Un cadre flexible pour l'apprentissage automatique interprétable : application à la classification d'images et d'audio

Jayneel Parekh

INSTITUT POLYTECHNIQUE DE PARIS

NNT : 2023IPPAT032

Thèse de doctorat

A Flexible Framework for Interpretable Machine Learning: Application to Image and Audio Classification

Thèse de doctorat de l'Institut Polytechnique de Paris
préparée à Télécom Paris

École doctorale n°626 École doctorale de l'Institut Polytechnique de Paris (ED IP Paris)
Spécialité de doctorat : Informatique

Thèse présentée et soutenue à Palaiseau, le 07 juillet 2023, par

JAYNEEL PAREKH

Composition du Jury :

Stéphane Canu Professeur, INSA Rouen (LITIS)	Président / Examinateur
Grégoire Montavon Professeur, Freie Universität Berlin	Rapporteur
Nicolas Thome Professeur, Sorbonne Université (ISIR)	Rapporteur
Patrick Pérez Directeur de Recherche, Valeo IA	Examinateur
David Alvarez-Melis Chargé de Recherche, Microsoft Research	Examinateur
Chloé Clavel Professeure, Télécom Paris (LTCI)	Examinatrice
Florence d'Alché-Buc Professeure, Télécom Paris (LTCI)	Directrice de thèse
Pavlo Mozharovskyi Maître de Conference, Télécom Paris (LTCI)	Co-directeur de thèse
Gaël Richard Professeur, Télécom Paris (LTCI)	Invité

Contents

Abstract

Machine learning systems, and specially neural networks, have rapidly grown in their ability to address complex learning problems. Consequently, they are being integrated into society with an ever-rising influence on all levels of human experience. This has resulted in a need to gain human-understandable insights in their decision making process to ensure the decisions are being made ethically and reliably. The study and development of methods which can generate such insights broadly constitutes the field of interpretable machine learning.

This thesis aims to develop a novel framework that can tackle two major problem settings in this field, post-hoc and by-design interpretation. We particularly tackle these problems in the context of deep neural networks. Post-hoc interpretability devises methods to interpret decisions of a pre-trained predictive model, while by-design interpretability targets to learn a single model capable of both prediction and interpretation. To this end, we extend the traditional supervised learning formulation to include interpretation as an additional task besides prediction, each addressed by separate but related models, a predictor and an interpreter. To learn to solve both tasks simultaneously, we propose dedicated loss functions for each one of them. Crucially, the interpreter is dependent on the predictor through its hidden layers and utilizes a dictionary of concepts as its representation for interpretation. The training of the framework is centered around learning the dictionary of concepts by formulating the interpretability loss function, which is constructed through a minimal set of properties implemented as losses. We additionally define a notion of local and global relevance for each concept to the classification decision. This helps us enable generation of both local and global interpretations.

The framework is separately instantiated to address interpretability problems in the context of image and audio classification. We demonstrate high predictive performance and fidelity of interpretations in both cases. This can be strongly attributed to the access of hidden layers by the interpreter. Despite adhering to the same underlying structure the two systems are distinctly designed for interpretations. The image interpretability system advances the pipeline for visualizing/discovering learnt concepts for improved understandability. In particular, we propose an activation maximization based tool to strongly emphasize detected patterns for visualization of concepts. Our proposed visualization pipeline is qualitatively evaluated through a subjective evaluation. Furthermore, we employ a novel criterion based on entropy

of concept activations to improve conciseness of interpretations. The image interpretation system is extensively evaluated on various popular and publicly available classification benchmarks. The audio interpretability system instead tackles a different goal. It is designed to facilitate listenable interpretations whilst modeling audio objects composing a scene. This is particularly important since visual saliency maps are not understandable for most end users. In order to achieve this, we propose a novel representation for dictionary of concepts based on non-negative matrix factorization (NMF). The interpreter learns to predict an embedding that corresponds to time activations of a NMF decomposition of the input. Our formulation leads to a simple pipeline to generate listenable interpretations. In the context of NMF literature, our method presents a unique way to link NMF and deep neural network representations. The unique structure also grants it the ability to evaluate faithfulness for post-hoc interpretations. The system is evaluated against multiple baselines on a diverse set of audio classification tasks, including environmental sound classification and music instrument tagging. For both systems individually, we analyze the impact of various hyperparameter choices and discuss ways to deepen them or broaden the scope of our general framework.

Résumé

Les systèmes d'apprentissage automatique, et en particulier les réseaux de neurones, ont rapidement développé leur capacité à résoudre des problèmes d'apprentissage complexes. Par conséquent, ils sont intégrés dans la société avec une influence de plus en plus grande sur tous les niveaux de l'expérience humaine. Cela a entraîné la nécessité d'acquérir des informations compréhensibles par l'homme dans leur processus de prise de décision pour s'assurer que les décisions soient prises de manière éthique et fiable. L'étude et le développement de méthodes capables de générer de telles informations constituent de manière générale le domaine de l'apprentissage automatique interprétable.

Cette thèse vise à développer un nouveau cadre pour aborder deux problématiques majeures dans ce domaine, l'interprétabilité post-hoc et par conception. Nous abordons particulièrement ces problèmes dans le contexte des réseaux de neurones profonds. L'interprétabilité post-hoc conçoit des méthodes pour analyser les décisions d'un modèle prédictif pré-entraîné, tandis que l'interprétabilité par conception vise à apprendre un modèle unique capable à la fois de prédiction et d'interprétation. Pour ce faire, nous étendons la formulation traditionnelle de l'apprentissage supervisé pour inclure l'interprétation en tant que tâche supplémentaire en plus de la prédiction, chacune étant traitée par des modèles distincts, mais liés: un prédicteur et un interpréteur. Pour apprendre à résoudre les deux tâches simultanément, nous proposons des fonctions de perte dédiées à chacune d'elles. Fondamentalement, l'interpréteur dépend du prédicteur à travers ses couches cachées et utilise un dictionnaire de concepts comme représentation pour l'interprétation. La formation du cadre se concentre sur l'apprentissage du dictionnaire de concepts en formulant la fonction de perte d'interprétabilité, qui est construite à travers un ensemble minimal de propriétés implémentées en tant que pertes. Nous définissons en outre une notion de pertinence locale et globale pour chaque concept dans la décision de classification. Cela nous aide à permettre la génération d'interprétations locales et globales.

Le cadre est instancié séparément pour résoudre les problèmes d'interprétation dans le contexte de la classification d'images et de sons. Dans les deux cas, nous démontrons des performances de prédiction élevées, ainsi qu'une haute fidélité des interprétations. Cela peut être fortement attribué à l'accès des couches cachées par l'interpréteur. Bien qu'ils adhèrent à la même structure sous-jacente, les deux systèmes sont distinctement conçus pour l'interprétation. Le système d'interprétabilité

d'image fait avancer le protocole de visualisation/découverte de concepts appris pour une meilleure compréhensibilité. En particulier, nous proposons un outil basé sur la maximisation de l'activation pour mettre fortement l'accent sur les modèles détectés pour la visualisation des concepts. Notre pipeline de visualisation proposé est évalué qualitativement par le biais d'une évaluation subjective. De plus, nous utilisons un nouveau critère basé sur l'entropie des activations de concepts pour améliorer la concision des interprétations. Le système d'interprétation d'images est évalué de manière approfondie sur divers critères de classification populaires et accessibles au public. Le système d'interprétabilité audio poursuit plutôt un objectif différent. Il est conçu pour faciliter les interprétations écoutables tout en modélisant les objets audio composant une scène. Ceci est particulièrement important puisque les cartes de saillance visuelle ne sont pas compréhensibles pour la plupart des utilisateurs finaux. Pour y parvenir, nous proposons une nouvelle représentation du dictionnaire de concepts basée sur la factorisation matricielle non négative (NMF). L'interpréteur apprend à prédire un plongement qui correspond aux activations temporelles d'une décomposition NMF de l'entrée de données. Notre formulation mène à un pipeline simple pour générer des interprétations écoutables. Dans le contexte de la littérature NMF, notre méthode présente une manière unique de relier les représentations du NMF et des réseaux neuronaux profonds. La structure unique lui confère également la capacité d'évaluer la fidélité des interprétations post-hoc. Le système est évalué par rapport à plusieurs références sur un ensemble diversifié de tâches de classification audio, notamment la classification des sons environnementaux et l'étiquetage des instruments de musique. Pour les deux systèmes individuellement, nous analysons l'impact de divers choix d'hyperparamètres et discutons des moyens de les approfondir ou d'élargir la portée de notre cadre général.

(French translation of the abstract edited by Quentin Bouniot and Arturo Castellanos Salinas.)

