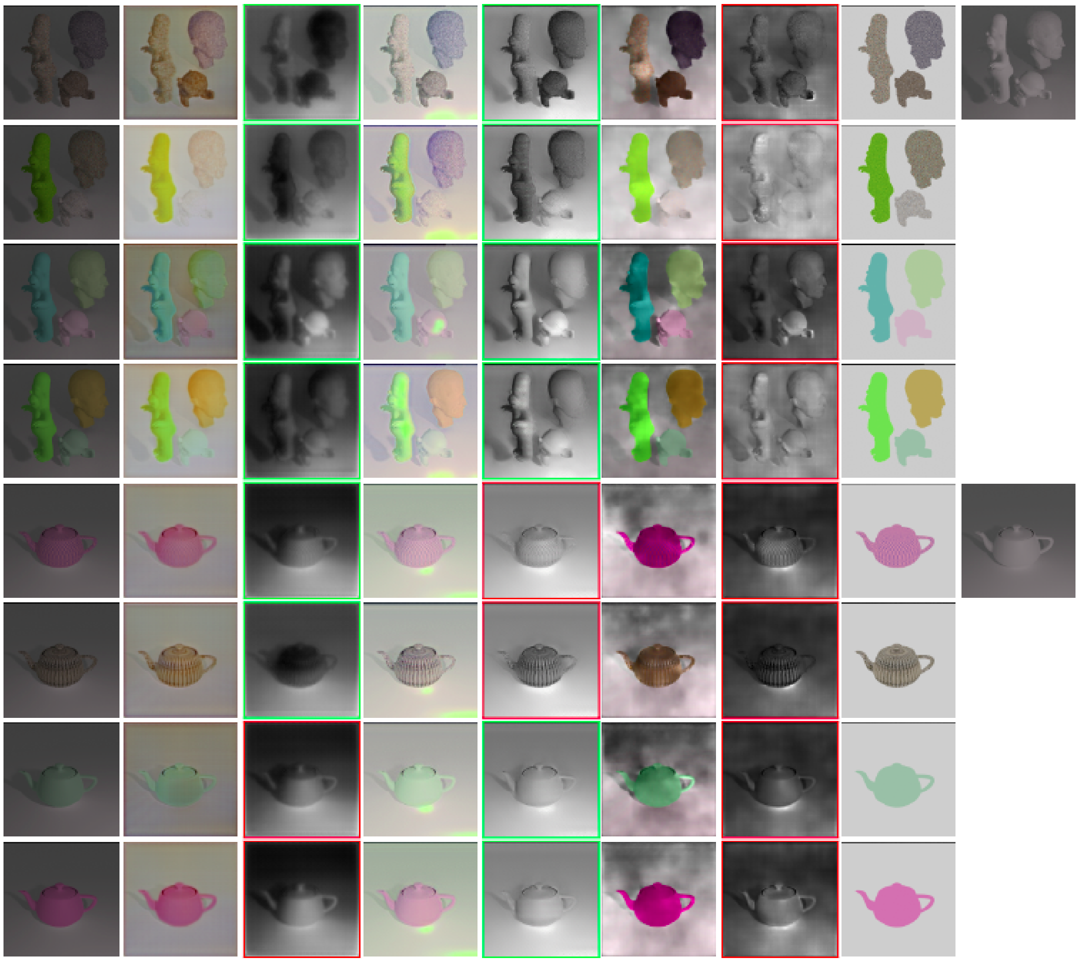
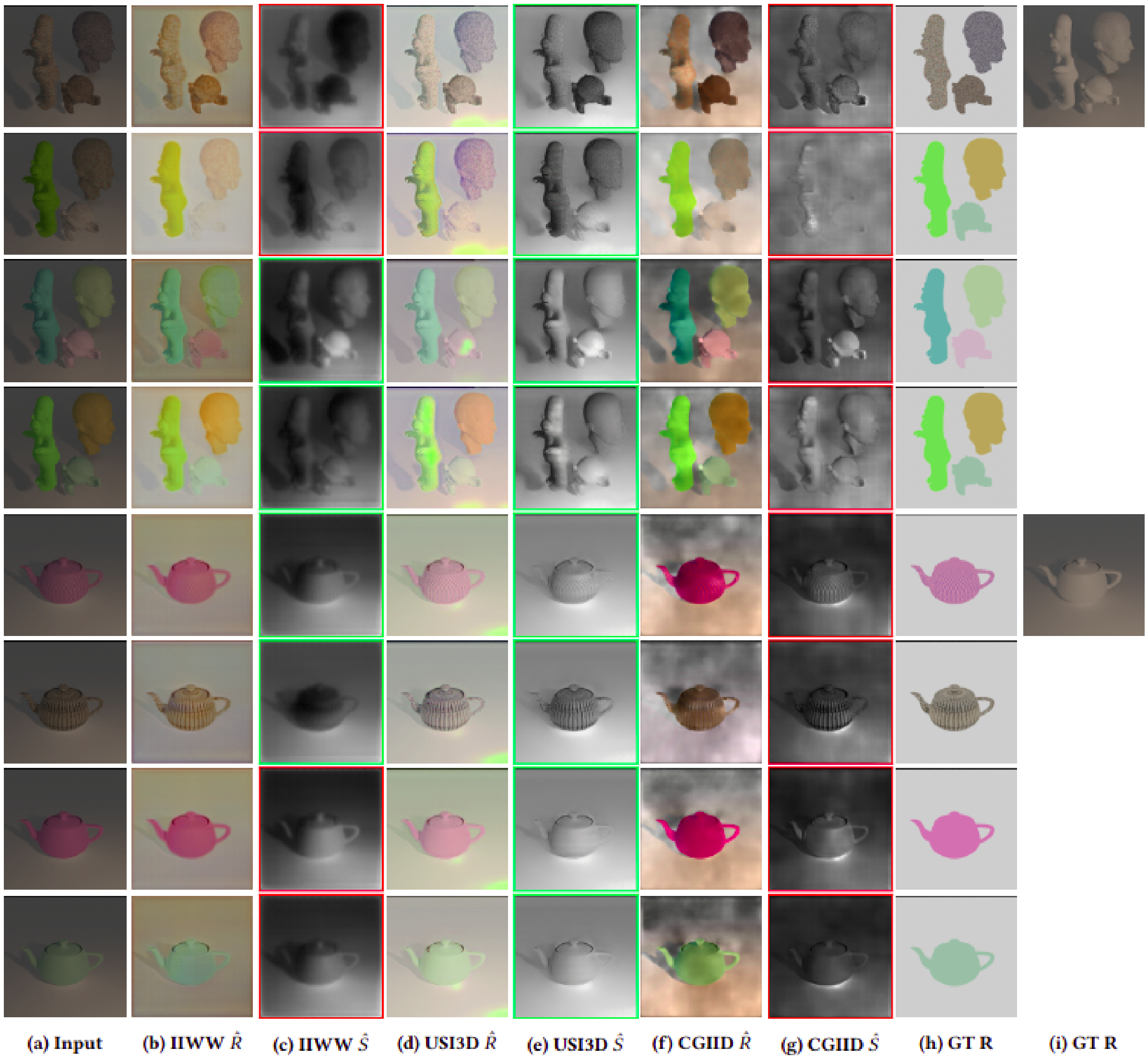


Figure 3.7: Sensitivity scores with number of CAV iterations for RGB-Simple concepts C_{Δ_a}. Both sensitivity scores R_{Δ_a} and S_{Δ_a} plateau after 80 iterations for all three models. Note: S_{Δ_a} for USI3D is constantly low since C_{Δ_a} is in-significant for its $\hat{S}$.



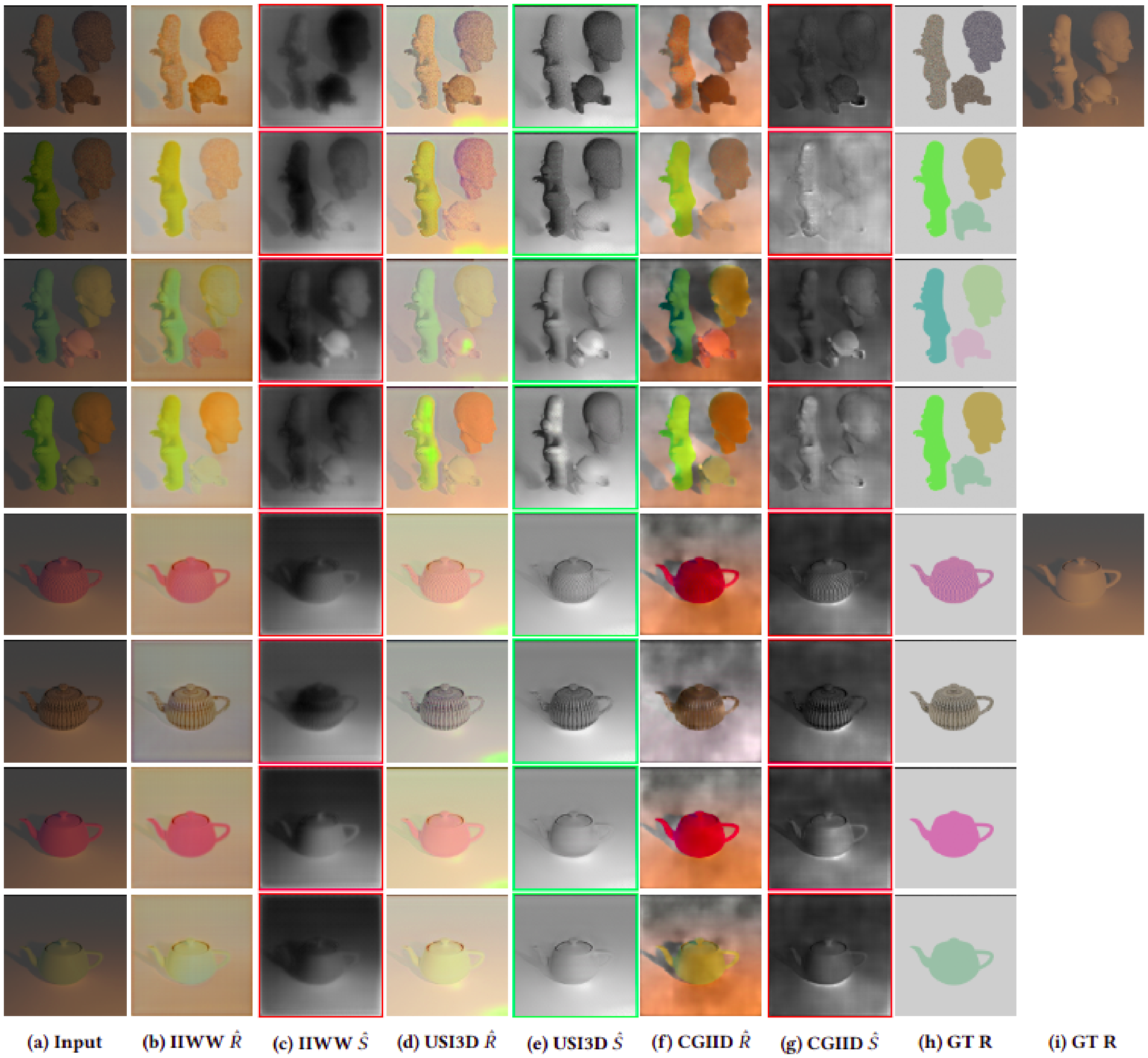
(a) Input (b) IIWW $\hat{R}$ (c) IIWW $\hat{S}$ (d) USI3D $\hat{R}$ (e) USI3D $\hat{R}$ (f) USI3D $\hat{S}$ (g) CGIID $\hat{S}$ (h) GT R (i) GT S

Figure 3.8: Albedo change experiments for T = 6500: Overall performance order is: USI3D=IIWW>>CGIID. Rows 1 and 2 are scenes in Textured-Complex setting: USI3D has the least S_{pred} variations followed by IIWW and CGIID which has significant global changes (shading intensity variations from light to dark). Also, CGIID's R_{pred} for the second row is very smooth, and most texture information is leaked in S_{pred}. The third and fourth rows have RGB-complex Δ_a: IIWW followed by USI3D have fewer changes in S_{pred} compared to CGIID which observes global changes. Fifth and sixth rows are Textured-Simple Δ_a: IIWW observes the least changes in S_{pred} over teapot followed by USI3D. CGIID and IIWW have significant S_{pred} variations in the background. For the second last and last rows which is RGB-Simple Δ_a setting: USI3D has a nearly constant $\hat{S}$, while IIWW has significant variations in the background(at the top) where intensity changes become darker and CGIID has shading intensity variations over teapot).



(a) Input (b) IIWW $\hat{R}$ (c) IIWW $\hat{S}$ (d) USI3D $\hat{R}$ (e) USI3D $\hat{S}$ (f) CGIID $\hat{R}$ (g) CGIID $\hat{S}$ (h) GT R (i) GT R

Figure 3.9: Albedo change experiments for T = 4500 Order of performance: USI3D > IIWW > CGIID. Note: Since shading is constant, we represent shading for rows 1, 2, 3, and 4 in row 1 and rows 5, 6, 7, and 8 in row 5.



(a) Input (b) IIWW $\hat{R}$ (c) IIWW $\hat{S}$ (d) USI3D $\hat{R}$ (e) USI3D $\hat{S}$ (f) CGIID $\hat{R}$ (g) CGIID $\hat{S}$ (h) GT R (i) GT R

Figure 3.10: Albedo change experiments for T = 2500 Order of performance: USI3D $>$ IIWW $>$ CGIID. Note: Since shading is constant, we represent shading for rows 1, 2, 3, and 4 in row 1 and rows 5, 6, 7, and 8 in row 5. Note: USI3D has sharper textures in S; hence in textured settings, textures might seem a bit change, but its light intensity is constantly compared to other models, which have both texture leakage and light intensity changes. IIWW has a smoother S leading to lesser texture leakage but has more light intensity changes(as seen from rows 5 and 6), while CGIID has both sharp texture leakages and light intensity changes. Hence USI3D gets a good CSM_S overall.

Performance in $\hat{S}$ albedo invariance: $\boldsymbol{CSM_S}$ Albedo variations for the same scene are rarely observed in the training sets. Due to this, supervised methods like CGIID perform poorly on CSM_S metric compared to unsupervised IIWW and USI3D. USI3D being completely unsupervised performs best, followed by IIWW which is partially unsupervised (assuming constant reflectance over time-lapse videos of varying illumination scenes). From Fig. 3.4, USI3D has least changes in $\hat{S}$ for Δa.