1 Introduction

Contents

1.1 Motivation

At least more than a century ago, humans had created machines capable of remarkable physical feats. Yet, despite all the awe-inspiring progress, until a few decades ago, machines were unable to recognize simple patterns (Bishop et al., 1995; Webb, 2003). As noted by the great physicist Richard Feynman himself in a lecture in 1985, “To recognize things, to recognize patterns, seems to be something we have not been able to put into a definite procedure". In a sense, this marked the dawn of a new era. Supported by an astonishing increase in computational and data storage capabilities, the 21^{st} century is continuing to witness algorithms powering machines with learning capabilities that thoroughly outperform humans (Bishop and Nasrabadi, 2006; Goodfellow et al., 2016). This has reached a point where a human might be forgiven if they cannot instantly name a task they can do better than any machine.

Following the blueprint of any novel piece of technology in human history, these machines are being integrated in our society. They are automating and transforming all aspects of our lives, ranging from daily tasks to entire workplaces (Bank, 2018).

This leads us to a critical juncture. A major issue regarding use of these models stems from the fact that they are typically optimized for performance on their respective tasks. To accomplish this, they learn to compute complex features on the input data. This can make their decision process incomprehensible for humans. In turn, this can

carry significant consequences, even catastrophic in certain cases, as these models can directly influence human lives. As an example, imagine a person getting their loan application rejected by some algorithm deployed in a bank. This can heavily impact the person's life choices. If the model is making its decision based on the race of the person, the decision would be considered antithetical to values of human society. This is not limited to just a single sector or specific application. There are a host of such examples in the fields of healthcare, defence, finance etc., where machine learning model decisions determine aspects of human lives (Bhatt et al., 2020b).

Thus, for a variety of applications, there is a moral and ethical requirement to understand the prediction process of a model and ensure that it is taken based on relevant information in the input. This has led to the emergence of studying *interpretability* of machine learning models. However, it is not just ethical or legal (Voigt and Von dem Bussche, 2017) needs that grant value to this endeavour. It can act as a great analysis tool that can offer novel insights about a complex process (Sturm et al., 2016; Schütt et al., 2017). Furthermore, it presents great prospects to assist or even enhance human decision-making (Koh et al., 2020). Most importantly though, its foundations lie in the innate desire of humans to understand any process they observe. To be able to understand and comprehend a model they create will always hold an intrinsic value for them.

The need for interpretability cannot be better epitomized than by the increasing use of deep neural networks as a tool. On one hand they have completely altered the horizons of learning applications in many domains including computer vision, natural language processing, audio/music processing, graph processing, etc. (Dong et al., 2021). At the same time, their architectures with multiple layers computing increasingly complex non-linear features in an end-to-end setting, renders their learnt representations entirely incomprehensible to a user. This motivated us to specifically investigate interpretability for deep neural networks. In particular, we explore this with respect to image and audio classification. This course of action was not only founded upon the immense popularity of these approaches for image/audio processing (Purwins et al., 2019; Voulodimos et al., 2018), but also because they represent some of most fundamental applications of machine learning. Among our five perceptual senses, *vision* and *sound* are arguably the most informative senses for our brain (Howes, 2011). We rely on them most frequently, almost every waking moment, to navigate, interact and experience the world around us. It is thus inevitable that as machine learning aims to assist, enhance and interact with our senses, these domains present the most fertile grounds for applications.

1.2 Key Themes and Research Challenges

The goal of this section is to highlight the interesting themes and challenges that underlie this thesis, and in general any research on interpretability.

1.2.1 What is an Interpretation?

The previous section motivates why it is important to gain insights about a machine learning (ML) model, but we have not precisely defined yet what *interpretability* means. As of yet, there is no universal agreed or a mathematical definition of *interpretability* (the ability to *interpret*) (Molnar, 2020; Lipton, 2018). One of the popular definitions that we align with is the one given by Doshi-Velez and Kim (2017). For machine learning systems they define *interpretability* as the "*ability to explain or to present in understandable terms to a human*". In the context of classification systems this corresponds to presenting the decision process of the system in human-understandable terms. As an additional note, we would like to mention the very closely related term to interpretability, *explainability*. There is no universal agreement upon if these two terms are synonyms or carry subtle differences in meaning. The differences, if any, are not particularly consequential to our research. The most frequent usage of explainability and what information constitutes an explanation in machine learning literature is functionally identical to our usage of interpretability (Rudin et al., 2022).

However, there are two follow up questions the above definition should raise. Firstly, what information constitutes this presentation? And secondly, what does it mean to be human-understandable? Unfortunately, there are again no objective answers to both. Context plays a strong role in any attempt to reach an objective resolution. It is easier to see this for the latter question as "understandability" is a subjective notion, existing in relation to human cognition. Any piece of information that might be *understandable* for one person need not be for others. A calculus textbook is *understandable* for a university professor but not for kindergarten students. That does not imply that one cannot objectively work with the notion of *understandability*. Our human experience can often guide what understandability entails and for uncertain cases, operational evaluation might be possible. For example if the interpretations are intended for use ML practitioners, one can subjectively evaluate the *understandability* by asking appropriate questions to many practitioners. This inherent subjectivity adds a unique challenge to interpretability research in regards to design of algorithms, their use and evaluation. One of our goals, aligned with the conventional approach to best address the subjectivity, is to convert as much as possible the subjective understanding about interpretability to numerical properties that can be imposed/assessed.

Representations for Interpretation

Here, we would like to raise the issue of the language of interpretation. Similar to *understandability*, **what** information constitutes an interpretation is not objectively defined. Insightful information about a decision process can be generated in different forms or at different levels of granularity. For example, in case of image classification an interpretation can indicate which regions for a given input are relevant for its decision, in other words, *where a model focuses* for its decision. On the other hand, an interpretation can also inform about *what a model focuses* on, i.e. what detected patterns are responsible for the decision. Both of them can equally be considered as providing insight about the decision process but in different ways that might serve

different purposes. However, what is indeed more important is that any interpretation algorithm specifies its objectives and is evaluated accordingly (Lipton, 2018).

A large number of algorithms rely on employing a human-understandable representation and then quantifying the importance of its elements to understand the decision process. Thus, one way to characterize the different algorithms and interpretation they offer is through the representation/means of interpretation they use. The role of this choice is similar to that of language in human communication. The environment someone is in, for instance the city/country location, can guide which language is suitable for conversation. Similarly, different requirements originating from context of an application can dictate which representation is more suitable. This choice forms an important theme in design of our interpretation systems.

1.2.2 Interpretability problems and flexibility of methods

If one recognizes that interpretability of a machine learning model as an equally important goal as its predictive performance, it raises a broad issue of what is an effective way to achieve both goals This issue has been grappled with in two different ways by the research community that correspond to two different problem types.

The first one relies on the availability of a predictive model trained and optimized for performance but not for interpretability and aims to devise an additional approach to interpret the given model. This problem setting is usually referred to **post-hoc interpretation**. The other setting aims to build an interpretable predictive model from the data. The challenge here is to demonstrate high predictive performance while maintaining interpretability in the same model. This setting is often referred to as the **by-design interpretation** problem. However, real-world scenarios of utilizing interpretability of machine learning can occur under variety of constraints and demands regarding deployment, level of interpretability and performance. Post-hoc approaches sustain a huge demand in the industry since companies routinely prioritize performance in designing their models and interpretability can arise as a secondary but required objective. On the other hand, by-design interpretable predictive models represent a panacea in this regard and feature as a more suitable long-term goal (Rudin et al., 2022). They even take precedence over post-hoc interpretation methods for decision-critical applications wherein, interpretability is a primary objective. Thus, from a practical standpoint, both problem settings hold independent value.

Note that while these are the two major types of problem there are other factors that differentiate between different interpretability problems. For example, a common differentiation is related to *scope of interpretability*, which consists of whether we wish to interpret the decision for a single sample (local) or for the model as a whole (global). Prior research contains multiple methods that are flexible in regard to input data modality they process or the decision function they interpret. However, all of them can address only one of the major problems. One of the key challenges we tackle is to design a framework that can flexibly be used to address different types of inter-

pretability problems.

1.2.3 Modality specific challenges

We instantiate our framework for image and audio classification tasks to tackle the interpretability problems highlighted earlier. However, even with a common backbone for both system designs, operating on different modalities can impose different desiderata on the interpretation in regard to how it is supposed to be used by the user. These differences reflect themselves not only in our respective system designs but in the interpretation generation process too. Moreover, while the evaluation of interpretations can resemble for the two modalities given the common set of applications tackled, the quantitative and qualitative evaluations need to take into account the specific objectives the interpretation designs are supposed to fulfil.

1.3 Contributions and Outline

1.3.1 Publications

The content discussed in this thesis has been a part of the following publications:

Conference or Journal papers

1. Jayneel Parekh, Sanjeel Parekh, Pavlo Mozharovskyi, Florence d'Alché-Buc, and Gaël Richard. "Tackling Interpretability for Audio Classification Networks with Non-negative Matrix Factorization". IEEE/ACM TASLP (submitted).

2. Jayneel Parekh, Sanjeel Parekh, Pavlo Mozharovskyi, Florence d'Alché-Buc, and Gaël Richard. "Listen to Interpret: Post-hoc Interpretability for Audio Networks with NMF". NeurIPS 2022.

3. Jayneel Parekh, Pavlo Mozharovskyi, and Florence d'Alché-Buc. "A Framework to Learn with Interpretation". NeurIPS 2021.

Preprints or Workshop papers

1. Winston Maxwell, Valérie Beaudouin, Isabelle Bloch, David Bounie, Stéphan Clémençon, Florence d'Alché-Buc, James Eagan, Pavlo Mozharovskyi, and Jayneel Parekh. "Identifying the 'Right' Level of Explanation in a Given Situation." NeHuAI workshop, ECAI 2020.

2. Valérie Beaudouin, Isabelle Bloch, David Bounie, Stéphan Clémençon, Florence d'Alché-Buc, James Eagan, Winston Maxwell, Pavlo Mozharovskyi, and Jayneel Parekh. "Flexible and context-specific AI explainability: a multidisciplinary approach." arXiv preprint 2020 arXiv:2003.07703.

1.3.2 Outline

We now provide an outline for the thesis below

In chapter 2, we review the prior literature about interpretability from the point of view of different contextual factors of an interpretability application. It comprises of three parts: progression of methods with respect to different factors defining an interpretability problem, details about popular methods and different representations of interpretation they proposed, and coverage of strategies for evaluating interpretations. We cater our discussion more towards methods for interpretation for image/audio classification.

In chapter 3, we develop a single framework to tackle post-hoc and by-design interpretation. To do so, we build on top of empirical risk minimization formulation for supervised learning to include interpretation as an additional task. We propose to train a predictor, and a related interpreter through a single learning objective that trains for both prediction and interpretation. We then outline the structure of the interpreter which includes its dependence on predictor, representation of interpretation based on dictionary of concepts, properties and corresponding loss functions to formulate interpretability loss for training, and a novel notion of local or global relevance for interpretations. We conclude the chapter by discussing potential metrics to evaluate the interpretations

In chapter 4 we instantiate the components of our framework and apply it for post-hoc and by-design interpretation for image classification by learning a unsupervised dictionary of concepts. We propose a novel entropy based loss to improve conciseness of interpretations and develop a pipeline to understand/discover the concepts for local and global interpretations. We extensively evaluate the interpretations quantitatively on multiple and popular image classification benchmarks and demonstrate improved predictive performance, fidelity to predictor and conciseness compared to related methods. We also a devise a study to subjectively evaluate the understandability of interpretations.

In chapter 5 we instantiate the framework for interpretability problems on audio classification. We motivate the need to generate listenable concept-based interpretations for audio signals and introduce a novel means of interpretation based on non-negative matrix factorization (NMF). The NMF based formulation results in a simple pipeline to generate interpretations and allows the possibility to measure faithfulness for post-hoc interpretations. We again extensively evaluate our interpretations quantitatively and qualitatively on large-scale audio classification datasets including environmental audio and music data, showing improvement in terms of performance, fidelity, faithfulness and understandability compared to the relevant baselines.

We discuss the contributions, limitations and future perspectives in chapter 6. This is followed by a section on ongoing work to extend the image interpretability system with generative models.

2

Background and Related Works

Contents

2.1 Introduction

There are two aims of this chapter. The first is to get a birds-eye view, that covers a big chunk of interpretability literature in machine learning. The second is to get a finer look at some of the research areas, closely related to this thesis. It is worth stating at the start that we primarily focus on interpretability problems with supervised learning as underlying task.

We organize this information from the point of view of various contextual factors that constitute an interpretability application. A high-level flow of such a pipeline is given in Fig. 2.1. It consists of a model f and sample x or potentially a dataset, given as input to some interpretation generating algorithm. The generated interpretations are given as outputs for humans to understand the model's decision. An interpretation is thus a function of the model and the data itself. The natural question we then want to ask is: What are the different factors/details that contextualize this pipeline in case of any specific application? It could be the type of data being operated on, scope of interpretability, real time constraints etc. We give a brief list of these factors with some of their selected keywords below:

- **Problem type**: Post-hoc interpretability, by-design interpretability
- **Scope of interpretability**: Local, global, or both.
- **Model type**: Convolutional neural networks, Recurrent neural networks, random forests, etc.
- **Model accessibility**: White-box, gray-box, black-box.
- **Other problem defining features**: Extent and point of human intervention, real-time interpretability, training time.
- **Input data type**: Audio, graph, image, tabular, text, video.
- **Means of interpretation**: Raw input, simplified input, prototypes, concepts, language.
- **Evaluation metrics**: Faithfulness, complexity, stability, etc.

Loosely speaking, the list progresses from broad factors defining a problem to those describing specific traits of an interpretation algorithm and then finally the evaluation of the pipeline. Many might consider some of the factors describing a problem as specific features of the algorithm itself, which is why there are no strict distinctions to be adhered to. The above information is summarized in Fig. 2.2. The rest of the chapter will delve into greater details about each of these factors and how they partition the literature:

2.2 Characteristics of the problem

2.2.1 Problem type

This is the broadest category one can construct to demarcate interpretability problems. There are primarily two types of problems: *post-hoc interpretation* and *by-design interpretation*. The former is tasked with generating interpretations for a pre-trained model by devising external methods for it, while the latter requires building a single

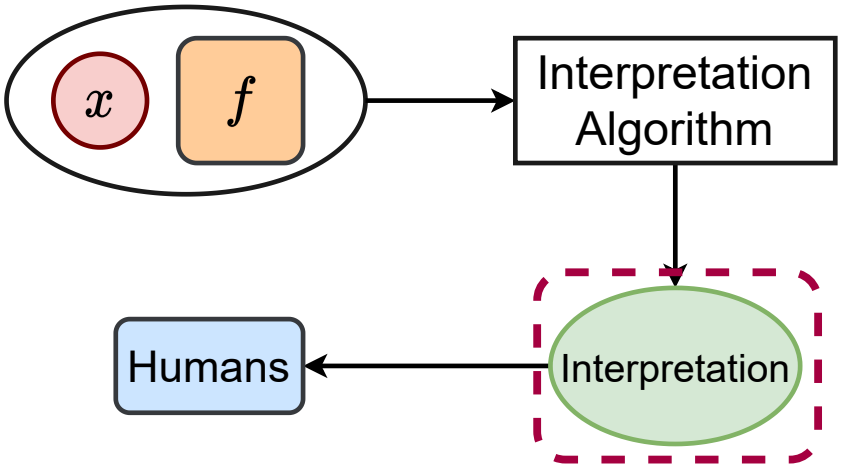


Figure 2.1: A high-level flow for a typical pipeline for an interpretability application

interpretable predictive model. With respect to Fig. 2.1, this corresponds to the following:

1. *Post-hoc interpretation*: f is a pre-trained and fixed predictive model. The goal is to interpret the decisions of f through an additional approach, either for a specific sample x or as a whole on the dataset $\mathcal{S}$.

2. *By-design interpretation*: We are required to design an f and train it on the given dataset such that $f(x)$ is optimized for both prediction performance and interpretability on the given task.

Both the problems offer differing challenges and unique pathways to address a learning problem with performance and interpretability as desiderata. Post-hoc interpretability assumes the "status quo", that is, a predictive model maximized for performance, and then searches for novel ways to best interpret its decisions by devising a separate method. Another way to look at is that it relies on traditional learning methods to independently maximize performance. Given this initial point it aims to maximize on the interpretability axis through a separate approach. By-design interpretation on the other hand can be seen in some sense to jointly optimize on both axes in a single learning model. It targets for inherent structures in the prediction model itself that result in higher level of interpretability. The bigger challenge then becomes to simultaneously ensure adequate level of performance and interpretability.