Performance in $\hat{R}$ illumination invariance: $\boldsymbol{CSM_R}$ CGIID has significantly higher CSM_R in all the four experimental settings, followed by IIWW and then USI3D as shown in Tab. 3.1 and verified from qualitative results in Fig. 3.4 where CGIID observes least changes in R for illumination variations. It also does well on real-world concept sets as seen from Tab. 3.3 and qualitative results shown in Fig. 3.6. The same trend is seen over complex scenes from ARAP dataset in Fig. 3.5 and MIT Intrinsics [65]. This is because illumination variations are captured to some extent in existing IID datasets and hence the concept C_{Δ_i} can be learned by supervision. Thus, CGIID being a completely supervised network, performs well on CSM_R metric. IIWW being trained on time-lapse videos of BigTime dataset [109] comes next, followed by USI3D which is completely unsupervised and relies on style distributions of R and S. The same trend is seen in MIT Intrinsics [65] (shown in qualitative results in the supplementary video).

Effect of different temperature settings: We show results on different T in Tab. 3.2. For CSM_S, the same trend is observed for T = 2500 and 4500: USI3D>IIWW>>CGIID, while for T=6500, IIWW is slightly better than USI3D (IIWW>USI3D>CGIID). With an increase in T, its occurrence gets rare in training sets. Thus, USI3D being an unsupervised method, does significantly better than its partially supervised (IIWW) and supervised (CGIID) counterparts. For T = 6500, IIWW has slightly better CSM_S, the reason being its training on illumination varying BigTime dataset having temperatures in that range (most natural images are near 6500 T). For CSM_R, same overall ordering of models is observed across temperatures: CGIID>IIWW>USI3D. We provide additional results on Δ_a concept for each of our 4 experimental settings for T = 6500, 4500 and 2500 in Figures 3.8, 3.9 and 3.10 respectively.

Effect of number of CAV iterations: We experiment with the number of iterations for stable CAV verification by t-testing as pointed in section 2.1.2 and find that any iterations above 80 work well (Fig. 3.7). We thus take 100 as our number of cav iterations. Individual comparisons of $\hat{R}$ and $\hat{S}$ with GT only consider how close $\hat{R}$, $\hat{S}$ are to GT and do not consider disentanglement. $\hat{R}$ and $\hat{S}$ closeness to GT does not guarantee disentanglement since the reconstruction can be good enough but entain illumination-albedo leakages. As shown in Tab. 3.4, $\hat{R}$ gets better MSE, LMSE, D-SSIM and LPIPS [197] values but still has shadow leakages because it resembles GT the most, except for pixels where shadows are there. USI3D on the other hand, achieves best performance in $\hat{R}$ in terms of existing metrics on MIT Intrinsics, ARAP as well as our synthetic concept sets, but it has clear illumination leakages in $\hat{R}$ for which its score is penalised by our method which gauges disentanglement.

Table 3.5: Pixel-wise comparison metrics on MIT Intrinsics datataset[65]

Model	MSE↓		LMSE↓		DSSIM↓	
	R	**S**	**R**	**S**	**R**	**S**
IIWW	**0.0147**	0.0135	0.0341	0.0253	0.1398	**0.1266**
USI3D	0.0156	**0.0102**	0.064	0.0474	**0.1158**	0.131
CGIID	0.0167	0.0127	**0.0319**	**0.0211**	0.1287	0.1376

Table 3.6: Pixel-wise comparison metrics on 42 scenes of ARAP dataset[23].

Model	MSE↓		LMSE↓		D-SSIM↓	
	R	**S**	**R**	**S**	**R**	**S**
IIWW	**0.056**	0.033	0.066	0.054	**0.448**	0.522
USI3D	0.095	**0.021**	0.072	**0.052**	0.486	**0.347**
CGIID	0.073	0.037	**0.064**	0.054	0.512	0.498

WHDR and SAW AP% are not designed for measuring R-S disentanglement by their very definition, which can be seen from Tab. 3.1 where they don't align with CSM_R and CSM_S along with qualitative results as shown in Fig. 3.4, 3.5, 3.6. Ideally R, and S must be disentangled by the definition of IID but must also resemble GT. For example, though CGIID is best in terms of disentangling illumination information from $\hat{R}$, the closeness of $\hat{R}$ to GT achieved best by USI3D is important as well (Fig. 3.4 Tab. 3.4). Hence, our method must be combined with existing GT comparison metrics for the most robust IID evaluation.

Cross-dataset performance: USI3D has significantly lower performance on SAW AP% metric which measures the quality of $\hat{S}$, but it exhibits the best performance in terms of pixel-wise comparisons on MIT Intrinsics[65], ARAP dataset[23], and our synthetic concept set. This highlights the issue of cross-dataset performance evaluation inconsistency for the standard IID metrics. On the other hand our proposed CSM_R metric performs consistently and exhibits the same trend in performance: $CGIID > IIWW > USI3D$ (see $\overline{CSM_R}$ in Tab. 3.1 and Tab. 3.3)

Existing pixel-wise metrics: We report the D-SSIM, LMSE and MSE metrics over MIT Intrinsics dataset[65], ARAP[23] and our newly introduced Δ_a and Δ_i concept sets in Tab. 3.5, Tab. 3.6 and Tab. 3.7 respectively. Each of these metrics measures different aspects of closeness to Ground Truth. MSE measures the average squared pixel-wise difference between predicted and GT. LMSE is local-

Table 3.7: Pixel-wise comparison metrics on scenes of our concept sets: USI3D does best in general in terms of the above metrics.

Scene type	**Albedo type**	**Model**	**Δ_a concept set**						**Δ_i concept set**					
			R			**S**			**R**			**S**		
			MSE	**LMSE**	**D-SSIM**	**MSE**	**LMSE**	D-SSIM	**MSE**	**LMSE**	**D-SSIM**	**MSE**	**LMSE**	**D-SSIM**
Simple	RGB	IIWW	0.116	0.027	0.277	0.053	0.006	0.409	0.145	0.031	0.339	0.021	0.006	0.415
		USI3D	**0.06**	**0.025**	**0.213**	0.041	**0.001**	**0.234**	**0.067**	**0.021**	**0.23**	0.035	**0.004**	**0.198**
		CGIID	0.134	0.033	0.344	**0.014**	0.002	0.265	0.183	0.04	0.426	**0.012**	0.005	0.23
	Textured	IIWW	0.057	0.012	0.208	0.075	0.005	0.348	0.076	0.015	0.337	0.071	**0.004**	0.355
		USI3D	**0.026**	**0.008**	**0.138**	0.054	0.005	0.326	**0.041**	**0.008**	**0.284**	0.045	0.007	0.31
		CGIID	0.062	0.016	0.255	**0.018**	**0.002**	**0.287**	0.08	0.019	0.359	**0.016**	0.005	**0.269**
Complex	RGB	IIWW	0.07	0.014	0.246	0.057	0.006	0.382	0.09	0.016	0.306	0.059	0.006	0.398
		USI3D	**0.024**	**0.007**	**0.19**	0.039	**0.003**	**0.268**	**0.034**	**0.006**	**0.228**	0.035	**0.004**	**0.23**
		CGIID	0.082	0.019	0.294	**0.022**	**0.003**	0.308	0.128	0.027	0.396	**0.02**	0.005	0.272
	Textured	IIWW	0.044	0.014	0.25	0.064	0.006	0.383	0.048	0.014	0.322	0.112	**0.005**	**0.414**
		USI3D	**0.019**	**0.009**	**0.184**	0.062	0.007	0.4	**0.031**	**0.01**	**0.31**	0.071	0.01	0.449
		CGIID	0.046	0.014	0.284	**0.02**	**0.005**	**0.372**	0.05	0.013	0.335	**0.024**	0.007	0.421

MSE and measures MSE patch-wise, while D-SSIM gives structural dis-similarity between predicted and GT images. MSE measures absolute error, not taking spatial information of pixels into account, whereas LMSE and D-SSIM consider spatially close pixels separately. On MIT Intrinsics[65] all models have comparable performance Tab. 3.5 and there is no common trend of performance established as such.

Raw sensitivity scores: Additionally, we report the raw sensitivity scores of experiments in Tab. 3.8. Note these scores have been used to calculate our CSM disentanglement scores as mentioned above. We observe that these scores are high for both R and S by models in some experiments. The concept captured by CAV vectors is dependent on the model's activations and some models might be affected by that concept for both R and S and hence the raw sensitivity scores by themselves do not provide much information about the importance given by the model for $\hat{R}$ vs $\hat{S}$. Note: TCAV does analysis in classification problems and thus calculates sensitivity over classifiers while we are using it in a decomposition(reconstruction problem) for a multi-branch network. Hence the sensitivity scores are standalone for classifier setting as by Kim et al. [80] but not for our problem of comparing outputs of multi-branch network. We are interested in the R *vs.*. S sensitivity scores, hence the ratios of scores matter for us and not the raw scores.

CSMs measure necessary condition for IID Given ill-posed nature of IID wherein many R-S combinations can lead to same I, it is not straightforward or event possible to come up with a sufficient metric for evaluation of IID. CSMs effectively measure the R-S disentanglement, a crucial aspect of IID. This measurement is not just beneficial but a *necessary* condition for accurately assessing IID. Unlike direct ground truth (GT) comparisons, which have limitations in this context, CSMs provide a more

Table 3.8: TCAV sensitivity scores for concepts albedo change and illumination change in our 4 experimental settings.

		Δ_a								Δ_i							
Temp	**Model**	**Textured**				**RGB**				**Textured**				**RGB**			
		Simple		**Complex**		**Simple**		**Complex**		**Simple**		**Complex**		**Simple**		**Complex**	
		R_{Δ_a}	S_{Δ_a}	R_{Δ_a}	S_{Δ_a}	R_{Δ_a}	S_{Δ_a}	R_{Δ_a}	S_{Δ_a}	R_{Δ_i}	S_{Δ_i}	R_{Δ_i}	S_{Δ_i}	R_{Δ_i}	S_{Δ_i}	R_{Δ_i}	S_{Δ_i}
2500	IIWW	0.34	0.196	0.309	0.278	0.336	0.376	0.338	0.237	0.326	0.333	0.307	0.337	0.349	0.347	0.346	0.262
	USI3D	0.331	0.08	0.311	0.111	0.337	0.082	0.296	0.112	0.37	0.166	0.36	0.146	0.375	0.103	0.285	0.184
	CGIID	0.53	0.588	0.71	0.515	0.347	0.626	0.613	0.583	0.518	0.604	0.422	0.596	0.419	0.637	0.353	0.586
4500	IIWW	0.385	0.193	0.366	0.237	0.385	0.403	0.383	0.204	0.379	0.305	0.335	0.328	0.387	0.343	0.389	0.3
	USI3D	0.294	0.125	0.335	0.149	0.267	0.099	0.271	0.176	0.325	0.21	0.352	0.175	0.301	0.116	0.304	0.196
	CGIID	0.421	0.565	0.671	0.503	0.445	0.63	0.532	0.591	0.447	0.602	0.468	0.6	0.346	0.643	0.333	0.62
6500	IIWW	0.395	0.158	0.378	0.303	0.386	0.4	0.386	0.196	0.384	0.293	0.34	0.299	0.387	0.36	0.391	0.271
	USI3D	0.263	0.241	0.275	0.214	0.264	0.128	0.277	0.221	0.257	0.242	0.323	0.239	0.285	0.146	0.315	0.229
	CGIID	0.365	0.597	0.648	0.507	0.353	0.633	0.487	0.582	0.375	0.609	0.468	0.606	0.3	0.644	0.327	0.61
Avg	IIWW	0.373	0.182	0.351	0.273	0.369	0.393	0.369	0.213	0.363	0.311	0.327	0.322	0.374	0.35	0.375	0.278
	USI3D	0.296	0.148	0.307	0.158	0.289	0.103	0.282	0.169	0.318	0.206	0.345	0.187	0.321	0.122	0.301	0.203
	CGIID	0.439	0.583	0.676	0.509	0.382	0.63	0.544	0.585	0.448	0.605	0.453	0.601	0.355	0.641	0.338	0.605

nuanced and effective evaluation method. They capture the subtleties of R-S disentanglement better than a straightforward GT comparison. To maximize the effectiveness of IID evaluation, it's advisable to use CSMs in conjunction with existing ground truth comparison metrics. This combined approach leverages the strengths of both methods, offering a more comprehensive and accurate assessment of IID performance.

Limitations of our concept sensitivity based approach: Models can be sensitive to concepts in ways that are not desirable. For example, a model might predict noisy R which keeps on changing (as seen from black foggy artifacts in CGIID predictions in Fig. 3.4) and get a high R_{Δ_a} score. Similarly, artifacts in $\hat{S}$ might lead to high sensitivity. In such rare cases, usually, R and S both have noise and taking the ratios cancels out and gives a lower score to the model.

3.5 Conclusion

We presented Concept Sensitivity Metric, a framework that adapts an ML interpretability method, to evaluate the quality of IID based on its definition. The CSM_R and CSM_S metrics evaluate the disentanglement of the recovered reflectance and shading. These metrics overcome several shortcomings of the current IID evaluation strategies. They are consistent over real-world and synthetic scenes and have lesser dependence on the evaluation set as we use model's sensitivity towards concepts rather than direct pixel-to-pixel comparison with ground truth annotations.

Since these metrics measure the quality of the output and can provide additional terms to the loss being minimized to improve the IID calculations like in a fine-tuning step. We intend to work on this in the future. The use of metrics defined for interpretability in a loop to improve the performance on the original problem has a wide scope of applicability.

The approach underlying Concept Sensitivity Metric has wider potential application beyond the IID problem. Choosing appropriate concepts and their activations, CSM can be used to evaluate results of image harmonization, style transfer, image enhancement, *etc*.

Chapter 4

Concept Distillation: Leveraging Human-Centered Explanations for Model Improvement

4.1 Introduction

EXplainable Artificial Intelligence (XAI) methods are useful to understand a trained model's behavior [199]. They open the black box of Deep Neural Networks (DNNs) to enable post-training identification of unintended correlations or biases using similarity scores or saliency maps. Humans, however, think in terms of abstract *concepts*, defined as groupings of similar entities [73]. Recent efforts in XAI have focused on concept-based model explanations to make them more aligned with human cognition. Kim et al. [80] introduce Concept Activation Vectors (CAVs) using a concept classifier hyperplane to quantify the importance given by the model to a particular concept. For instance, CAVs can determine the model's sensitivity on 'striped-ness' or 'dotted-ness' to classify Zebra or Cheetah using user-provided concept samples. They measure the concept sensitivity of the model's final layer prediction with respect to intermediate layer activations (outputs). Such post-hoc analysis can evaluate the transparency, accountability, and reliability of a learned model [32] and can identify biases or unintended correlations acquired by the models via shortcut learning [9, 80].

The question we ask in this work is: If CAVs can identify and quantify sensitivity to concepts, can they also be used to improve the model? Can we learn less biased and more human-centered models? In this work, we extend CAVs to ante-hoc model improvement through a novel *concept loss* to desensitize/sensitize against concepts. We also leverage the broader conceptual knowledge of a large pre-trained model as a teacher in a *concept distillation* framework for it.