Among the two problem types, a bigger chunk of research has focused on post-hoc interpretation. While interpretability as a research topic has been around for quite a while (Shortliffe and Buchanan, 1975; Clancey, 1983), it's popularity in the past decade to a good extent can be attributed to the work by Simonyan et al. (2013) on gradient based class visualization in convolutional networks. This is not to say there were no efforts in fields just preceding this work (Montavon et al., 2013), but in an unofficial sense it popularized research on post-hoc interpretation as well as gradient

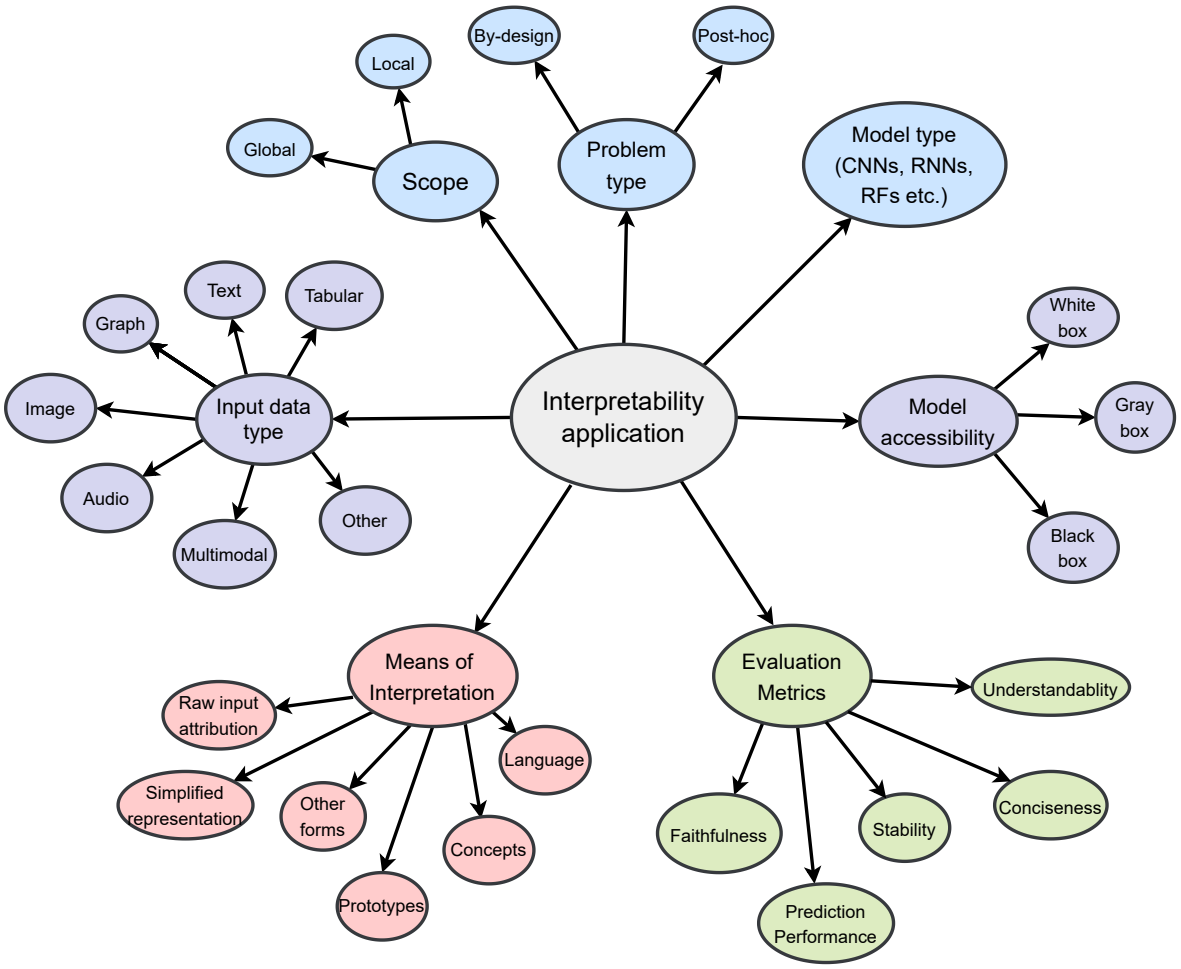


Figure 2.2: Various factors determining context of an interpretability application. The factors in blue are related to problem definition, while those in red are related to characteristics of the algorithm/method being used. Factors in purple can be considered part of both these aspects. Factors in green are different axes of interpretation evaluation.

based saliency maps. This work was further followed by multiple popular saliency map based visualization techniques (Springenberg et al., 2014; Selvaraju et al., 2017; Smilkov et al., 2017; Bach et al., 2015; Sundararajan et al., 2017) applied on CNNs trained for computer vision tasks. A second wave of research in post-hoc interpretation followed after the proposal of LIME (Ribeiro et al., 2016) algorithm. LIME offered a framework for local post-hoc approximation of complex black box models for single samples via significantly more interpretable models (decision trees, linear models). The simple and general structure of LIME has resulted in large class of works improving, formalizing, and applying it in various domains (Lundberg and Lee, 2017; Mishra et al., 2017; Lakkaraju et al., 2019). Beyond the above mentioned works, there have been plenty other proposed approaches and newly tackled domains that will be covered in greater detail in rest of the chapter.

It's hard to pinpoint one single work to popularized working on by-design interpretability in recent years. A bunch of different works in relatively close time-period (Al-Shedivat et al., 2017; Li et al., 2018; Alvarez-Melis and Jaakkola, 2018a; Yoon et al., 2018) proposed their own methodologies of interpretable neural networks which exhibit high performance. All these approaches modify the architecture of the model to achieve interpretability. A different set of approaches pushed ahead in the direction of modifying the objective function to incorporate interpretability in the model (Zhang et al., 2018b; Lee et al., 2019). Research for this problem has rapidly picked up pace since these initial set of works.

Research for by-design interpretability has also gained traction partly because of criticisms leveled at various post-hoc approaches. Kindermans et al. (2019) showed that many saliency map methods failed to generate consistent attributions upon a simple transformation of shift on the input. Alvarez-Melis and Jaakkola (2018b) raise questions about robustness of various saliency map methods. The faithfulness of post-hoc approaches has also been called into question (Rudin, 2019). This begs the question of if there is value in only solving by-design interpretation and no need for post-hoc approaches. A few works dive deeper into comparing the two approaches. Rudin (2019) is a popular work in this regard. It strongly argues in favour of refraining from post-hoc approaches and opting for by-design approaches for high-stakes or critical decision making applications. These applications can be viewed as cases where knowing insights and reasoning behind a decision is essential.

Nevertheless, from a practical standpoint both problems hold independent research value. Real-world scenarios of utilizing interpretability of networks can occur under variety of constraints and demands regarding deployment, level of interpretability and performance. The introduction of GDPR (Voigt and Von dem Bussche, 2017) has also lead to a large number of applications with interpretability requirements of varying degree. For instance, a company might consider performance of a model to be absolutely essential for service and insights from interpretable algorithms as secondary mechanism for transparency with users. In this case they are lot more likely to lean towards a post-hoc approach to interpretability.

Beyond the two broad applications discussed here, there have been other proposals for intermediary problem types. Plumb et al. (2020) consider a problem of regularizing a pre-trained black-box model for improved interpretability without harming the performance. Sarkar et al. (2022) propose the problem of "ante-hoc" explainability wherein they start from a pre-trained classification network and aim to learn a by-design interpretable model from it. Both these problems cannot be clearly categorized either as post-hoc or by-design interpretable system. Both take as initial input a pre-trained model with good performance but intend to modify it. However, the first still intends to use the same architecture for final prediction, while the latter intends to use an architecturally modified model for prediction. In this sense these problems could be seen as points on a spectrum as shown in fig. 2.3.

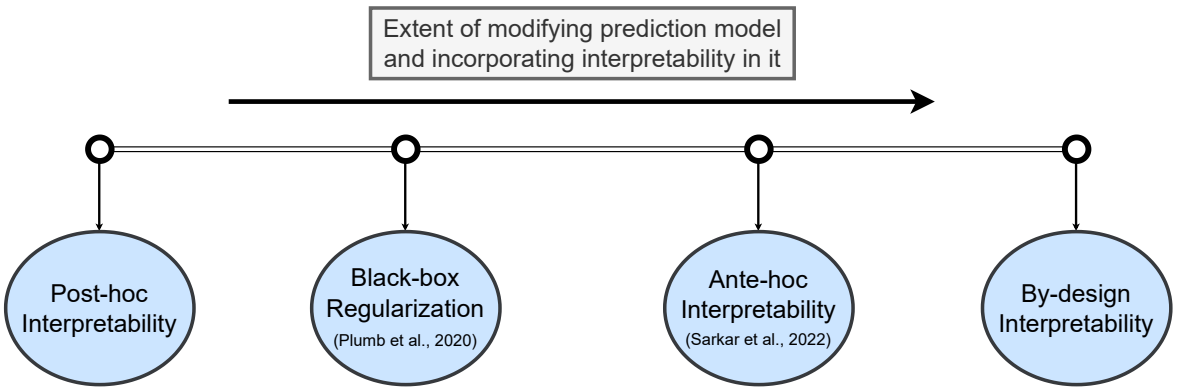


Figure 2.3: Spectrum of various types of interpretability problems according to the extent they modify a typical prediction model and incorporate interpretability in their final prediction model. In case of post-hoc interpretation, the final prediction model is unchanged. The other extreme, by-design interpretation either completely trains a novel architecture or trains with a novel objective function for interpretability.

2.2.2 Scope of interpretability

The issue of gaining insights about a model can be posed at two different levels of granularity. The first can be to understand the model's behaviour for a single sample (or *locally*). This attends to the question of what features in a specific sample led to the models decision. The other level is to understand the model as "a whole" (or *globally*). This generally considers the question of what set of features does the model primarily rely on across the dataset for its output. This aspect is often referred to as scope of interpretability. We will from now on refer to interpretations with local scope as *local interpretation* and with global scope as *global interpretation*.