XAI has been used for ante-hoc model improvement during training [7, 77, 83]. They typically make fundamental changes to the model architecture or need significant concept supervision, making extensions to other applications difficult. For example, Koh et al. [83] condition the model by first predicting the underlying concept and then using it for class prediction. Our method can sensitize or desensitize the model to user-defined concepts without modifications to the architecture or direct supervision.

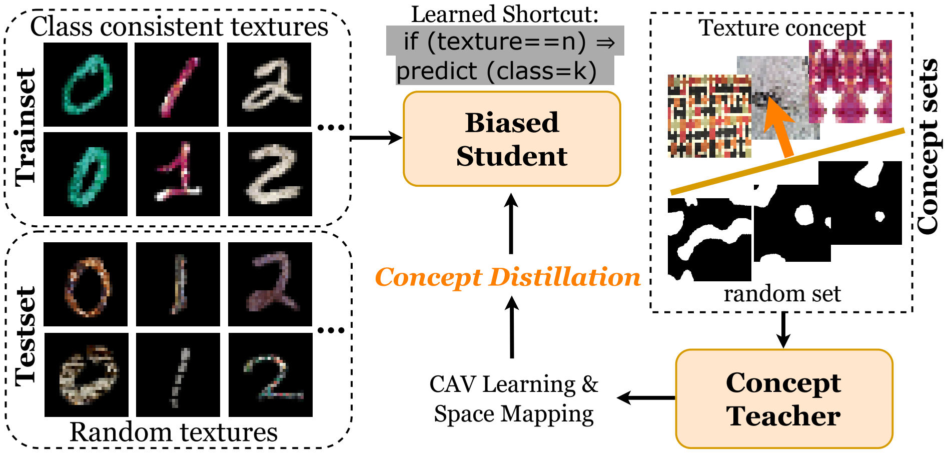


Figure 4.1: Overview of our approach: The generic conceptual knowledge of a capable teacher can be distilled to a student for performance improvement through bias removal and prior induction.

Our approach relies on the sensitivity of a trained model to human-specified concepts. We want the model to be sensitive to relevant concepts and indifferent to others. For instance, a cow classifier might be focusing excessively on the grass associated with cow images. If we can estimate the sensitivities of the classifier to different concepts, we can steer it away from irrelevant concepts. We do that using a *concept loss* term L_C and fine-tuning the trained base model with it. Since the base models could be small and biased in different ways, we use *concept distillation* using a large, pre-trained teacher model that understands common concepts better.

We also extend concepts to work effectively on intermediate layers of the model, where the sensitivity is more pronounced. Kim et al. [80] measure the final layer's sensitivity to any intermediate layer outputs. They ask the question: if any changes in activations are done in the intermediate layer, what is its effect on the final layer prediction? They used the final layer's loss/logit to estimate the model sensitivity as their interest was to study concept sensitivities for interpretable model prediction. We, on the other hand, aim to fine-tune a model by (de)sensitizing it towards a given concept which may be strongest in another layer [4]. Thus, it is crucial for us to measure the sensitivity in *any* layer by evaluating the effect of the changes in activations in one intermediate layer on another. We employ prototypes or average class representations in that layer for this purpose. Prototypes are estimated by clustering the class sample activations [29, 102, 107, 131, 132, 175, 192]. Our method, thus, allows intervention in any layer.

In this work, we present a simple but powerful framework for model improvement using concept loss and concept distillation for a user-given concept defined in any layer of the network. We leverage ideas from post-hoc global explanation techniques and use them in an ante-hoc setting by encoding concepts as CAVs via a teacher model. Our method also admits sample-specific explanations via a local loss [139] along with the global concepts whenever possible. We improve state-of-the-art on classification problems like ColorMNIST and DecoyMNIST [9, 49, 138, 139], resulting in improved accuracies and generalization. We introduce and benchmark on a new and more challenging TextureMNIST dataset with texture bias associated with digits. We demonstrate concept distillation on two applications: *(i)* debiasing extreme biases on classification problems involving synthetic MNIST datasets [49, 105] and complex and sensitive age-*vs.*-gender bias in the real-world gender classification on BFFHQ dataset [81] and *(ii)* prior induction by infusing domain knowledge in the reconstruction problem of Intrinsic Image Decomposition (IID) [90] by measuring and improving disentanglement of albedo and shading concepts. We also release code and refer readers to our website[1] for more details.

To summarize, we:

- Extend CAVs from post-hoc explanations to ante-hoc model improvement method to sensitize/desensitize models on specific concepts without changing the base architecture.
- Propose a Novel Concept loss for concept sensitive finetuning of DNNs.
- Extend the model CAV sensitivity calculation from only final layer to *any* layer and enhance it by making it more global using prototypes.
- Introduce concept distillation to exploit the inherent knowledge of large pretrained models as a teacher in concept definition.
- Benchmark results on standard biased MNIST datasets and on a challenging TextureMNIST dataset that we introduce.
- Show application on a severely biased classification problem involving age bias.
- Show application beyond classification to the challenging multi-branch Intrinsic Image Decomposition problem by inducing human-centered concepts as priors. To the best of our knowledge, this is the first foray of concept-based techniques into non-classification problems.

4.2 Concept Guidance and Concept Distillation

Concepts have been used to explain model behavior in a post-hoc manner in the past. Response to abstract concepts can also demonstrate the model's intrinsic preferences, biases, etc. Can we use concepts to guide the behavior of a trained base model in desirable ways in an ante-hoc manner? We describe a method to add a *concept loss* to achieve this. We also present concept distillation as a way to take advantage of large foundational models with more exposure to a wide variety of images.

4.2.1 Concept Sensitivity to Concept Loss

Building on Kim et al. [80], we represent a concept C using a Concept Activation Vector (CAV) as the normal v_C^l to a linear decision boundary between concept samples C from others C' in a layer l of the model's activation space (Fig. 4.2). The model's sensitivity $S_{C,l}(\boldsymbol{x}) = \nabla L_o\left(f_l(\boldsymbol{x})\right) \cdot v_C^l$ to C is the directional derivative of final layer loss L_o for samples $\boldsymbol{x}$ along v_C^l [80]. The sensitivity score quantifies

[1] https://avani17101.github.io/Concept-Distilllation/

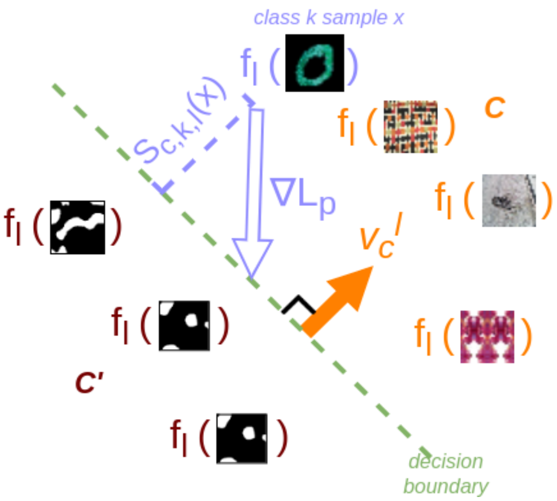


Figure 4.2: v_c^l is calculated as normal to the separating hyperplane of concept set activations (textures C *vs*.random set C' here). A model biased towards C will have its class samples's loss gradient ∇L_p along v_c^l (measured by sensitivity $S_{C,k,l}(x)$). To desensitize the model for C, we perturb ∇L_p to be parallel to the decision boundary by minimizing the cosine of the projection angle.

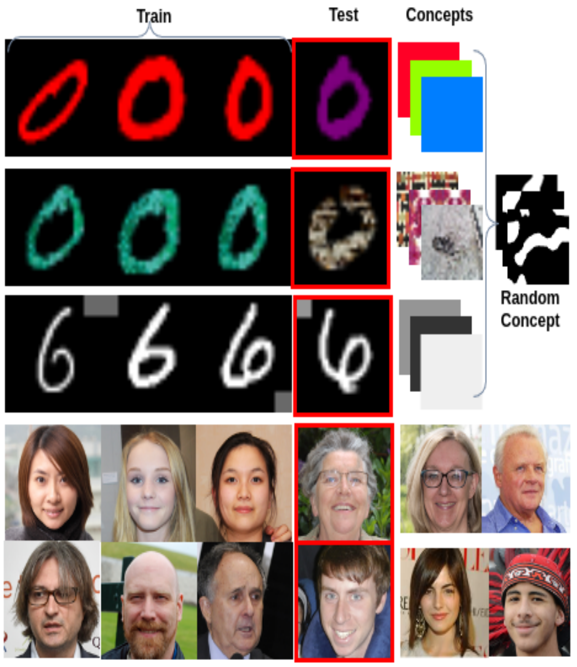


Figure 4.3: Datasets used: ColorMNIST (top row), TextureMNIST (next row), DecoyMNIST (third row), and BFFHQ (bottom rows). Concepts used include color, textured and gray patches, and bias-conflicting samples shown on the right.

the concept's influence on the model's prediction. A high sensitivity for color concept may indicate a color bias in the model.

These scores were used for post-hoc analysis before ([80]). We use them ante-hoc to desensitize or sensitize the base model to concepts by perturbing it away from or towards the CAV direction (Fig. 4.2). The gradient of loss indicates the direction of maximum change. Nudging the gradients away from the CAV direction encourages the model to be less sensitive to the concept and vice versa. For this, we define a concept loss L_C as the absolute cosine of the angle between the loss gradient and the CAV direction

$$L_C(\boldsymbol{x}) = |\cos(\nabla L_o\left(f_l(\boldsymbol{x})\right), \boldsymbol{v}_C^l)|, \tag{4.1}$$

which is minimized when the CAV lies on the classifier hyperplane (Fig. 4.2). We use the absolute value to not introduce the opposite bias by pushing the loss gradient in the opposite direction. A loss of $(1 - L_C(x))$ will sensitize the model to C. We fine-tune the trained base model for a few epochs using a total loss of $L = L_o + \lambda L_C$ to desensitize it to concept C, where L_o the base model loss.

4.2.2 Concepts using Prototypes

Concepts can be present in any layer l, though the above discussion focuses on the sensitivity calculation of the final layer using model loss L_o. The final convolutional layer is proven to learn concepts better than other layers [4]. We can estimate the concept sensitivity of any layer using a loss for that layer. How do we get a loss for an intermediate layer, as no ground truth is available for it?

Class prototypes have been used as pseudo-labels in intermediate layers before [29, 102, 192]. We adapt prototypes to define a loss in intermediate layers. Let $f_l(x)$ be the activation of layer l for sample x. We group the $f_l(x)$ values of the samples from each class into K clusters. The cluster centers P_i together form the prototype for that class. We then define prototype loss for each training sample x using the prototype corresponding to its class as

$$L_p(x) = \frac{1}{K}\sum_{k=1}^{K} |f_l(x) - P_k|^2. \tag{4.2}$$

We use L_p in place of L_o in Eq. 4.1 to define the concept loss in layer l. The prototype loss facilitates the use of intermediate layers for concept (de)sensitization. Experiments reported in Tab. 4.2 confirm the effectiveness of L_p. Prototypes also capture sample sensitivity at a global level using all samples of a class beyond the sample-specific levels. We update the prototypes after a few iterations as the activation space evolves. If P^n is the prototype at Step n and P^c the cluster centres using the current $f_l(x)$ values, the next prototype is $P_k^{n+1} = (1-\alpha)P_k^n + \alpha P_k^c$ for each cluster k.

4.2.3 Concept Distillation using a teacher

Concepts are learned from the base model in the above formulation. Base models may have wrong concept associations due to their training bias or limited exposure to concepts. Can we alleviate this problem using a larger model that has seen vast amounts of data as a teacher in a distillation framework?

We use the DINO [30], a self-supervised model trained on a large number of images, as the teacher and the base model as the student for concept distillation. The teacher and student models typically have different activation spaces. We map the teacher space to the student space before concept learning. The mapping uses an autoencoder [72] consisting of an encoder E_M and a decoder D_M (Fig. 4.4). As a first step, the autoencoder (Fig. 4.4) is trained to minimize the loss $L_{D_M} + L_{E_M}$. L_{D_M} is the pixel-wise L2 loss between the original (f_t) and decoded ($\hat{f_t}$) teacher activations and L_{E_M} is the pixel-wise L2 loss between the mapped teacher ($\hat{f_s}$) and the student (f_s) activations. The mapping is learned over the concept set of images C and C'. See the dashed purple lines in Fig. 4.4.

Next, we learn the CAVs in the distilled teacher space $\hat{f_s}$, keeping the teacher, student, and mapping modules fixed. This is computationally light as only a few (50-150) concept set images are involved. The learned CAV is used in concept loss given in Eq. 4.1. Please note that E_M is used only to align the two spaces and can be a small capacity encoder, even a single layer trained in a few epochs.