The vast majority of approaches until 2018 addressed local post-hoc interpretability problem. There are two distinct families most of them fell into. The first is the family of saliency map approaches which rely on some form of gradient or relevance backpropagation to generate input attribution map as interpretation (Montavon et al., 2018). The second is the family of *perturbation* based approaches. These methods typically treat the underlying model as a black box and fit a simpler model over multiple perturbed versions of input-output samples (Ribeiro et al., 2016; Lundberg and Lee, 2017; Lakkaraju et al., 2019, 2020).

Compared to local interpretations there are relatively much fewer methods that generate global interpretations (or both). One way to partly explain this is to consider the potentially higher practical value in addressing local interpretability. However this does not provide the complete picture. It should be noted that local interpretation approaches can potentially be extended to produce global interpretations (for example LIME Ribeiro et al. (2016)). However, these extensions are generally only valid for tabular data. This raises a question about the difficulty in extending for other modalities. The bottleneck for extending on other modalities typically comes from the

feature representation used for interpretation. By this, we refer to the representation over which the importance values are generated. In a way this forms the "language of interpretation" through which humans gain the understanding. It is generally hard to extend local approaches for global interpretation because they often rely on representations that have a locally derived understanding, and do not generalize outside the neighborhood of a sample. For example, saliency maps in case of images generated importances over the space of pixels. High importance of certain group of pixels can be considered insightful or interpretable if it corresponds to some high level object in the image, such as a face. However, the understanding of a face associated with it, only exists in context of the given sample. For a different image same set of pixels might hold a completely different meaning. Conversely, for tabular data, input features hold a high-level meaning that is general across the whole space. The feature for area of house might take different values for different samples but it always retains its high level meaning.

In this regard, the proposals of prototype based approaches Li et al. (2018); Zinemanas et al. (2021) and concept based approaches (Kim et al., 2017; Alvarez-Melis and Jaakkola, 2018a; Ghorbani et al., 2019) have been innovative. They aim to define or learn representations of interpretation that have an underlying semantic structure extending beyond any specific input instance. This allows the possibility to generate both local and global interpretations. At this point we do not go much further into the detail of these methods. We elaborate on them in discussion about different methods and their representation of interpretation in section 2.3.

2.2.3 Input data type

For any ML practitioner, what type of data they have to operate on, is a fundamental and influential factor. The same follows for an interpretability problem too. It is thus sensible to partition the methods in literature to assess development for various data modalities. Different modalities offer different challenges to work with. In certain cases they carry unique relationships for an interpretation algorithm to exploit. On the other hand they can constrain an algorithm in multiple ways. These factors help create interesting scenarios and novel methods to address them. The two modalities prominent in this thesis, Image and Audio, will be covered later in section 2.4 since we also want a relatively more detailed view about the prior methods proposed for them.

Tabular – Similar to how many popular supervised learning algorithms were first benchmarked on tabular datasets , interpretability algorithms have sketched a similar path. It as arguably the modality with most applicable number of methods. While the relative computational ease in terms of memory and storage is an important driving factor for the same, there are two crucial reasons why this has received such heavy attention from interpretability researchers. Firstly, the features already have a human understandable meaning. This makes quantitatively evaluation and qualitative demonstration of interpretability considerably more comfortable. Secondly, real-world applications of tabular data are immense. Consequently, many critical

ML applications where interpretability might assume utmost priority, come under its wing, for instance concerning financial or healthcare data.

Various saliency map methods (Bach et al., 2015; Sundararajan et al., 2017; Smilkov et al., 2017), perturbation approaches (Ribeiro et al., 2016; Lundberg and Lee, 2017; Lakkaraju et al., 2020, 2019), by-design interpretable networks (Radenovic et al., 2022; Agarwal et al., 2020; Al-Shedivat et al., 2017; Alvarez-Melis and Jaakkola, 2018a; Yoon et al., 2018) have all been applied for tabular datasets.

Text – Apart from tabular and image data, interpretability of natural language processing models has received the most attention. This is partly because of massive networks exhibiting human-like capabilities to process and produce text (Kenton and Toutanova, 2019). However, from the point of view of interpretation text data offers an interesting domain to test algorithms. Words are often treated as tokens over which importances are generated. The effortlessly provides a manner to deliver understandable interpretations. Akin to tabular data, research methods for all types of problems discussed previously have been proposed, ranging from local post-hoc approaches through saliency maps (Bahdanau et al., 2014; Mullenbach et al., 2018; Bang et al., 2021; Ross et al., 2017) or perturbation approaches (Alvarez-Melis and Jaakkola, 2017) to by-design interpretable approaches with local (Al-Shedivat et al., 2017; Croce et al., 2019) or global interpretability . These methods most commonly address text classification or question-answering tasks.

Graphs Research on interpretations of models processing graph data has significantly increased in the past 3-4 years. Work in this domain has evolved in three different directions (Li et al., 2022). There have been a group of methods extending previous post-hoc interpretation techniques to graphs, including LIME (Huang et al., 2022), CAM (Pope et al., 2019) and LRP (Schnake et al., 2021; Cho et al., 2020). A newer class of methods, specifically targeting graph neural networks (GNN) or graph convolutional networks (GCNN) aim to exploit the architecture of the network for improved interpretations. That include a subset of methods offering interpretations through subgraphs (Ying et al., 2019; Vu and Thai, 2020; Lin et al., 2020; Yuan et al., 2021), and a second subset considering complete graph structure (Luo et al., 2020; Yuan et al., 2020).

Video/multimodal Interpretability for models processing multiple modalities is still a relatively under-explored, but interesting problem domain. Current works are predominantly for models processing image and text for visual question answering (VQA) tasks (Park et al., 2018; Strout et al., 2019; Selvaraju et al., 2020). There have been some efforts in interpretability of video processing networks. Kanehira et al. (2019) for instance target video classification task and Tian et al. (2019) propose interpretable audio-visual captioning method.

Other data types – Many chemical or biological data processing tasks arise as high-stakes applications for ML models. Thus, interpretability can potentially be a strong requirement in such cases. There are rich fields developing interpretability methods for bioinformatics applications with gene expresstion data (Hanczar et al., 2020;

Bourgeais et al., 2021; Xing et al., 2021), processing EEG or ECG data (Ma and Zhang, 2019), molecular information based datasets (Lee et al., 2019) etc. To some degree, depending upon the modality, these methods are influenced by the works developed as generic interpretability methods. Nevertheless, each of these applications carry their unique challenges which inevitably results in novel modifications.

2.2.4 Other factors

Having discussed the most relevant contextual factors describing a problem, we now complete the discussion about the factors characterizing an interpretability application by briefly covering two sets of factors below:

Model type & information

A practical aspect that differentiates many interpretation applications is information about the model being interpreted. It itself comprises of primarily two types of information. The first is the type of model being interpreted. This could vary between a random forest (Bénard et al., 2021), a recurrent neural network (Van Luong et al., 2021), an artificial neural network (Boz, 2002) etc. The second type of information, relevant only for post-hoc interpretation approaches, is the extent of model information (for eg. weights of a neural network) available to the interpretation algorithm. A sizeable fraction of the proposed algorithms consider the model as a black-box and thus they only have access to input and output of a model (Ribeiro et al., 2016; Chen et al., 2018). The other extreme consists of gradient based approaches which have complete information about internal parameters (needed for backpropagation). A number of approaches fall into neither of these categories and treat the underlying model as a "gray-box", with access to selected parts of the model's parameters/representations. Most common examples of this are methods that access certain hidden layer of a neural network (Selvaraju et al., 2017; Kim et al., 2017; Schulz et al., 2019).

Human intervention

A second set of practical aspect which offers a unique flavour to any application is human intervention. This includes the extent and point of intervention. Vast majority of methods simply provide their outputs to be analyzed by humans and thus involve no human intervention in their outputs. Concept bottleneck models (Koh et al., 2020) explore the idea of studying their model while allowing human intervention to modify the models prediction for concept labels. One can also modify the point where signals from human are used. Arous et al. (2021) for instance incorporate human rationales to improve interpretability and performance of text classification. The research topic of human-in-the-loop learning (Wu et al., 2022) can be explored further to dive deeper in the nuances of this factor.

2.3 Algorithms and means of interpretation

Until now we primarily viewed the literature from the categories of factors contextualizing an interpretability learning problem. Now we turn our attention towards factors contextualizing an interpretability generating algorithm, that is, what factors affect and differentiate various methods to generate interpretation.

As of yet, there is no universally agreed generic mathematical definition for an interpretation. In our view, any interpretability method (local) consists of two integral components. First, a function $\Omega : \mathcal{X} \times \mathcal{F} \rightarrow \mathcal{Z}^d$, that computes a data representation $\Omega(x, f)$ used for interpretation. The representation consists of d individual elements, each in a space $\mathcal{Z}$. These individual elements form the units of interpretation that a human should be able to understand. We refer to this as the means of interpretation or representation for interpretation. The second key component is a relevance/importance function $r(x, f) \in \mathbb{R}^d$ which computes importances of different elements of $\Omega(x, f)$ for the decision $f(x)$. The idea behind Ω is a more general version of "interpretable data representation" as discussed in LIME (Ribeiro et al., 2016). Herein, we also allow the data representation to be computed using the classifying function f, which is an occurrence in multiple methods (Li et al., 2018; Alvarez-Melis and Jaakkola, 2018a; Ghorbani et al., 2019).