Algorithm 1 Concept Distillation Pipeline

Given: A pretrained student model which is to be fine-tuned for concept (de)sensitization and a pretrained teacher model which will be used for concept distillation. Known Concepts C and negative counterparts (or random samples) C', student training dataset $\mathcal{D}$, and student bottleneck layer l, #iterations to recalculate CAVs cav_update_frequency, #iterations to update prototypes proto_update_frequency.

```
Concept Distillation:
  For all class samples in $\mathcal{D}$, estimate class prototypes $P^0_{k \in \{0,K\}}$ with K-means.
  Current iteration $n = 0$, initial prototypes $P^0 = P^c$.
  While not converge do:
    If n = 0 or (update_cavs and n % cav_update_frequency = 0) then:
      Learn Mapping module:
        Forward pass $x \in C \cup C'$ from Teacher and Student to get their concept activations $f_t$ and $f_s$.
        Learn the mapping module as autoencoders $E_M$ and $D_M$.
      CAV learning in mapped teacher's space:
        $\mathbf{v}^l_C$ learned by binary linear classifier as normal to decision boundary of $E_M(f_t(x))$ for $x \in C$ vs $E_M(f_t(x'))$ for $x' \in C'$.
    If $n$ % proto_update_frequency $= 0$ and $n \neq 0$ then:
      Estimate new class prototypes $P^c_{k \in \{0,K\}}$ with K-means.
      Weighted Proto-type $P^{n+1}_k = (1 - \alpha)P^n_k + \alpha P^c_k$.
    Else:
      $P^{n+1}_k = P^n_k$
    Train student with loss $L_C + L_o$.
    $n+ = 1$.
```

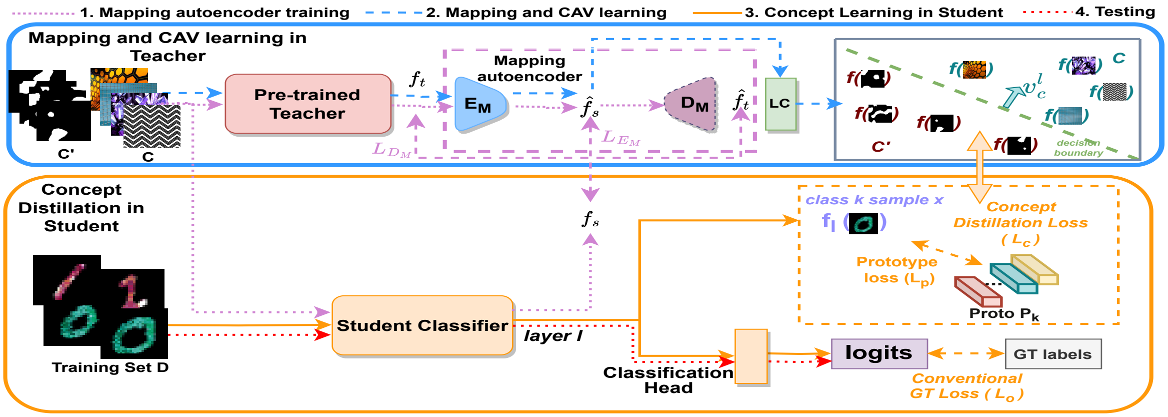


Figure 4.4: Our framework comprises a concept teacher and a student classifier and has the following four steps: 1) Mapping teacher space to student space for concepts C and C' by training an autoencoder E_M and D_M (dotted purple lines); 2) CAV ($\boldsymbol{v_c^l}$) learning in mapped teacher space via a linear classifier LC (dashed blue lines); 3) Training the student model with Concept Distillation (solid orange lines): We use $\boldsymbol{v_c^l}$ and class prototypes loss L_p to define our concept distillation loss L_c and use it with the original training loss L_o to (de)sensitize the model for concept C; 4) Testing where the trained model is applied (dotted red lines)

4.3 Experiments

We demonstrate the impact of the concept distillation method on the debiasing of classification problems as well as improving the results of a real-world reconstruction problem. Classification experiments cover two categories of Fig. 2.5: the zero-shot scenario, where no unbiased data is seen, and the few-shot scenario, which sees a few unbiased data samples. We use pretrained DINO ViT-B8 transformer [30] as the teacher. We use the last convolution layer of the student model for concept distillation. Our framework is computationally light. The mapping module ($< 120K$ parameters, < 10MB), CAV estimations (simple logistic regression, < 1MB), and prototype calculations all complete within a few iterations of training, taking 15-30 secs on a single 12GB Nvidia 1080 Ti GPU. Where relevant, we report average accuracy over five training runs with random seeds (± indicates variance).

CAV learning: For CAV learning, we use a logistic regression implemented by a single perceptron layer with sigmoid activation. We train it to distinguish between model activations of concept set (C) and its negative counterpart (C') in layer l using a cross-entropy loss for binary classification. This is theoretically the same but implementation-wise slightly different from [80], who also use a logistic regression but from sklearn [135]. We get the same results from either of them though our perceptron based implementation is slightly faster in terms of computation.

Mapping Module details: For the mapping module, we choose a pair of one down-convolutional and up-convolutional layers as encoder and decoder (depending on students and teacher's dimensions, it is determined whether the encoder is up or down-convolutional and vice versa). In our experiments, we train the autoencoder for a maximum of five epochs and select the encoder from the best of the first five epochs as our activation space mapping module M. We also tried with other deeper autoencoder architectures in our initial experiments but found the above simple one to give good results while being computationally cheapest. Due to the simple architecture (logistic regression or single up-down convolutions), both our CAV learning and Mapping module training are very lightweight ($<$ 120K parameters) and train within a minute for 10-15 concept sets having a number of images between 50-200 on a single 12GB Nvidia 1080 Ti GPU. We store the CAVs which are just the weights (coefficients) of the logistic regression model having negligible storage cost ($<$ 1Mb) for each CAV.

Proto-type calculation cost is dependent on the number of classes. In our experiments, we pass 200 samples of each class in a single forward pass through the smaller student model and only store the cluster means for each class as prototypes ($<$100Mb overall for the experiments in Table 1,2).

We experiment over the number of images for mnist tasks and find any number of images between 100-150 to work best while we fixate the number of images as 50 for BFFHQ.

Table 4.1: Comparison of the accuracy of our method with other zero-shot interpretable model-improvement methods. All methods require user-level intervention: Our method requires concept sets, while others (CDEP, RRR, EG) require user-provided rules.

Dataset	**Bias**	**Base**	**CDEP**[138]	**RRR**[139]	**EG**[49]	**Ours w/o Teacher**	**Ours**	**Ours+L**
ColorMNIST	Digit color	0.1	31.0	0.1	10.0	26.97	41.83	$\mathbf{50.93}_{\pm 1.42}$
DecoyMNIST	Spatial patches	52.84	97.2	**99.0**	97.8	87.49	98.58	$\mathbf{98.98}_{\pm 0.20}$
TextureMNIST	Digit textures	11.23	10.18	11.35	10.43	38.72	48.82	$\mathbf{56.57}_{\pm 0.79}$

4.3.1 Concept Sensitive Debiasings

We show results on two standard biased datasets (ColorMNIST [105] and DecoyMNIST [49]) and introduce a more challenging TextureMNIST dataset for quantitative evaluations. We also experimented on a real-world gender classification dataset BFFHQ [81] that is biased based on age. We compare with other state-of-the-art interpretable model improvement methods [9, 49, 98, 138, 139, 176]. Fig. 4.3 summarizes the datasets, their biases, and the concept sets used.

Poisoned MNIST Datasets: ColorMNIST [105] has MNIST digit classes mapped to a particular color in the training set [138]. The colors are reversed in the test set. The baseline CNN model (*Base*) trained on ColorMNIST gets 0% accuracy on the 100% poisoned test set, indicating that the model learned

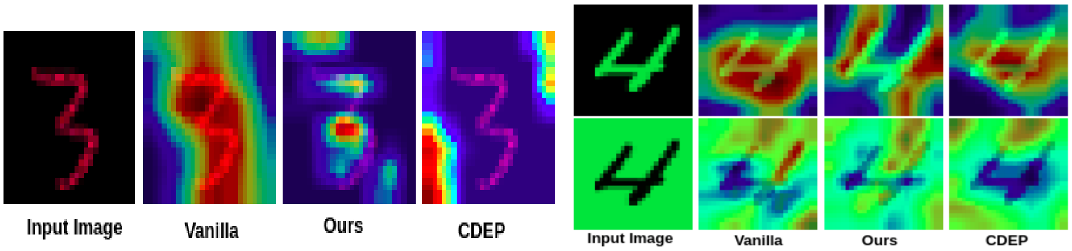


Figure 4.5: GradCAM [157] visualizations (more red, more important) results. Left: Our method desensitizes the model to pixel color (less red on foreground digit). Right: For the extreme case when the color bias is in the background instead of foreground (initial training), our model focuses on the shape more than CDEP (which is more blue in the foreground).

color shortcuts instead of digit shapes. We debias the model using a concept loss for color using color patches *vs*.random shapes (negative concept) to estimate CAV for color (Fig. 4.3). We show results comparison on the usage of teacher and prototypes component of our method in Tab. 4.2. As can be seen from Tab. 4.2, our method of using intermediate layer sensitivity via prototypes as described in subsection 4.2.2 yields better results. Similarly, usage of teacher (described in subsection 4.2.3) facilitates better student concept learning (also shown in CAV comparisons in supplementary). Our concept sensitive training not only improves student accuracy but observes evident reduction in TCAV scores [80] of bias concept as seen from Tab. 4.3.

Tab. 4.4 compares our results with the best zero-shot interpretability based methods, *i.e*., CDEP [138], RRR [139], and EG [49]. Our method improves the accuracy from 31% by CDEP to 41.83%, and further to 50.93% additionally with the local explanation loss from [139].

Table 4.2: Main components of our method shown on ColorMNIST dataset: Ours usage of teacher and proto-types yields the best performance.

Teacher?	**Prototype?**	**Accuracy**
✗	✗	9.96
✗	✓	26.97
✓	✗	30.94
✓	✓	**50.93**

GradCAM [157] visualizations in Fig. 4.5 show that our trained model focuses on highly relevant regions. Fig. 4.5 left compares our class-wise accuracy on 20% biased ColorMNIST with ClArC [9], a few-shot global method that uses CAVs for artifact removal. (ClArC results are directly reported from their main work due to the unavailability of the public codebase.) ClArC learns separate CAVs for each class by separating the biased color digit images (*e.g*.red zeros) from the unbiased images (*e.g*.zeros

Table 4.3: TCAV scores of bias concept: Our concept sensitive training significantly decreases the sensitivity of model towards bias.

Dataset	**Concept**	**Base Model**	**Ours**
ColorMNIST	Color	0.52	**0.21**
DecoyMNIST	Spatial patches	0.57	**0.45**
TextureMNIST	Textures	0.68	**0.43**
BFFHQ	Age	0.78	**0.13**

Table 4.4: Comparison of the accuracy of our method with other zero-shot interpretable model-improvement methods. All methods require user-level intervention: Our method requires concept sets, while others (CDEP, RRR, EG) require user-provided rules.

Dataset	**Bias**	**Base**	**CDEP**[138]	**RRR**[139]	**EG**[49]	**Ours w/o Teacher**	**Ours**	**Ours+L**
ColorMNIST	Digit color	0.1	31.0	0.1	10.0	26.97	41.83	$\mathbf{50.93}_{\pm 1.42}$
DecoyMNIST	Spatial patches	52.84	97.2	**99.0**	97.8	87.49	98.58	$\mathbf{98.98}_{\pm 0.20}$
TextureMNIST	Digit textures	11.23	10.18	11.35	10.43	38.72	48.82	$\mathbf{56.57}_{\pm 0.79}$

in all colors). A feature space linear transformation is then applied to move input sample activations away/towards the learned CAV direction. Their class-specific CAVs definition can not be generalized to test sets with multiple classes. This results in a higher test accuracy variance as seen in Fig. 4.5. Further, as they learn CAVs in the biased model's activation space, a common concept set cannot be used across classes.

DecoyMNIST [49] has class indicative gray patches on image boundary that biased models learn instead of the shape. We define concept sets as gray patches *vs*.random set (Fig. 4.3) and report them in Tab. 4.4 second row. All methods perform comparably on this task as the introduced bias does not directly corrupt the class intrinsic attributes (shape or color), making it easy to debias. **TextureMNIST** is a more challenging dataset that we have created for further research in the area. Being an amalgam of colors and patterns, textures are more challenging as a biasing attribute. Our method improves the performance while others struggle on this task (Tab. 4.4 last row). Fig. 4.5 shows that our method can focus on the right aspects for this task.

Generalization to multiple biasing situations is another important aspect. Tab. 4.5 shows the performance on different types of biases by differently debiased models. Our method performs better than CDEP in all situations. Interestingly, our texture-debiased model performs better on color biases than CDEP debiased on the color bias! We believe this is because textures inherently contain color infor-

Table 4.5: Our method improves the model using human-centered concepts and shows better generalization to different datasets, while CDEP, which uses pixel-wise color rules, cannot.

	ColorMNIST Trained			TextureMNIST Trained		
Test Dataset	**Base**	**CDEP**[138]	**Ours+L**	**Base**	**CDEP**[138]	**Ours+L**
Invert color	0.00	23.38	**50.93**	11.35	10.18	**45.36**
Random color	16.63	37.40	**46.62**	11.35	10.18	**64.96**
Random texture	15.76	28.66	**32.30**	11.35	10.18	**56.57**
Pixel-hard	15.87	33.11	**38.88**	11.35	10.18	**61.29**

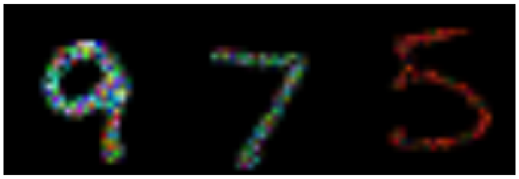


Figure 4.6: pixel hard MNIST test set

mation, which our method can leverage efficiently. Our method effectively addresses an extreme case of color bias introduced in the background of ColorMNIST, a dataset originally biased in foreground color. By introducing the same color bias to the background and evaluating our ColorMNIST debiased models, we test the capability of our approach. In an ideal scenario where color debiasing is accurate, the model should prioritize shape over color, a result achieved by our method but not by CDEP as shown in Fig. 4.5.

BFFHQ [81] dataset is used for the gender classification problem. It consists of images of young women and old men. The model learns entangled age attributes along with gender and gets wrong predictions on the reversed test set *i.e.*old women and young men. We use the bias conflicting samples by Lee et al. [98] specifically, old women *vs.*women and young men *vs.*men as class-specific concept sets.We compare against recent debiasing methods EnD [176] and DFA [98]. Tab. 4.6 shows our method getting a comparable accuracy of 63%. We also tried other concept set combinations: *(i)* old *vs.*young

Table 4.6: Comparisons in few-shot setting: Our method is not limited to user-provided concept sets and can also work with bias-conflicting samples. We compare our accuracy over the BFFHQ dataset [81] with other few-shot debiasing methods.

Dataset	**Bias**	**Base**	**EnD** [176]	**DFA** [98]	**Ours w/o Teacher**	**Ours**
BFFHQ	Age	$56.87_{\pm 2.69}$	$56.87_{\pm 1.42}$	$61.27_{\pm 3.26}$	59.4	$\mathbf{63_{\pm 0.79}}$

Table 4.7: Increasing the bias of the teacher on ColorMNIST reduces accuracy.

Teacher's Training Data Bias%	No Distil	5	10	25	75	90	100
Student Accuracy%	26.97	40.47	33.18	30.38	28.74	23.23	23.63

(where both young and old concepts should not affect) → 62.8% accuracy, *(ii)* old *vs*.mix (men and women of all ages) and young *vs*.mix → 62.8% accuracy, *(iii)* old *vs*.random set (consisting of random internet images from [80]) and young *vs*.random → 63% accuracy. These experiments indicate the stability of our method to the concept set definitions. This proves our method can also work with bias-conflicting samples [98] and does not necessarily require concept sets (Tab. 4.6). We also experimented by training class-wise (for all women removing young bias followed by men removing old bias and vice versa) vs training for all classes together (both men, women removing age bias as described above) and observed similar results, suggesting that our concept sensitive training is robust to class-wise or all class agnostic training. Local interpretable improvement methods like CDEP, RRR, and EG are not reported here as they cannot capture complex concepts like age due to their pixel-wise loss or rule-based nature.

Discussion ***No Distillation Case:*** We additionally show our method without teacher (Our Concept loss with CAVs learned directly in student) in all experiments as ”Ours w/o Teacher” and find inferior performance when compared to Our method with Distillation.

Bias in Teacher: We check the effect of bias in teacher by training it with varying fractions of OOD samples to bias them. Specifically, we use the same student architecture for the teacher. The teacher is trained on the ColorMNIST dataset with 5, 10, 25, 50, 75, and 90% biased color samples in the trainset (*e.g.*k% bias indicates k% red zeros). The resulting concept-distilled system is then tested on the standard 100% reverse color setting of ColorMNIST. Tab. 4.7 shows that concept distillation improves performance even with high teacher bias, though accuracy decreases with the increasing bias in teacher. Apparently, 100% bias in teacher in this setting of teacher with same architecture as the student is essentially CAV learning in same model case (No Distil). Here, the improvements are due to prototypes, and as can be seen, there is a slight degradation in performance of *100% bias* vs *No Distil* (23.63 vs 26.97). This can be attributed to an error due to the mapping module.

Mapping Module: Case of Ideal Mapping In theory, if the mapping module has a zero-loss, it could make the distillation case the same as the case without distillation, but this is not observed in our experiments due to two main reasons: *(i)* We use the mapping module to map *only* the conceptual knowledge in CAV and train it only for concept sets and not the training samples. *(ii)* Due to major differences in the perceived notion of concepts in teacher and student networks and due to a simple mapping autoencoder (one upconv and downconv layer) the MSE loss never goes to zero (e.g, in ColorMNIST, it starts from 11 and converges at 5 for DINO teacher to biased student alignment). In our initial experiments, we tried bigger architectures (ResNet18+) and found improved mapping losses but decreased student performances. Mapping Encoder encodes an expert's knowledge into the system via the provided concept sets, quantifies this knowledge as CAV via a generalized teacher model trained on large amount of data, and thus helps in inducing it via distillation into the student model. This brings threefold advantages in our system: expert's intuition, large model's generality, and efficiency of distillation.

An experiment: CAV Learning comparison In ColorMNIST, zeros are always associated with the color *red*. We learn a CAV for the concept of *red* (CAV_{red} separating red patches with other colored patches) in each teacher, mapped teacher and base model (biased initial Student) and measure its cosine similarity (cs) with the respective model representations for concept images of red, red-zeros.

As seen from Tab. 4.8, the Teacher and Mapped Teacher have cs(red) ¿ cs(red_zeros) ¡ cs(red_non_zeros) which indicate correct concept learning while the Student (Base Model) has cs(red) ¡ cs(red_zeros) ¡ cs(red_non-zeros) that indicates confusion of bias with concept (concept red-zeros confused with concept red). Thus, teacher's CAV and its mapped version capture the intended concept well, while the base model (student) confuses the red-zeros concept with CAV. This small demonstration shows (a) why the teacher is needed for good CAV learning and (b) how well CAV's are transferred to the student via the mapping module. Mapping module does not bias the CAV representation.

Table 4.8: Cosine Similarity Order of concepts with CAV_{red}

Model	'red'	'red_zeros'	'red_non_zeros'	'non_red_zeros'
Teacher	**0.084**	**-0.013**	-0.009	0.005
Mapped Teacher	0.287	-0.044	0.014	0.024
Base Model	-0.023	-0.019	-0.013	0.000

Fixing vs Varying CAVs in Experiments In our experimentation reported in main work, we fix CAVs but we experimented with updating CAVs in the student after every few training iterations (50, 100, 200, etc.). Specifically, we experimented in the following settings and report the concept loss L_c curves with iterations in Fig. 4.7.

- cav_update_iter200: CAV updates every 200 iterations.
- cav_update_iter100: CAV updates every 100 iterations.
- constant_cav: Fixed CAV throughout training.

As seen in Fig. 4.7, there is a recurring pattern with concept loss L_C increasing on CAV update (due to an abrupt change in objectives), followed by a subsequent decrease due to optimization. However, L_C remains lower than the initial value the model started with (from 105.98 at iteration 0, loss dropped to 9.51 at iteration 199 (before the CAV update) but jumps to 20.61 following the CAV update), underscoring the efficacy of our approach. Similar patterns were consistently observed when updating CAVs in varying numbers of iterations. This trend persisted across diverse datasets during training as well.

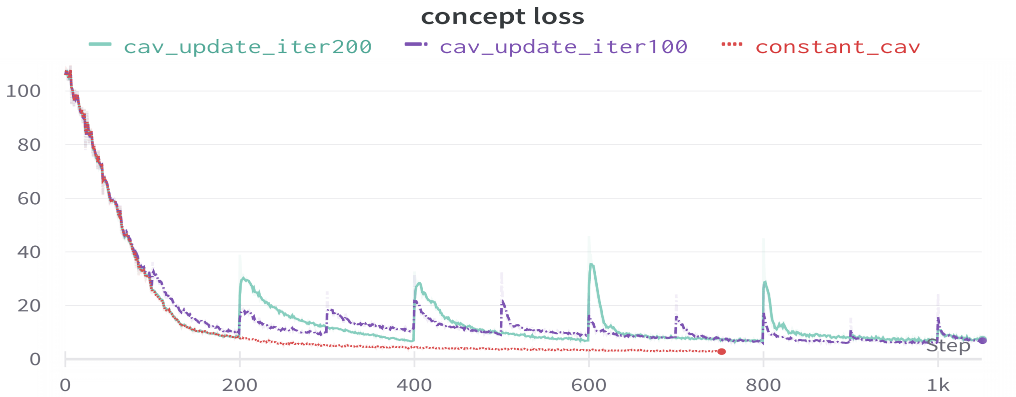


Figure 4.7: Loss Curves in varying cav update iterations: frequent pattern observed, convergence quickly

The given graph is shown for experimentation over ColorMNIST, but we find the same recurring patterns across all other datasets. Additionally, the best validation accuracy (and also corresponding test accuracy) values for all the settings mentioned above (whether fixed or varying CAVs) are the same (within $< 0.3\%$ accuracy changes amongst the settings). The graph also shows that our design choice of keeping CAVs fixed is good practically as the loss quickly converges to a lower value.

Table 4.9: Impact of different variants of CAV sensitivity calculation gradient ∇X in proposed loss Eq. 4.1 on the final results

X	logit	L_o	L_p, fixed proto	L_p, varying prototypes (Ours)
Accuracy %	25.55	30.94	40.02	41.83

Ablations We evaluate the impact of different components of our framework apart from the ones already mentioned in the work.

We experimented with different ways of calculating sensitivity used for L_c in Eq. 4.1 by replacing ∇L_o with ∇X where X is taken as *(i)* last layer outputs or logits; *(ii)* last layer loss L_o and *(iii)* intermediate layer prototype based loss L_p in two settings: fixed prototypes where prototypes are kept fixed as initial ($\alpha = 0$) vs prototypes are varied with α. Settings *(i)* and *(ii)* are essentially final layer sensitivity calculation according to the original implementation by Kim et al. [80] while Setting *(iii)* is our proposed intermediate layer sensitivity using prototypes. We also show results when *(a)* KNN k in the prototype calculation is varied and found k=7 to work best Fig. 4.8 *(b)* number of images (#imgs) in the concept set are varied and we observe a peak in #imgs = 150 which is chosen as the KNN k and #imgs in our experiments.

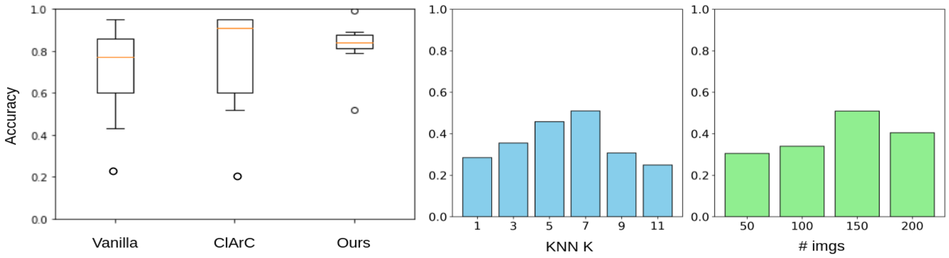


Figure 4.8: Left: Comparison of our method and ClArC [9] on 20% biased ColorMNIST. Right: Impact of varying the concept set size and number of means on 100% biased ColorMNIST.

Design Choices and Implementation Details **Choosing the Student Layer for Concept Distillation:** CAVs can be calculated for any model layer. Which student layers should be used? We show results only for the last convolution layer of the student model in the work, but theoretically, our framework can be extended to any number of layers at any depth. Our design choice is based on the fact that the deeper layers of the model encode complex higher-order class-level features while the shallower layers encode low-level features. The last convolution layer represents more abstract features, which are easily representable for humans in the form of concepts rather than low-level features in other layers. For the same reasons, [4] too use the last convolution layers for conceptual explanation generation.

Teacher Selection: For a teacher, we experimented with various model architectures (including the biased teacher ones shown in Sec. 4 Ablations) and chose a pre-trained DINO transformer [30] for the main reason of scalability. DINO has been proven to work well for a variety of tasks [181, 188]. A large model knows a variety of concepts and can be used as a teacher for various tasks, as shown in our experiments. We use the same DINO teacher on very different classification datasets like biased MNIST (ColorMNIST, DecoyMNIST, and TextureMNIST) and BFFHQ, as well as over a completely different problem of IID.

Among the DINO variants, we found ViT-B/8 (85M parameters) to perform the best, aiding student to get 50.93% accuracy on ColorMNIST while ViT-S/8 (21M parameters) aced 39% student accuracy. We thus picked DINO ViT-B/8 for all our experiments. We used the code implementation of the DINO feature extractor by Tschernezki et al. [182] and loaded the checkpoints for DINO variants from [30]. The DINO ViT-B/8 gives 768-dimensional feature images, further reduced to 64 using PCA.

Apart from biased small teacher experiments in Tab. 6 (as shown in the work) where we vary the bias in teacher from 5% to 100%, we also experimented when the small teacher network is (pre)trained on 0% bias (simply MNIST dataset [97]). We found the student to achieve an accuracy of 23.6% with this teacher network. This accuracy is lesser than that in biased teacher due to the fact that the teacher trained on the MNIST dataset having grayscale images has never seen the concept of color. Henceforth as mentioned before, it is important for teacher to have the knowledge of concepts for our proposed concept distillation to work well.

Taking only Robust CAVs for Distillation [80] use t-testing with concept vs multiple random samples to filter robust CAVs for TCAV score estimation. Similar to them, we employed t-testing initially by taking concept vs multiple random samples and selecting only the significant CAVs. This proved to be too expensive computationally during training, especially during frequent CAV updates. Currently, we have a simple filter on CAV classification accuracy > 0.7 to select only the good CAVs (i.e., CAVs that can differentiate concept vs random). The concept loss corresponding to all such valid CAVs is then averaged before backpropagating. This design simplification was empirically verified and found to work equivalently to [80] t-testing).

Student Architecture details: For the student network in classification debiasing applications, we use two convolution layers followed by two fully connected layers as done by [138] [98] for all the three biased MNIST experiments . We use Resnet18 (no pre-training) for BFFHQ student architecture as done by [98] and apply our method over the ”layer4.1.conv1” layer. We use Adam [82] optimizer with a learning rate of 10e-4, $0.9 \leq \beta \leq 0.999$, and $ep = 1e-08$ with a weight decay of 0 (all default pytorch values except learning late). We use a batch size of 32 for MNIST experiments and 64 for BFFHQ experiments. For the mapping module, we use one up-convolution and one down-convolution layer for encoders and decoders. We train the autoencoders with an L2 loss. For training the MNIST and BFFHQ models, we use the cross-entropy loss as the Ground Truth loss (L_o). Our concept loss L_C is weighted by a parameter λ varied from 0.01 to 10e5 in our experiments. We found it to work best for values close to 20 in our experimentation. Other parameter values which we found to work best are number of clusters in K-Means $k = 7$ and proto-types updation weight $\alpha = 0.3$. Our student model converges within 2-3 epochs of training with a training time of less than 1.5-2 hours for MNIST datasets and within 4 hours for BFFHQ dataset (on one Nvidia 1080 Ti GPU).

For MNIST datasets (ColorMNIST and DecoyMNIST), we use the splits by Rieger et al. [138]. For the creation of TextureMNIST, we use the above-obtained splits of digits and replace the colors with textures from DTD [39] (all colored removed and rather texture bias added). We use random flat-colored patches as the concept ”color,” random textures as the ”textures” concept, and gray-colored patches as the ”gray” concept set. For the negative concept set, we create randomly shaped white blobs in black backgrounds. ColorMNIST and TextureMNIST datasets have test sets comprising flipped colors and random textures digits. To check the generalization of our model, we tested the ColorMNIST and TextureMNIST trained models on other test sets having random colors and textures (Tab. 2) . Also, we created our Pixel Hard MNIST test set (Fig. 4.6), which has color in each pixel in digits randomized in contrast to one color in the entire digit in others(ColorMNIST and TextureMNIST). .

For BFFHQ, we take the 48 images each for young men and old women (bias conflicting samples as given by Lee et al. [98]) as our concept sets for class-specific training (separate concepts for each class). While for the negative concept set of class-specific training, we take mixed images of both young and old women for women concept and young and old men for men concept. We use the same train-test-validation split as done by [98].

4.3.2 Prior Knowledge Induction

As we discussed earlier, good ground truth for Intrinsic Image Decomposition is hard to create and IID algorithms are evaluated on synthetic data or using some sparse manual annotations [19, 86]. In chapter 2 we proposed Concept Sensitivity Metric to measure the disentanglement of R and S. Using our concept distillation framework, we extended the proposed post-hoc quality evaluation method to ante-hoc training of the IID network to increase disentanglement between R and S. We observed improved CSM scores along with MSE, SSIM metrics and qualitative results. For details please refer to Second Author Thesis.

Concept Distillation *vs*.Knowledge Distillation Concept Distillation and Knowledge Distillation are two distinct approaches in the realm of machine learning, particularly in the context of transferring information from a complex model (teacher) to a simpler one (student). Here's a clearer comparison between the two:

- **Concept Distillation:**
  - *Concept Activation Vectors (CAVs):* Concept Distillation leverages CAVs from the teacher model. These vectors represent high-level concepts learned by the teacher.
  - *Loss Function:* It employs a specialized loss function that aligns the student's concept activations with the desired direction, as indicated by the teacher's CAVs.
  - *Cosine Similarity Tuning:* This method controls the influence of specific concepts on the student model by adjusting the cosine similarity between the teacher's and student's concept activations.
  - *Weighted Loss with Ground Truth (GT):* Concept Distillation also incorporates a weighted loss function that combines the concept alignment loss with the traditional loss against ground truth. This helps balance the influence of the teacher's concepts with the actual task performance.
- **Knowledge Distillation:**
  - *Soft Labels from Teacher:* In Knowledge Distillation, the student model learns from the soft labels (or logits) generated by the teacher model. These soft labels contain richer information compared to hard labels, as they reflect the teacher's confidence across all possible classes.
  - *Loss Function:* The student model's loss function aims to minimize the difference between its predicted logits and the teacher's logits, effectively making the student mimic the teacher's output distribution.
  - *Temperature Parameter:* Knowledge Distillation uses a temperature parameter in the softmax function to control the softness of the teacher's output distribution. A higher temperature leads to softer probabilities, which can be more informative for the student model.