From a methodological perspective, most algorithms differ in how they design Ω (which includes defining $\mathcal{Z}$), or how they propose to generate $r(x, f)$. It is important to recognize the wildly varying possibilities of designing Ω or r. For example for backpropagation based saliency map approaches, $\Omega(x, f) = x$. For images, this is a low-level representation with $\mathcal{Z}$ being space of RGB or grayscale pixel values. r implements a procedure of modified backpropagation to determine importance of individual pixel for decision $f(x)$. On the other hand for prototype interpretation methods, $\mathcal{Z} = \mathbb{R}$, $\Omega(x, f)$ represents distance of x to d prototypes learnt by f and $r(x, f)$ can simply be a fixed weight matrix used to make final predictions, but also indicating which prototypes are important for which output logits. The possibilities are so wide that one can even represent human language interpretations about a model decision wherein one considers $\mathcal{Z}$ as the space of words, $\Omega(x, f)$ as a fixed language dictionary independent of x or f, and $r(x, f)$ to determine which words to select (along with their ordering), to generate a meaningful sentence. Note that the above discussion does not in anyway convey information about how understandable the interpretations are, which is highly subjective.

We discuss below the common possibilities of $\Omega(x, f)$ that have been explored in prior literature along with the most representative interpretation algorithms for each possibility:

2.3.1 Raw input attribution

The single most frequent way of generating interpretations is based on estimating raw input feature importance, i.e. $\Omega(x, f) = x$. Part of the reason of its frequency is its applicability, as it's the most obvious choice of representation to begin with. The common theme behind raw input attribution approaches is that they generate importances over the space of original input domain. As discussed previously, back-propagation based saliency map approaches are most representative of this group.

More specifically, given a classifier function f from a space of predictive models $\mathcal{F}$, $f : \mathcal{X} \to \mathcal{Y}$ with $\mathcal{X} = \mathbb{R}^d, \mathcal{Y} = \mathbb{R}^+$, a saliency map S can be described as $S : \mathbb{R}^d \times \mathcal{F} \to \mathbb{R}^d$, that is, $S(x, f)$ takes as a input a sample x and function f and computes importance values over features of x. Note that the notion of relevance function $r(x, f)$ is indeed the saliency map $S(x, f)$ in case of raw input attribution. For multi-class classification, $f(x)$ can denote output probability of a single class. Different methods have proposed different ways of computing $S(x, f)$, some of which we'll mention below. However, the underlying idea behind them from the point of view of means of interpretation is that the way a human visualizes/understands an input sample x, they can use the same means to visualize/understand $S(x, f)$ and identify which parts x were deemed relevant for the decision by the saliency map. If $\mathcal{X}$ is a set of images, then the saliency map highlights what regions of an image were relevant to the decision. Examples of many saliency maps are illustrated in Fig. 2.4

Simonyan et al. (2013) proposed using gradient of classifier output w.r.t input as method to compute S, that is, $S(x, f) = \nabla_x f(x)$. Shrikumar et al. (2016) discusses use of positive part of $x \odot \nabla_x f(x)$ as the saliency map, which is the element-wise product of input and gradient. GuidedBackProp (Springenberg et al., 2014) proposes use of a modified gradient-backpropagation approach, of backpropagating only through positively activated neurons as procedure to compute saliency map. LRP (Bach et al., 2015) instead of relying on gradients propose their own methodology of propagating relevance across layers based on a layer-wise conservation principle. Another popular approach of generating saliency maps, different from the previous ones is that of GradCAM (Selvaraju et al., 2017), wherein they first compute weighted average of convolutional maps from the last layer according to magnitude of their gradient. This map is upscaled to input image size and combined with GuidedBackProp output to generate final saliency map.

Except for tabular data, where some interpretability methods can also provide global interpretations, input attribution is almost exclusively used for generating local interpretations. The major advantage in using this means is its wide applicability. However a potential criticism for this means arises from the fact that raw input features need not be highly meaningful for human reasoning. This is typically the case for modalities like image, audio, graphs etc. For eg. raw time domain samples or time-frequency bins of spectrograms in case of audio is a very odd basis for humans to understand a decision as. Similar is the case of using raw pixels to understand decisions for images. For saliency maps this criticism can also be encapsulated in a

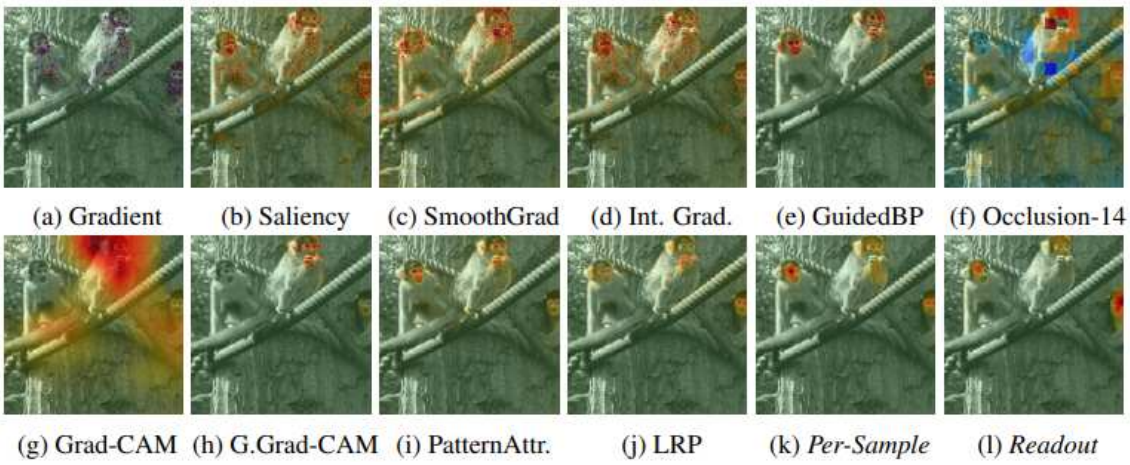


Figure 2.4: Examples of saliency maps from various methods. Image from Schulz et al. (2019)

different way. Saliency maps can identify **where** relevant features for a model are (in case of images) but poor at indicating **what** those features are and how they are being used for the decision (Thomas et al., 2022). A partial fix for this is through the development of simplified representations which are covered next.

2.3.2 Simplified representations

These representations generally appear in the context of methods generating local post-hoc interpretations for non-tabular input modalities. The motivation behind using them is that we want to represent the input in a "simplified" manner, that is suitable for interpretation. This representation is utilized by the interpretation algorithm, which them computes the feature importances over this more "human-friendly" representation. The most popular methods employing this means of interpretation is LIME (Ribeiro et al., 2016) and SHAP (Lundberg and Lee, 2017). We cover the LIME algorithm below along with specific instances of the representations in case of text, image and audio:

Given a classifier function as before $f : \mathcal{X} \rightarrow \mathbb{R}^+, \mathcal{X} = \mathbb{R}^d$, and a sample x for which we wish to interpret $f(x)$, LIME interprets via a model $g \in G, g : \mathcal{X}^{'} \rightarrow \mathbb{R}$, where G is a class of potentially interpretable models, for example linear models or decision trees operating on simplified representation of the input. They define $\Omega(g)$ as a measure of complexity of g. In case of linear g, $\Omega(g)$ is he number of non-zero weights while in case of decision trees $\Omega(g)$ is the depth of the tree. Additionally, they define unfaithfulness between f and g with the distance between their outputs in the local neighborhood of x, by $\mathcal{L}(f, g, \pi_x)$, where $\pi_x(z)$ denotes proximity of instance z to x and helps mark neighborhood of x. To then generate local interpretation for $f(x)$ they solve the following optimization problem which tries to balance between fidelity of g to f and its complexity:

$$g_x = \underset{g \in G}{\operatorname{argmin}} \mathcal{L}(f, g, \pi_x) + \Omega(g)$$

The key remaining detail is about the simplified representation domain $\mathcal{X}^{'}$ and how it helps to provide interpretation. LIME proposes designs of $\mathcal{X}^{'}$ for text and image modalities. For text classification, they propose to represent input text by a binary vector indicating the presence or absence of a word. For image classification, they represent an image by a binary vector indicating the "presence" or "absence" of a super-pixel (Achanta et al., 2012). The domain $\mathcal{X}^{'}$ is then uniquely defined for a given x by the binary vector $\{0,1\}^{d'}$ wherein each vector contains information about presence or absence of a word/super-pixel. The proximity measure $\pi_x(z)$ is also defined in the domain of simplified representation, for eg. how many word/superpixels are removed in z compared to x. Computationally, LIME solves the optimization problem on a set of samples obtained by perturbing x multiple times in simplified representation domain and modelling the variation in output of f over these samples through a simpler function g_x. The interpretation is then generated according to the structure of g_x. For example, for linear models coefficients of g_x denote the importance of a superpixel or word. In this case, the coefficients and the corresponding features they attend to serve as the final interpretation. Note that $\Omega(x, f)$ in this case is based on $\mathcal{X}^{'}$. It is only locally meaningful for each x denoting the set of superpixels/words composing x.