In summary, while both Concept Distillation and Knowledge Distillation aim to transfer knowledge from a teacher to a student model, they differ in their approaches. Concept Distillation focuses on aligning high-level concept representations between the teacher and student, whereas Knowledge Distillation is about making the student mimic the teacher's output distribution, particularly the confidence levels across different classes.

Discussion and Limitations: Our concept distillation framework can work on different classification and reconstruction problems, as we demonstrated. Our method can work well in both zero-shot (with concept sets) and few-shot (with bias-conflicting samples) scenarios. Bias-conflicting samples may not be easy to obtain for many real-world applications. Our required user-provided concept samples can incur annotation costs, though concept samples are usually easier to obtain than bias-conflicting samples. When neither bias-conflicting samples nor user-provided concept sets are available, concept discovery methods like ACE [61] could be used. ACE discovers concepts used by the model by image super-pixel clustering. Automatic bias detection methods like Bahadori and Heckerman [12] can be used to discover or synthesize bias-conflicting samples for our method. Our method can also be used to induce prior knowledge into complex reconstruction/generation problems, as we demonstrated with IID. The dependence on the teacher for conceptual knowledge could be another drawback of our method, as with all distillation frameworks [71].

4.4 Conclusion

We presented a concept distillation framework that can leverage human-centered explanations and the conceptual knowledge of a pre-trained teacher to distill explanations into a student model. Our method can desensitize ML models to selected concepts by perturbing the activations away from the CAV direction without modifying its underlying architecture. We presented results on multiple classification problems. We also showed how prior knowledge can be induced into the real-world IID problem. In future, we would like to extend our work to exploit automatic bias detection and concept-set definition. Our approach also has potential to be applied to domain generalization and multitask learning problems.

Chapter 5

Summary and Future Work

In this thesis, we study various XAI-based approaches, specifically human concept-based approaches for model improvement and evaluation. We build a Concept Distillation method that leverages global explanations and the power of pre-trained large networks to distill explanations to a model. We use concept activation vectors, CAVs for concept representations. Our method can sensitize or desensitize the model towards the user given concepts by perturbing the activation of class samples towards or away from the direction of CAV. We learn our concepts in a large teacher model and distill this knowledge in the classifier student via our novel concept distillation loss. We present results on multiple datasets for classification problems and also show how our method can be used to induce prior knowledge in difficult real-world reconstruction problems. In the future, we would like to extend our Concept Distillation work aiming at automatic bias detection and automatic concept-set extraction via methods like ACE [61]. We would also like to test our idea in the domain generalization, Distributionally Robust Optimization [142], and multitask setting. Currently, we map teacher to student via a mapping module (keeping teacher fixed), which needs a mapping module to be re-learned. We would like to work towards a universal teacher module (no need for retraining of mapping module), which can be used for any student.

We also build a Concept Sensitivity Metric to evaluate the quality of IID based on its definition. The CSM_R and CSM_S metrics evaluate the disentanglement of the recovered reflectance and shading. These metrics overcome several shortcomings of the current IID evaluation strategies. They are consistent over real-world and synthetic scenes and have lesser dependence on the evaluation set as we use the model's sensitivity towards concepts rather than direct pixel-to-pixel comparison with ground truth annotations.

Since these metrics measure the quality of the output and can provide additional terms to the loss being minimized to improve the IID calculations, like in a fine-tuning step, we intend to work on this in the future. The use of metrics defined for interpretability in a loop to improve the performance of the original problem has a wide scope of applicability. The approach underlying Concept Sensitivity Metric has wider potential application beyond the IID problem. Choosing appropriate concepts and their activations, CSM can be used to evaluate results of image harmonization, style transfer, image enhancement, *etc*.

Related Publications

- **Avani Gupta**, Saurabh Saini, P J Narayanan. *Interpreting Intrinsic Image Decomposition using Concept Activations*. Thirteenth Indian Conference on Computer Vision, Graphics and Image Processing (ICVGIP '22). (Oral and Best Paper Award)
- **Avani Gupta**, Saurabh Saini, P J Narayanan. *Concept Distillation: Leveraging Human-Centered Explanations for Model Improvement* (Neurips '23)
- **Avani Gupta**, P J Narayanan. *A survey on Concept-based Approaches For Model Improvement(arxiv)*

Bibliography

[1] C Aditya, S Anirban, D Abhishek, and H Prantik. Grad-cam++: improved visual explanations for deep convolutional networks. arxiv. *arXiv preprint arXiv:1710.11063*, 2018. 2, 3

[2] AI Guys. Knowledge distillation in neural networks. https://medium.com/aiguys/knowledge-distillation-in-neural-networks-37f416ede203, June 2019. Accessed on April 18, 2023. x, 12

[3] Naveed Akhtar. A survey of explainable ai in deep visual modeling: Methods and metrics. *ArXiv*, abs/2301.13445, 2023. 15

[4] Arjun Akula, Shuai Wang, and Song-Chun Zhu. Cocox: Generating conceptual and counterfactual explanations via fault-lines. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 2594–2601, 2020. 18, 19, 51, 54, 64

[5] Kamran Alipour, Aditya Lahiri, Ehsan Adeli, Babak Salimi, and Michael J. Pazzani. Explaining image classifiers using contrastive counterfactuals in generative latent spaces. *ArXiv*, abs/2206.05257, 2022. 18

[6] Neil Alldrin, Todd Zickler, and David Kriegman. Photometric stereo with non-parametric and spatially-varying reflectance. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008. xii, 36, 40, 41

[7] David Alvarez Melis and Tommi Jaakkola. Towards robust interpretability with self-explaining neural networks. *Adv. Neural Inform. Process. Syst.*, 2018. xi, 20, 21, 24, 50

[8] Dario Amodei, Sundaram Ananthanarayanan, Rishita Anubhai, Jingliang Bai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Qiang Cheng, Guoliang Chen, et al. Deep speech 2: End-to-end speech recognition in english and mandarin. In *International conference on machine learning*, pages 173–182. PMLR, 2016. 1

[9] Christopher J Anders, Leander Weber, David Neumann, Wojciech Samek, Klaus-Robert Müller, and Sebastian Lapuschkin. Finding and removing clever hans: Using explanation methods to debug and improve deep models. *Information Fusion*, 77:261–295, 2022. xiii, 20, 21, 26, 29, 50, 52, 57, 58, 64

[10] Maximilian Augustin, Valentyn Boreiko, Francesco Croce, and Matthias Hein. Diffusion visual counterfactual explanations. *arXiv preprint arXiv:2210.11841*, 2022. 18

[11] 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):e0130140, 2015. 29

[12] Mohammad Taha Bahadori and David E Heckerman. Debiasing concept-based explanations with causal analysis. *arXiv preprint arXiv:2007.11500*, 2020. 20, 21, 26, 67

[13] Andrew Bai, Chih-Kuan Yeh, Pradeep Ravikumar, Neil YC Lin, and Cho-Jui Hsieh. Concept gradient: Concept-based interpretation without linear assumption. *arXiv preprint arXiv:2208.14966*, 2022. 3, 17, 19

[14] Jonathan T Barron and Jitendra Malik. Intrinsic scene properties from a single rgb-d image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 17–24, 2013. 30

[15] Anil S Baslamisli, Yang Liu, Sezer Karaoglu, and Theo Gevers. Physics-based shading reconstruction for intrinsic image decomposition. *Computer Vision and Image Understanding*, 205: 103183, 2021. 30

[16] David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. Network dissection: Quantifying interpretability of deep visual representations. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3319–3327, 2017. 2, 4, 16, 17

[17] Katharina Beckh, Sebastian Müller, Matthias Jakobs, Vanessa Toborek, Hanxiao Tan, Raphael Fischer, Pascal Welke, Sebastian Houben, and Laura von Rueden. Explainable machine learning with prior knowledge: An overview. *arXiv preprint arXiv:2105.10172*, 2021. 11

[18] Katharina Beckh, Sebastian Müller, Matthias Jakobs, Vanessa Toborek, Hanxiao Tan, Raphael Fischer, Pascal Welke, Sebastian Houben, and Laura von Rueden. Sok: Harnessing prior knowledge for explainable machine learning: An overview. In *First IEEE Conference on Secure and Trustworthy Machine Learning*, 2022. 11, 20

[19] Sean Bell, Kavita Bala, and Noah Snavely. Intrinsic images in the wild. *ACM Transactions on Graphics (TOG)*, 33:1–12, 2014. 66

[20] Sean Bell, Kavita Bala, and Noah Snavely. Intrinsic images in the wild. *ACM Transactions on Graphics (TOG)*, 33(4):1–12, 2014. 30, 31, 32, 33

[21] Sunay Bhat, Jeffrey Jiang, Omead Pooladzandi, and Gregory Pottie. De-biasing generative models using counterfactual methods. *arXiv preprint arXiv:2207.01575*, 2022. 6

[22] Alexander Binder, Grégoire Montavon, Sebastian Lapuschkin, Klaus-Robert Müller, and Wojciech Samek. Layer-wise relevance propagation for neural networks with local renormalization layers. In *Artificial Neural Networks and Machine Learning–ICANN 2016: 25th International Conference on Artificial Neural Networks, Barcelona, Spain, September 6-9, 2016, Proceedings, Part II 25*, pages 63–71. Springer, 2016. 20

[23] Nicolas Bonneel, Balazs Kovacs, Sylvain Paris, and Kavita Bala. Intrinsic Decompositions for Image Editing. *Computer Graphics Forum (Eurographics State of The Art Report)*, 2017. xiv, 30, 32, 33, 36, 46

[24] Nicolas Bonneel, Balazs Kovacs, Sylvain Paris, and Kavita Bala. Intrinsic decompositions for image editing. In *Computer Graphics Forum*, volume 36, pages 593–609. Wiley Online Library, 2017. 31, 32

[25] Andrea Bontempelli, Fausto Giunchiglia, Andrea Passerini, and Stefano Teso. Toward a unified framework for debugging concept-based models. 2021. 20, 21, 27

[26] Andrea Bontempelli, Stefano Teso, Fausto Giunchiglia, and Andrea Passerini. Concept-level debugging of part-prototype networks. 2023. 3, 17, 20, 21, 27

[27] Daniel J Butler, Jonas Wulff, Garrett B Stanley, and Michael J Black. A naturalistic open source movie for optical flow evaluation. In *European conference on computer vision*, pages 611–625. Springer, 2012. 30, 32, 33

[28] Vladimir Bychkovsky, Sylvain Paris, Eric Chan, and Frédo Durand. Learning photographic global tonal adjustment with a database of input / output image pairs. In *The Twenty-Fourth IEEE Conference on Computer Vision and Pattern Recognition*, 2011. 35

[29] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European conference on computer vision (ECCV)*, pages 132–149, 2018. 51, 54

[30] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021. 54, 56, 64

[31] Daniel T. Chang. Concept-oriented deep learning. *ArXiv*, abs/1806.01756, 2018. 8, 11

[32] Daniel T Chang. Concept-oriented deep learning: Generative concept representations. *arXiv preprint arXiv:1811.06622*, 2018. 3, 50

[33] Martin Charachon, Paul-Henry Cournède, Céline Hudelot, and Roberto Ardon. Leveraging conditional generative models in a general explanation framework of classifier decisions. *Future Generation Computer Systems*, 132:223–238, 2022. 18

[34] Kushal Chauhan, Rishabh Tiwari, Jana von Freyberg, Pradeep Shenoy, and Krishnamurthy Dvijotham. Interactive concept bottleneck models. *ArXiv*, abs/2212.07430, 2022. 21, 27

[35] Chaofan Chen, Oscar Li, Daniel Tao, Alina Barnett, Cynthia Rudin, and Jonathan K Su. This looks like that: deep learning for interpretable image recognition. *Advances in neural information processing systems*, 32, 2019. 17

[36] Jiefeng Chen, Xi Wu, Vaibhav Rastogi, Yingyu Liang, and Somesh Jha. Robust attribution regularization. *Advances in Neural Information Processing Systems*, 32, 2019. 20

[37] Zhi Chen, Yijie Bei, and Cynthia Rudin. Concept whitening for interpretable image recognition. *Nature Machine Intelligence*, 2(12):772–782, 2020. 17, 19

[38] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, , and A. Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014. 37

[39] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3606–3613, 2014. 65

[40] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018. URL http://www.blender.org. 35

[41] Vargha Dadvar. Poem: pattern-oriented explanations of cnn models. Master's thesis, University of Waterloo, 2022. 17

[42] Partha Das, Sezer Karaoglu, and Theo Gevers. Pie-net: Photometric invariant edge guided network for intrinsic image decomposition. *ArXiv*, abs/2203.16670, 2022. 30, 32

[43] Fabio De Sio and Chantal Marazia. Clever hans and his effects: Karl krall and the origins of experimental parapsychology in germany. *Studies in History and Philosophy of Science Part C: Studies in History and Philosophy of Biological and Biomedical Sciences*, 48:94–102, 2014. 5

[44] Shi Dong, Ping Wang, and Khushnood Abbas. A survey on deep learning and its applications. *Computer Science Review*, 40:100379, 2021. 1

[45] Jean Donham. Deep learning through concept-based inquiry. *School Library Monthly*, 27(1): 8–15, 2010. 1

[46] Finale Doshi-Velez and Been Kim. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*, 2017. 2

[47] Mengnan Du, Ninghao Liu, and Xia Hu. Techniques for interpretable machine learning. *Communications of the ACM*, 63(1):68–77, 2019. 1, 15

[48] Sylvain Duchêne, Clement Riant, Gaurav Chaurasia, Jorge Lopez-Moreno, Pierre-Yves Laffont, Stefan Popov, Adrien Bousseau, and George Drettakis. Multi-view intrinsic images of outdoors scenes with an application to relighting. *ACM Transactions on Graphics*, page 16, 2015. 30, 32

[49] Gabriel Erion, Joseph D Janizek, Pascal Sturmfels, Scott M Lundberg, and Su-In Lee. Improving performance of deep learning models with axiomatic attribution priors and expected gradients. *Nature machine intelligence*, 3(7):620–631, 2021. 28, 29, 52, 57, 58, 59

[50] Qingnan Fan, Jiaolong Yang, Gang Hua, Baoquan Chen, and David Wipf. Revisiting deep intrinsic image decompositions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8944–8952, 2018. 30

[51] Ruth Fong and Andrea Vedaldi. Net2vec: Quantifying and explaining how concepts are encoded by filters in deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8730–8738, 2018. 2, 4, 16

[52] Ruth C Fong and Andrea Vedaldi. Interpretable explanations of black boxes by meaningful perturbation. In *Proceedings of the IEEE international conference on computer vision*, pages 3429–3437, 2017. 15, 16

[53] Felix Friedrich, Wolfgang Stammer, Patrick Schramowski, and Kristian Kersting. A typology to explore the mitigation of shortcut behavior. 2022. x, 9, 20, 25, 26