Interestingly, one can easily define the simplified representation to align with the original input domain $\mathbb{R}^d$ as $\mathcal{X}' = \{0,1\}^d$, where a binary vector would denote presence/absence of an individual feature. This is indeed the case for when LIME is applied to tabular data. However this quickly becomes a poorly structured problem for input spaces of high-dimensionality. The total possible perturbations is 2^d i.e., exponential in d. For modalities like images or audio d can easily run into thousands or even millions. Even if one need not cover the complete neighborhood, the number of required perturbations to cover reasonable amount of this space grows high. This in turn worsens the fidelity-complexity trade-off of g since with large number of samples, maintaining high fidelity requires much more complex linear model. A simplified representation thus not only has a interpretability motivation, but also a computational motivation in case of perturbation-based methods. The number of superpixels are typically considerably less than number of pixels making these algorithms lot more practical to use.

Another commonly used simplified representation for images is that of dividing them in rectangular patches. This is utilized by some information bottleneck based approaches for local post-hoc interpretations, L2X (Chen et al., 2018) and VIBI (Bang et al., 2021). Their algorithms select a predefined number of patches and their interpreter approximates the classifier decision using only the selected patches as input. A very similar simplified representation is also utilized by an extension of LIME algorithm for audio signals, termed Sound-LIME (SLIME) (Mishra et al., 2017). Their algorithm divides an input spectrogram into non-overlapping time-frequency patches and then applies the LIME algorithm to extract a predefined number of most

important patches for singing voice detection task. It is worth observing that while choice of this representation is relevant for the interpretation algorithm, the interpretation itself and the process of understanding it is almost the same as for raw input attribution.

They also come with their own set of limitations. From a practical perspective the key limitation is that choosing a reasonable set of hyperparameters can be highly sample-dependent, requiring frequent human supervision. Moreover, these methods are generally not capable of generating global interpretations, since a simplified representation cannot be generalized across multiple samples (except for tabular data).

2.3.3 Prototypes

Along with concepts, this is one of the most recent means of interpretation proposed. They have primarily been applied to address by-design interpretation (Li et al., 2018; Angelov and Soares, 2020), through a neural network design based on prototype classification techniques (Bien and Tibshirani, 2011). The design has also been adapted for other types of modalities, including audio (Zinemanas et al., 2021; Loiseau et al., 2022). All of them can be loosely grouped under the umbrella term'prototypical networks'. The idea behind prototype classification is to classify a sample based on its proximity to prototype observations from a dataset. The modification prototypical networks make is to represent prototypes as points in a latent space. The distance of any input is computed with the prototypes in the latent space and the decision is made by processing the distance vector, typically through a linear classifier. The distance vector in this case plays the role of $\Omega(x, f)$ with an underlying structure of learnt prototypes governing it. Chen et al. (2019) slightly differ in their modelling of prototypes for image classification. They instead model patches in an image as prototypes. For a new input they make a decision based on the distance *matrix* of patches in the input image w.r.t all prototypes. A local interpretation for any sample consists of prototypes that were determined closest by the network. A user is then typically able to visually identify reasons of closeness of the sample to the prototypes. While these approaches generally do not explicitly consider problem of global interpretation, the set of prototypes and their weight matrix leading to final decision (assuming a linear classifier on top of distance vector) can be considered as a form of global interpretation. For completeness, we briefly cover the system design of Li et al. (2018), and adaptations for their audio counterparts.

Li et al. (2018) consider their model as composed of three networks, an encoder $f : \mathcal{X} \rightarrow \mathbb{R}^q, \mathcal{X} = \mathbb{R}^d$, a decoder $d : \mathbb{R}^q \rightarrow \mathcal{X}$ and a prototype classification network $h : \mathbb{R}^q \rightarrow \mathcal{Y}, \mathcal{Y} = \mathbb{R}^K$. The networks f, d learn m prototypes $p_1, p_2, ..., p_m \in \mathbb{R}^q$. The decoder is essential to be able to visualize all the prototypes as samples in input space. The final classification network is $h \circ f$. The network h is divided into two primary parts: a prototype layer $p : \mathbb{R}^q \rightarrow \mathbb{R}^m$ computes squared ℓ_2-distance of an embedding $z = f(x)$ to all the prototypes as $p(z) = [\|z - p_1\|_2^2, \|z - p_2\|_2^2, ..., \|z - p_m\|_2^2]$. The second part of h is a linear layer with softmax activation. Given a dataset $D = \{(x_i, y_i)\}_{i=1}^n$, the following loss function is proposed to train the networks.

$$\mathcal{L}(f,g,h,D) = CE(h \circ f, D) + \lambda R(g \circ f, D) + \lambda_1 R_1(p_1,...,p_m,D) + \lambda_2 R_2(p_1,...,p_m,D)$$

$$R_1(p_1,...,p_m,D) = \frac{1}{m}\sum_{j=1}^{m} \min_{i\in[1,n]} \|p_j - f(x_i)\|_2^2$$

$$R_2(p_1,...,p_m,D) = \frac{1}{n}\sum_{i=1}^{n} \min_{j\in[1,m]} \|f(x_i) - p_j\|_2^2$$

The term $CE(h \circ f, D)$ is the cross-entropy loss on the classification output. The second term $R(g \circ f, D)$ is the MSE reconstruction loss to train the encoder and decoder to learn to map dataset points to latent space and back. The remaining two losses R_1, R_2 are proposed prototype losses. R_1 encourages each prototype to remain close to some training sample. This helps in visualization of a prototype as it is expected to be close to some training sample. The loss R_2 encourages each sample to be close to some prototype. This helps the set of prototypes to cover the training space efficiently.

This design of prototypical networks has been modified in various ways. A recent work ProtoVAE (Gautam et al., 2022) proposes to replace the autoencoder with a variational autoencoder and introduces a orthonormality constraint to learn more diverse set of prototypes. Zinemanas et al. (2021) proposed a version of prototypical networks suitable for audio, abbreviated as APNet (Audio Prototypical Networks). They propose a different strategy to compute distances between prototypes, much more suited for mel-spectrogram like embeddings than ℓ_2 distance. Work by Loiseau et al. (2022) further improved upon APNet by additionally learning and predicting parameters that transform audio-specific properties for each prototype, essentially allowing them to learn controllable prototypes.

The biggest advantage of prototypical networks is their adaptability for diverse use-cases. They can be applied for various modalities, can be used for local and global interpretations and have a straightforward mechanism for interpretation. However they have their own drawbacks as well. They are used only for by-design interpretation and not post-hoc interpretation. They are highly dependent on good reconstruction for understandability of prototypes. The flexibility in designing the original or latent space for prototypes can sometimes address this issue. Nevertheless, for its most common usage, high-quality reconstruction is essential. Lastly, as highlighted previously, the user is tasked to visually identify reasons a prototype is considered close to the given sample for the network. In this regard, prototypical networks can improve upon interpretability of their underlying decision process. Hoffmann et al. (2021) highlight this issue in a realistic setting of compression artefacts in input.

2.3.4 Concepts

A relatively more recent means of interpretation, mainly arisen in the context of deep neural networks whose hidden layers have been empirically shown to capture high-level features. It is based on idea of using representing high-level abstract concepts in

a network. They offer significant potential upsides in terms of understandability since humans also reason and communicate through such abstract concepts. Moreover, they aim to interpret a decision at finer scale with more detailed information. For example, given an image of an ant, instead of presenting the most salient input region as interpretation which is the case for most methods, concept-based methods attempt to quantify the individual impact of finer details constituting the input such as 'blobs' and 'tentacles/thin-legs'. Beyond these upsides, these methods can generally also be adapted to provide global interpretation of the model. The key challenge in developing such methods is learning or visualizing them reliably in an understandable manner. Research on this means of interpretation can be broadly divided into three directions:

Concept activation vectors

Kim et al. (2017) propose the idea of concept activation vectors (CAV) for interpretation of deep neural networks. To ease understanding, it is useful to break the TCAV proposal in three major parts: (a) Representation of concept, (b) translating representation to hidden layers and (c) generating relevance of concept for a class.