[54] Hiroshi Fukui, Tsubasa Hirakawa, Takayoshi Yamashita, and Hironobu Fujiyoshi. Attention branch network: Learning of attention mechanism for visual explanation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10705–10714, 2019. 20

[55] Yuyang Gao, Siyi Gu, Junji Jiang, Sungsoo Ray Hong, Dazhou Yu, and Liang Zhao. Going beyond xai: A systematic survey for explanation-guided learning. *arXiv preprint arXiv:2212.03954*, 2022. x, 1, 7, 8, 20

[56] Elena Garces, Adolfo Munoz, Jorge Lopez-Moreno, and Diego Gutierrez. Intrinsic images by clustering. In *Computer graphics forum*, volume 31, pages 1415–1424. Wiley Online Library, 2012. 30

[57] Srishti Gautam, Marina M-C Höhne, Stine Hansen, Robert Jenssen, and Michael Kampffmeyer. This looks more like that: Enhancing self-explaining models by prototypical relevance propagation. *Pattern Recognition*, 136:109172, 2023. 20

[58] Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020. x, 5, 6, 7

[59] Bhavya Ghai, Q Vera Liao, Yunfeng Zhang, Rachel Bellamy, and Klaus Mueller. Explainable active learning (xal) toward ai explanations as interfaces for machine teachers. *Proceedings of the ACM on Human-Computer Interaction*, 4(CSCW3):1–28, 2021. 8, 10

[60] Asma Ghandeharioun, Been Kim, Chun-Liang Li, Brendan Jou, Brian Eoff, and Rosalind W Picard. Dissect: Disentangled simultaneous explanations via concept traversals. *arXiv preprint arXiv:2105.15164*, 2021. 2, 18, 19

[61] Amirata Ghorbani, James Wexler, James Zou, and Been Kim. Towards automatic concept-based explanations. *arXiv preprint arXiv:1902.03129*, 2019. 18, 19, 67, 68

[62] Tom Goldstein and Stanley Osher. The split bregman method for l1-regularized problems. *SIAM journal on imaging sciences*, 2(2):323–343, 2009. 30

[63] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014. 15, 16

[64] Yash Goyal, Amir Feder, Uri Shalit, and Been Kim. Explaining classifiers with causal concept effect (cace). *ArXiv*, abs/1907.07165, 2019. 17, 19

[65] Roger Grosse, Micah K Johnson, Edward H Adelson, and William T Freeman. Ground truth dataset and baseline evaluations for intrinsic image algorithms. In *2009 IEEE 12th International Conference on Computer Vision*, pages 2335–2342. IEEE, 2009. xiv, 30, 32, 33, 36, 45, 46, 47

[66] David Gunning. Explainable artificial intelligence (xai). *Defense advanced research projects agency (DARPA), nd Web*, 2(2):1, 2017. 1

[67] Avani Gupta and P J Narayana. A systematic review of concept based approaches. *arXiv preprint arXiv:XXXX.XXXX*, 2023. 3, 19, 20

[68] Jin ha Lee, Ik hee Shin, Sang gu Jeong, Seung-Ik Lee, Muhamamad Zaigham Zaheer, and Beom-Su Seo. Improvement in deep networks for optimization using explainable artificial intelligence. In *2019 International Conference on Information and Communication Technology Convergence (ICTC)*, pages 525–530. IEEE, 2019. 20

[69] Peter Hase and Mohit Bansal. When can models learn from explanations? a formal framework for understanding the roles of explanation data. *arXiv preprint arXiv:2102.02201*, 2021. 20

[70] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1

[71] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 11, 13, 67

[72] Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507, 2006. 54

[73] P Hitzler and M Sarker. Human-centered concept explanations for neural networks. *Neuro-Symbolic Artificial Intelligence: The State of the Art*, 342(337):2, 2022. 19, 50

[74] Lars Holmberg, Paul Davidsson, and Per Linde. Mapping knowledge representations to concepts: A review and new perspectives. *ArXiv*, abs/2301.00189, 2022. 19, 20

[75] Shichao Jia, Peiwen Lin, Zeyu Li, Jiawan Zhang, and Shixia Liu. Visualizing surrogate decision trees of convolutional neural networks. *Journal of Visualization*, 23:141–156, 2019. 17

[76] Vidhya Kamakshi, Uday Gupta, and N. C. Krishnan. Pace: Posthoc architecture-agnostic concept extractor for explaining cnns. *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2021. 18

[77] Dmitry Kazhdan, Botty Dimanov, Mateja Jamnik, Pietro Liò, and Adrian Weller. Now you see me (cme): concept-based model extraction. *arXiv preprint arXiv:2010.13233*, 2020. xi, 20, 21, 23, 33, 50

[78] Dmitry Kazhdan, B. Dimanov, Helena Andrés-Terré, Mateja Jamnik, Pietro Lio', and Adrian Weller. Is disentanglement all you need? comparing concept-based & disentanglement approaches. *ArXiv*, abs/2104.06917, 2021. 1

[79] Monish Keswani, Sriranjani Ramakrishnan, Nishant Reddy, and Vineeth N Balasubramanian. Proto2proto: Can you recognize the car, the way i do? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10233–10243, 2022. x, 12, 13, 20, 21, 25, 29

[80] Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, et al. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *International conference on machine learning*, pages 2668–2677. PMLR, 2018. x, xi, 2, 3, 4, 5, 13, 14, 17, 18, 19, 26, 33, 34, 36, 47, 50, 51, 52, 53, 56, 58, 61, 63, 65

[81] Eungyeup Kim, Jihyeon Lee, and Jaegul Choo. Biaswap: Removing dataset bias with bias-tailored swapping augmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14992–15001, 2021. xv, 6, 52, 57, 60

[82] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 65

[83] Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. In *International Conference on Machine Learning*, pages 5338–5348. PMLR, 2020. x, 20, 21, 22, 50

[84] Avinash Kori, Ben Glocker, and Francesca Toni. Visual debates. *arXiv preprint arXiv:2210.09015*, 2022. 17

[85] Avinash Kori, Parth Natekar, Balaji Srinivasan, and Ganapathy Krishnamurthi. Interpreting deep neural networks for medical imaging using concept graphs. In *AI for Disease Surveillance and Pandemic Intelligence: Intelligent Disease Detection in Action*, pages 201–216. Springer, 2022. 17, 18

[86] Balazs Kovacs, Sean Bell, Noah Snavely, and Kavita Bala. Shading annotations in the wild. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 850–859, 2017. 66

[87] Balazs Kovacs, Sean Bell, Noah Snavely, and Kavita Bala. Shading annotations in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6998–7007, 2017. 30, 31, 32, 33

[88] Jan Kronenberger and Anselm Haselhoff. Dependency decomposition and a reject option for explainable models. *arXiv preprint arXiv:2012.06523*, 2020. 20, 21, 27

[89] Vivek Kwatra, Mei Han, and Shengyang Dai. Shadow removal for aerial imagery by information theoretic intrinsic image analysis. In *2012 IEEE International Conference on Computational Photography (ICCP)*, pages 1–8. IEEE, 2012. 32

[90] MCJJ L EH. Lightness and retinex theory. *J. Opt. Soc. Am.*, 61(1):1–11, 1971. 14, 32, 52

[91] Pierre-Yves Laffont, Adrien Bousseau, and George Drettakis. Rich intrinsic image decomposition of outdoor scenes from multiple views. *IEEE transactions on visualization and computer graphics*, 19(2):210–224, 2012. 30

[92] Isaac Lage and Finale Doshi-Velez. Learning interpretable concept-based models with human feedback. *arXiv preprint arXiv:2012.02898*, 2020. 20, 21, 25

[93] Brenden M Lake, Tomer D Ullman, Joshua B Tenenbaum, and Samuel J Gershman. Building machines that learn and think like people. *Behavioral and brain sciences*, 40, 2017. 1, 29

[94] Edwin H Land and John J McCann. Lightness and retinex theory. *Josa*, 61(1):1–11, 1971. 30

[95] Oran Lang, Yossi Gandelsman, Michal Yarom, Yoav Wald, Gal Elidan, Avinatan Hassidim, William T Freeman, Phillip Isola, Amir Globerson, Michal Irani, et al. Explaining in style: Training a gan to explain a classifier in stylespace. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 693–702, 2021. 18

[96] Sebastian Lapuschkin, Stephan Wäldchen, Alexander Binder, Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. Unmasking clever hans predictors and assessing what machines really learn. *Nature communications*, 10(1):1096, 2019. 5

[97] Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998. 64

[98] Jungsoo Lee, Eungyeup Kim, Juyoung Lee, Jihyeon Lee, and Jaegul Choo. Learning debiased representation via disentangled feature augmentation. *Advances in Neural Information Processing Systems*, 34:25123–25133, 2021. 6, 28, 57, 60, 61, 65

[99] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Deeper, broader and artier domain generalization. In *Proceedings of the IEEE international conference on computer vision*, pages 5542–5550, 2017. 27

[100] Haoliang Li, Sinno Jialin Pan, Shiqi Wang, and Alex C Kot. Domain generalization with adversarial feature learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5400–5409, 2018. 27

[101] Junbing Li, Changqing Zhang, Joey Tianyi Zhou, Huazhu Fu, Shuyin Xia, and Qinghua Hu. Deep-lift: deep label-specific feature learning for image annotation. *IEEE Transactions on Cybernetics*, 52(8):7732–7741, 2021. 20

[102] Junnan Li, Pan Zhou, Caiming Xiong, and Steven CH Hoi. Prototypical contrastive learning of unsupervised representations. *arXiv preprint arXiv:2005.04966*, 2020. 51, 54

[103] Oscar Li, Hao Liu, Chaofan Chen, and Cynthia Rudin. Deep learning for case-based reasoning through prototypes: A neural network that explains its predictions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018. 17

[104] Pan Li, Da Li, Wei Li, Shaogang Gong, Yanwei Fu, and Timothy M Hospedales. A simple feature augmentation for domain generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8886–8895, 2021. 27

[105] Yi Li and Nuno Vasconcelos. Repair: Removing representation bias by dataset resampling. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9572–9581, 2019. 52, 57

[106] Yu Li and Michael S Brown. Single image layer separation using relative smoothness. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2752–2759, 2014. 30, 32

[107] Yundong Li, Longxia Guo, and Yizheng Ge. Pseudo labels for unsupervised domain adaptation: A review. *Electronics*, 12(15):3325, 2023. 51

[108] Zhengqi Li and Noah Snavely. Cgintrinsics: Better intrinsic image decomposition through physically-based rendering. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 371–387, 2018. xi, 30, 32, 36, 38, 39, 40

[109] Zhengqi Li and Noah Snavely. Learning intrinsic image decomposition from watching the world. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9039–9048, 2018. xi, 30, 32, 36, 38, 39, 40, 45

[110] Zhiheng Li, Anthony Hoogs, and Chenliang Xu. Discover and mitigate unknown biases with debiasing alternate networks. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XIII*, pages 270–288. Springer, 2022. 6, 28

[111] 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 *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 19

[112] Pantelis Linardatos, Vasilis Papastefanopoulos, and Sotiris Kotsiantis. Explainable ai: A review of machine learning interpretability methods. *Entropy*, 23(1):18, 2020. 1, 15

[113] Pantelis Linardatos, Vasilis Papastefanopoulos, and Sotiris B. Kotsiantis. Explainable ai: A review of machine learning interpretability methods. *Entropy*, 23, 2021. 16

[114] Qiuhua Liu, Xuejun Liao, and Lawrence Carin. Semi-supervised multitask learning. *Advances in Neural Information Processing Systems*, 20, 2007. 23

[115] Xiaopei Liu, Liang Wan, Yingge Qu, Tien-Tsin Wong, Stephen Lin, Chi-Sing Leung, and Pheng-Ann Heng. Intrinsic colorization. In *ACM SIGGRAPH Asia 2008 papers*, pages 1–9. 2008. 32

[116] Yunfei Liu, Yu Li, Shaodi You, and Feng Lu. Unsupervised learning for intrinsic image decomposition from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3248–3257, 2020. xi, 30, 32, 36, 38, 39, 40

[117] Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. *Advances in Neural Information Processing Systems*, 33:11525–11538, 2020. 18, 25

[118] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017. 2

[119] Emanuele Marconato, Gianpaolo Bontempo, Elisa Ficarra, Simone Calderara, Andrea Passerini, and Stefano Teso. Neuro symbolic continual learning: Knowledge, reasoning shortcuts and concept rehearsal. *ArXiv*, abs/2302.01242, 2023. 17

[120] Emanuele Marconato, Gianpaolo Bontempo, Elisa Ficarra, Simone Calderara, Andrea Passerini, and Stefano Teso. Neuro symbolic continual learning: Knowledge, reasoning shortcuts and concept rehearsal. *arXiv preprint arXiv:2302.01242*, 2023. 17

[121] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. A survey on bias and fairness in machine learning. *ACM Computing Surveys (CSUR)*, 54(6):1–35, 2021. 5

[122] Tim Miller. Explanation in artificial intelligence: Insights from the social sciences. *Artificial intelligence*, 267:1–38, 2019. 2

[123] Christoph Molnar, Giuseppe Casalicchio, and Bernd Bischl. Interpretable machine learning–a brief history, state-of-the-art and challenges. In *ECML PKDD 2020 Workshops: Workshops of the European Conference on Machine Learning and Knowledge Discovery in Databases (ECML PKDD 2020): SoGood 2020, PDFL 2020, MLCS 2020, NFMCP 2020, DINA 2020, EDML 2020, XKDD 2020 and INRA 2020, Ghent, Belgium, September 14–18, 2020, Proceedings*, pages 417–431. Springer, 2021. 1

[124] Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. Domain generalization via invariant feature representation. In *International conference on machine learning*, pages 10–18. PMLR, 2013. 27

[125] W. James Murdoch, Peter J. Liu, and Bin Yu. Beyond word importance: Contextual decomposition to extract interactions from LSTMs. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=rkRwGg-0Z. 29

[126] Lukas Murmann, Michael Gharbi, Miika Aittala, and Fredo Durand. A multi-illumination dataset of indoor object appearance. In *2019 IEEE International Conference on Computer Vision (ICCV)*, Oct 2019. xii, 36, 40, 41

[127] Gregory L Murphy. The big book of concepts. a bradford book, 2002. 1

[128] Gayda Mutahar and Tim Miller. Concept-based explanations using non-negative concept activation vectors and decision tree for cnn models. *ArXiv*, abs/2211.10807, 2022. 17

[129] Vineel Nagisetty, Laura Graves, Joseph Scott, and Vijay Ganesh. xai-gan: Enhancing generative adversarial networks via explainable ai systems. *arXiv preprint arXiv:2002.10438*, 2020. 20

[130] Takuya Narihira, Michael Maire, and Stella X Yu. Direct intrinsics: Learning albedo-shading decomposition by convolutional regression. In *Proceedings of the IEEE international conference on computer vision*, pages 2992–2992, 2015. 30

[131] Islam Nassar, Munawar Hayat, Ehsan Abbasnejad, Hamid Rezatofighi, and Gholamreza Haffari. Protocon: Pseudo-label refinement via online clustering and prototypical consistency for efficient

semi-supervised learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11641–11650, 2023. 51

[132] Chuang Niu, Hongming Shan, and Ge Wang. Spice: Semantic pseudo-labeling for image clustering. *IEEE Transactions on Image Processing*, 31:7264–7278, 2022. 51

[133] Matthew R. O'Shaughnessy, Gregory H. Canal, Marissa Connor, Mark A. Davenport, and Christopher J. Rozell. Generative causal explanations of black-box classifiers. *ArXiv*, abs/2006.13913, 2020. 18

[134] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. URL http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf. 36

[135] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011. 56

[136] Pengzhen Ren, Yun Xiao, Xiaojun Chang, Po-Yao Huang, Zhihui Li, Brij B Gupta, Xiaojiang Chen, and Xin Wang. A survey of deep active learning. *ACM computing surveys (CSUR)*, 54(9): 1–40, 2021. 10

[137] 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*, pages 1135–1144, 2016. 2, 15, 16

[138] Laura Rieger, Chandan Singh, William Murdoch, and Bin Yu. Interpretations are useful: penalizing explanations to align neural networks with prior knowledge. In *International conference on machine learning*, pages 8116–8126. PMLR, 2020. 20, 28, 29, 52, 57, 58, 59, 60, 65

[139] Andrew Slavin Ross, Michael C Hughes, and Finale Doshi-Velez. Right for the right reasons: Training differentiable models by constraining their explanations. *arXiv preprint arXiv:1703.03717*, 2017. 20, 27, 28, 29, 51, 52, 57, 58, 59

[140] Muhammad Sabih, Frank Hannig, and Juergen Teich. Utilizing explainable ai for quantization and pruning of deep neural networks. *arXiv preprint arXiv:2008.09072*, 2020. 20

[141] Mikołaj Sacha, Dawid Rymarczyk, Łukasz Struski, Jacek Tabor, and Bartosz Zieliński. Protoseg: Interpretable semantic segmentation with prototypical parts. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1481–1492, 2023. 20, 21, 26

[142] Shiori Sagawa, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *arXiv preprint arXiv:1911.08731*, 2019. 68

[143] Gobinda Saha and Kaushik Roy. Saliency guided experience packing for replay in continual learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 5273–5283, 2023. 2, 3

[144] Saurabh Saini and P. J. Narayanan. Semantic priors for intrinsic image decomposition. In *Brit. Mach. Vis. Conf.*, 2018. 30

[145] Saurabh Saini and PJ Narayanan. Semantic hierarchical priors for intrinsic image decomposition. *arXiv preprint arXiv:1902.03830*, 2019. 30

[146] Saurabh Saini, Parikshit Sakurikar, and P. J. Narayanan. Intrinsic image decomposition using focal stacks. In *Indian Conference on Computer Vision, Graphics and Image Processing (ICVGIP)*, 2016. 30

[147] Wojciech Samek, Thomas Wiegand, and Klaus-Robert Müller. Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models. *arXiv preprint arXiv:1708.08296*, 2017. 15, 16

[148] Ainkaran Santhirasekaram, Avinash Kori, Andrea Rockall, Mathias Winkler, Francesca Toni, and Ben Glocker. Hierarchical symbolic reasoning in hyperbolic space for deep discriminative models. *arXiv preprint arXiv:2207.01916*, 2022. 17

[149] Anirban Sarkar, Deepak Vijaykeerthy, Anindya Sarkar, and Vineeth N Balasubramanian. A framework for learning ante-hoc explainable models via concepts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10286–10295, 2022. 21, 25

[150] Yoshihide Sawada and Keigo Nakamura. C-senn: Contrastive self-explaining neural network. *ArXiv*, abs/2206.09575, 2022. 20, 21, 25

[151] Yoshihide Sawada and Keigo Nakamura. Concept bottleneck model with additional unsupervised concepts. *IEEE Access*, 10:41758–41765, 2022. 3, 20, 21, 24

[152] Patrick Schramowski, Wolfgang Stammer, Stefano Teso, Anna Brugger, Franziska Herbert, Xiaoting Shao, Hans-Georg Luigs, Anne-Katrin Mahlein, and Kristian Kersting. Making deep neural networks right for the right scientific reasons by interacting with their explanations. *Nature Machine Intelligence*, 2(8):476–486, 2020. 20, 21, 27

[153] Jessica Schrouff, Sebastien Baur, Shaobo Hou, Diana Mincu, Eric Loreaux, Ralph Blanes, James Wexler, Alan Karthikesalingam, and Been Kim. Best of both worlds: local and global explanations with human-understandable concepts. *arXiv preprint arXiv:2106.08641*, 2021. 17, 19

[154] Gesina Schwalbe. Concept embedding analysis: A review. *arXiv preprint arXiv:2203.13909*, 2022. 19

[155] Gesina Schwalbe and Bettina Finzel. A comprehensive taxonomy for explainable artificial intelligence: a systematic survey of surveys on methods and concepts. *Data Mining and Knowledge Discovery*, pages 1–59, 2023. 1, 15

[156] Gesina Schwalbe and Ute Schmid. Concept enforcement and modularization for the iso 26262 safety case of neural networks. 2020. 3

[157] 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 *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017. x, xiii, 2, 3, 15, 16, 19, 58

[158] Ramprasaath R Selvaraju, Stefan Lee, Yilin Shen, Hongxia Jin, Shalini Ghosh, Larry Heck, Dhruv Batra, and Devi Parikh. Taking a hint: Leveraging explanations to make vision and language models more grounded. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2591–2600, 2019. 20

[159] Xiaoting Shao, Karl Stelzner, and Kristian Kersting. Right for the right latent factors: Debiasing generative models via disentanglement. *arXiv preprint arXiv:2202.00391*, 2022. 20, 27

[160] Radwa El Shawi, Youssef Mohamed, and Sherif Sakr. Towards automated concept-based decision treeexplanations for cnns. In *International Conference on Extending Database Technology*, 2021. 17, 18

[161] Li Shen, Ping Tan, and Stephen Lin. Intrinsic image decomposition with non-local texture cues. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–7. IEEE, 2008. 32

[162] Yujun Shen, Ceyuan Yang, Xiaoou Tang, and Bolei Zhou. Interfacegan: Interpreting the disentangled face representation learned by gans. *IEEE transactions on pattern analysis and machine intelligence*, 44(4):2004–2018, 2020. 18

[163] Rui Shu, Yining Chen, Abhishek Kumar, Stefano Ermon, and Ben Poole. Weakly supervised disentanglement with guarantees. *arXiv preprint arXiv:1910.09772*, 2019. 20, 27

[164] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016. 1

[165] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*, 2017. 2, 3

[166] Youngjae Song, Sung Kuk Shyn, and Kwang-su Kim. Img2tab: Automatic class relevant concept discovery from stylegan features for explainable image classification. *arXiv preprint arXiv:2301.06324*, 2023. 18, 20, 21, 26

[167] Rahul Soni, Naresh Shah, Chua Tat Seng, and Jimmy D. Moore. Adversarial tcav - robust and effective interpretation of intermediate layers in neural networks. *ArXiv*, abs/2002.03549, 2020. 17, 19

[168] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*, 2014. 3

[169] Johannes Stallkamp, Marc Schlipsing, Jan Salmen, and Christian Igel. Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. *Neural networks*, 32:323–332, 2012. 27

[170] Wolfgang Stammer, Patrick Schramowski, and Kristian Kersting. Right for the right concept: Revising neuro-symbolic concepts by interacting with their explanations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3619–3629, 2021. 17, 18, 20, 21, 25, 27, 29

[171] Wolfgang Stammer, Marius Memmel, Patrick Schramowski, and Kristian Kersting. Interactive disentanglement: Learning concepts by interacting with their prototype representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10317–10328, 2022. 18, 20, 25, 29

[172] Jiamei Sun, Sebastian Lapuschkin, Wojciech Samek, Yunqing Zhao, Ngai-Man Cheung, and Alexander Binder. Explanation-guided training for cross-domain few-shot classification. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 7609–7616. IEEE, 2021. 29

[173] Jiamei Sun, Sebastian Lapuschkin, Wojciech Samek, and Alexander Binder. Explain and improve: Lrp-inference fine-tuning for image captioning models. *Information Fusion*, 77:233–246, 2022. 29

[174] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International conference on machine learning*, pages 3319–3328. PMLR, 2017. 2, 3

[175] Korawat Tanwisuth, Xinjie Fan, Huangjie Zheng, Shujian Zhang, Hao Zhang, Bo Chen, and Mingyuan Zhou. A prototype-oriented framework for unsupervised domain adaptation. *Advances in Neural Information Processing Systems*, 34:17194–17208, 2021. 51

[176] Enzo Tartaglione, Carlo Alberto Barbano, and Marco Grangetto. End: Entangling and disentangling deep representations for bias correction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13508–13517, 2021. 57, 60

[177] Stefano Teso. Toward faithful explanatory active learning with self-explainable neural nets. In *Proceedings of the Workshop on Interactive Adaptive Learning (IAL 2019)*, pages 4–16. CEUR Workshop Proceedings, 2019. 20, 21, 27

[178] Stefano Teso and Kristian Kersting. Explanatory interactive machine learning. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, pages 239–245, 2019. 8, 9, 27

[179] Stefano Teso, Öznur Alkan, Wolfgang Stammer, and Elizabeth M. Daly. Leveraging explanations in interactive machine learning: An overview. *ArXiv*, abs/2207.14526, 2022. 20, 25

[180] Thien Q. Tran, Kazuto Fukuchi, Youhei Akimoto, and Jun Sakuma. Unsupervised causal binary concepts discovery with vae for black-box model explanation. In *AAAI Conference on Artificial Intelligence*, 2021. 18

[181] Vadim Tschernezki, Iro Laina, Diane Larlus, and Andrea Vedaldi. Neural feature fusion fields: 3d distillation of self-supervised 2d image representations. *arXiv preprint arXiv:2209.03494*, 2022. 64

[182] Vadim Tschernezki, Iro Laina, Diane Larlus, and Andrea Vedaldi. Neural feature fusion fields: 3D distillation of self-supervised 2D image representations. In *Proceedings of the International Conference on 3D Vision (3DV)*, 2022. 64

[183] Johanna Vielhaben, Stefan Blücher, and Nils Strodthoff. Multi-dimensional concept discovery (mcd): A unifying framework with completeness guarantees. *ArXiv*, abs/2301.11911, 2023. 18

[184] Stanislav Vojíř and Tomáš Kliegr. Editable machine learning models? a rule-based framework for user studies of explainability. *Advances in Data Analysis and Classification*, 14(4):785–799, 2020. 1, 15

[185] Andong Wang, Wei-Ning Lee, and Xiaojuan Qi. Hint: Hierarchical neuron concept explainer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10254–10264, 2022. 17, 18

[186] Chong Wang, Yuyuan Liu, Yuanhong Chen, Fengbei Liu, Yu Tian, Davis J McCarthy, Helen Frazer, and Gustavo Carneiro. Learning support and trivial prototypes for interpretable image classification. *arXiv preprint arXiv:2301.04011*, 2023. 20, 21, 25

[187] Dan Wang, Xinrui Cui, and Z Jane Wang. Chain: Concept-harmonized hierarchical inference interpretation of deep convolutional neural networks. *arXiv preprint arXiv:2002.01660*, 2020. 17

[188] Xinye Wanyan, Sachith Seneviratne, Shuchang Shen, and Michael Kirley. Dino-mc: Self-supervised contrastive learning for remote sensing imagery with multi-sized local crops. *arXiv preprint arXiv:2303.06670*, 2023. 64

[189] Leander Weber, Sebastian Lapuschkin, Alexander Binder, and Wojciech Samek. Beyond explaining: Opportunities and challenges of xai-based model improvement. *Information Fusion*, 2022. x, 1, 9, 10, 20

[190] Mike Wu, Michael Hughes, Sonali Parbhoo, Maurizio Zazzi, Volker Roth, and Finale Doshi-Velez. Beyond sparsity: Tree regularization of deep models for interpretability. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018. 15, 16

[191] Mengqi Xue, Qihan Huang, Haofei Zhang, Lechao Cheng, Jie Song, Minghui Wu, and Mingli Song. Protopformer: Concentrating on prototypical parts in vision transformers for interpretable image recognition. *arXiv preprint arXiv:2208.10431*, 2022. 20, 21, 25

[192] Linda Yang, Baohua Huang, Shiqian Guo, Yunjie Lin, and Tong Zhao. A small-sample text classification model based on pseudo-label fusion clustering algorithm. *Applied Sciences*, 13(8): 4716, 2023. 51, 54

[193] Chih-Kuan Yeh, Been Kim, Sercan Ö. Arik, Chun-Liang Li, Tomas Pfister, and Pradeep Ravikumar. On completeness-aware concept-based explanations in deep neural networks. *arXiv: Learning*, 2019. 17, 19

[194] Chih-Kuan Yeh, Been Kim, Sercan Arik, Chun-Liang Li, Tomas Pfister, and Pradeep Ravikumar. On completeness-aware concept-based explanations in deep neural networks. *Advances in Neural Information Processing Systems*, 33:20554–20565, 2020. 18, 19

[195] Seul-Ki Yeom, Philipp Seegerer, Sebastian Lapuschkin, Alexander Binder, Simon Wiedemann, Klaus-Robert Müller, and Wojciech Samek. Pruning by explaining: A novel criterion for deep neural network pruning. *Pattern Recognition*, 115:107899, 2021. 20

[196] Quanshi Zhang, Ruiming Cao, Feng Shi, Ying Nian Wu, and Song-Chun Zhu. Knowledge via an explanatory graph. 2017. 17

[197] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *IEEE/CVF Conf. Comput. Vis. Pattern Recog.*, 2018. 40, 45

[198] Ruihan Zhang, Prashan Madumal, Tim Miller, Krista A Ehinger, and Benjamin IP Rubinstein. Invertible concept-based explanations for cnn models with non-negative concept activation vectors. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 11682–11690, 2021. 17, 19

[199] Yu Zhang, Peter Tiňo, Aleš Leonardis, and Ke Tang. A survey on neural network interpretability. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 5(5):726–742, 2021. x, 1, 15, 16, 50

[200] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929, 2016. 20

[201] Bolei Zhou, Yiyou Sun, David Bau, and Antonio Torralba. Interpretable basis decomposition for visual explanation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 119–134, 2018. 17

[202] Tinghui Zhou, Philipp Krahenbuhl, and Alexei A Efros. Learning data-driven reflectance priors for intrinsic image decomposition. In *Proceedings of the IEEE international conference on computer vision*, pages 3469–3477, 2015. 30