TCAV proposes using a set of positive and a set of negative examples to represent a high-level human-understandable concept. For example, in case of object classification of images, the concept of "stripes" can be represented by a bunch of examples from striped objects (zebra, tiger, food etc.) which form the set of positive examples and a set of random examples from other classes with non-striped objects. These examples are human annotated/selected. This gives a flexible mechanism to user to represent any high level concept they like. The second step consists of translating this representation of a concept as sets of positive and negative examples to representation in terms of hidden layers of network being interpreted. Specifically, when dealing with a deep neural network classifier $f : \mathcal{X} \to \mathbb{R}^K, \mathcal{X} = \mathbb{R}^n$, K is the number of classes, they select a hidden layer l, and the f can be written as $h_l \circ f_l$, where $f_l : \mathbb{R}^n \to \mathbb{R}^m$ is output of hidden layer l and $h_l : \mathbb{R}^m \to \mathbb{R}^K$ denotes remaining layers of network f, until the output layer. Representing a concept C, denoted by set of positive examples P_C and negative examples N, in terms of hidden layer l consists of first collecting layer activations for both sets of examples $\{f_l(x), x \in P_C\}$ and $\{f_l(x), x \in N\}$, and training a linear classifier in the activation space to classify the sets of positive and negative examples. This classifier, represented by the normal vector of hyperplane $v_C \in \mathbb{R}^m$ is termed as concept activation vector. The final piece of the method, to compute relevance of concept for a class k consists of computing directional derivative of the output neuron, denoted as $h_{l,k}$ w.r.t concept activation vector v_C, that is, the quantity $S_{C,l,k} = \nabla h_{l,k}(f_l(x))^T v_C$. This quantity is computed and fraction of samples from class k with positive directional derivatives is used as the estimate for importance of any concept C in prediction for class k.

A key limitation of the TCAV approach is the dependence over human supervision to define a concept. This limitation was addressed by the work of Ghorbani et al. (2019) who proposed an approach for automatic concept extraction (ACE). The core of their

approach consists of first generating various superpixels at multiple scales and for many images of a class. A representation of each superpixel is than extracted through the use of their network's hidden layers. These representations undergo clustering wherein perceptually similar superpixels tend to be close to each other. Each cluster is finally used to define a concept. While ACE is able to remove dependence on human supervision, it faces its own sets of limitations. Their algorithm is limited to networks that can meaningfully cluster superpixels. This is not an obvious behaviour one can expect since the scaled superpixels are not representative of input distribution. While this has empirically been ascertained for large neural networks trained on ImageNet but nonetheless, it is difficult to generalize to other settings.

Self-explaining neural networks (SENN)

Alvarez-Melis and Jaakkola (2018a) proposed a by-design interpretable network based on generalized linear models. Their approach employs two sub-networks $\theta : \mathcal{X} \rightarrow \mathbb{R}^{C \times J}$ and $h : \mathcal{X} \rightarrow \mathbb{R}^J$, both of which operate on input. The final prediction is computed as $f : \mathcal{X} \rightarrow \mathbb{R}^C$, $f(x) = \theta(x)^T h(x)$. The network θ learns to generate coefficients of the model $\theta(x)$. The network h learns a dictionary of concepts and thus $h(x)$ fulfills the role of $\Omega(x, f)$. Note that the representation for interpretation is a part of computation of $f(x)$. If the input is a tabular data then h is simply fixed as the identity function $h(x) = x$ as the features already carry high-level meaning. In case of other modalities, for eg. images, the dictionary of concepts is learnt in an 'unsupervised' manner. It is important to make it clear at this point that when referring to learning concepts in an 'unsupervised' fashion, we refer to unavailability of any additional information or supervision regarding learning of concepts. It does not refer to the learning problem for our final prediction from this model. The learning of this dictionary is carried by imposing constraints on it, implemented as loss functions. Each element of this dictionary $h(x)_i \in \mathbb{R}$ encodes a high-level concept that needs to be understandable for the user. We will discuss below the loss function to train the complete model. While the work explores deeper the direction of learning the coefficients stably, our focus is more on the learning and understanding of dictionary of concepts. The complete training loss writes as follows:

$$\begin{aligned} \mathcal{L}(f, x, y) &= \mathcal{L}_y(f(x), y) + \lambda \mathcal{L}_\theta(f(x)) + \zeta \mathcal{L}_h(x, \hat{x}) \\ \mathcal{L}_\theta(f(x)) &= \|\nabla_x f(x) - \theta(x)^T J_x^h(x)\|_2^2 \\ \mathcal{L}_h(x, \hat{x}) &= \|x - \hat{x}\|_2^2 + \|h(x)\|_1 \end{aligned}$$

The loss function consists of three terms. The first term $\mathcal{L}_y(f(x), y)$ is the classification loss on output of f. The second term $\mathcal{L}_\theta$ is a proxy term to encourage stability of coefficients with respect to features, that is to encourage the property that the coefficients change little in the vicinity of any point in feature space. It requires computing the jacobian of concepts w.r.t input features. The final term $\mathcal{L}_h$ is used to learn the dictionary of concepts. It is composed of two terms. The first term trains the dictionary as embedding of an autoencoder, via the network $h_{dec} : \mathbb{R}^J \rightarrow \mathcal{X}, \hat{x} = h_{dec}(h(x))$.

This constraint is proposed to promote fidelity to input and preserving the relevant information about it in $h(x)$. The second term promotes sparsity of $h(x)$. The reasoning behind this term is that any input sample should be representable through few non-overlapping concepts. To understand an individual atom of the dictionary $h(x)_i$, the authors propose to visualize the set of training samples maximally activating $h(x)_i$. By understanding the common patterns among the maximally activating training samples, the user derives understanding about the underlying concept.

Concept-bottleneck models

Contrary to previous two directions which involved defining/learning representation of concepts as part of their research agenda, concept bottleneck models Koh et al. (2020) assume availability of known dictionary of concepts. This also makes learning of concepts relatively lot more straightforward. This research direction instead studies the utility of concept-based interpretations with human intervention.

It is worth noting that concept representations in examples with known dictionaries is similar to representations for unsupervised learning proposed in SENN. In a way this highlights that using representation of set of concepts in images as real-valued activation vector is grounded in existing examples. Work on "ante-hoc explainability" Sarkar et al. (2022) follows a similar theme where they tackle cases of both known dictionary of concepts and unsupervised learning of concepts via the same structure of representation.

The concept bottleneck models have gained recent popularity with multiple research works analyzing its ideas or building from it Margeloiu et al. (2021); Sawada and Nakamura (2022); Yuksekgonul et al. (2022). There are some other notable methods that too apply use of known dictionary of concepts Kazhdan et al. (2020). Chen et al. (2020) for instance use it to transform the latent space to align with the concepts.

2.3.5 Natural language

Using 'natural language' as means of interpretation refers to understanding the decision directly through language, the way any human would explain it to anyone else. This is potentially the form of interpretation which is easiest to understand. Moreover, this can also be considered an important part to achieve interactivity of interpretation systems, a highly desirable quality Lakkaraju et al. (2022). However, there is a major challenge pertaining this. Conveying insights about a model in terms of language requires a mathematical object to encode information about the interpretation and a language understanding module that can transform this information to a communicable piece of a language. Interpretations relying on language as means of interpretation have largely been limited to text processing models Abujabal et al. (2017); Rajani et al. (2019); Sydorova et al. (2019) since their design often has basis in language structure (typically English). Tackling this problem in other scenarios is highly challenging. One might consider research on development of vision-language models (for example for image captioning) as important steps of progression in this

direction. However, this further reinforces the complexity of the task as this reflects progression only for visual modality. Research for other modalities in this regard is lacking relative to vision.

However, for vision, there have been some notable attempts in the direction of developing natural language based interpretation methods. Hendricks et al. (2016) is arguably the most popular of these. They train to generate text-based interpretations from visual features extracted from hidden layers of a classifier. The recurrent model generating the text is conditioned on the predicted class label. The model is trained on the CUB-Birds classification dataset with the help of content descriptions for each image Reed et al. (2016).

2.3.6 Other forms of interpretation

Contrary to the categories discussed until now, which at their core proposed different representations for interpretations, there are two other forms of interpretations which are not necessarily tied down to any specific representation Ω but rather have a different method to determining relevance $r(x, f)$. They determine the important features in $\Omega(x, f)$ **without explicitly quantifying** importance of individual elements.

Logical rules: This refers to the use IF–THEN rules to provide interpretations over $\Omega(x, f)$. The key advantage in opting for them is that they offer a definite interpretations, with virtually no subjectivity, requiring only minimal efforts from a human to understand them (provided $\Omega(.,.)$ is interpretable). Methods using rule-based interpretations exist for both post-hoc (Lakkaraju et al., 2019; Moradi and Samwald, 2021) and by-design interpretation (Kusters et al., 2022; Angelov and Soares, 2020; Qiao et al., 2021), offering local or global interpretations. Their most common usage in literature has been with raw-input or simplified input as Ω.

However the major downside in their usage is the strong fidelity/performance – interpretability tradeoff. Without employing use of complex logic, they are limited in performance, specially for high-dimensional modalities such as images, graphs etc. However, complex logical rules severely affect their understandability. Moreover, reasoning over low-level representations of these modalities can even be intractable, for example raw pixels in an image. Thus they are generally applied to tabular and text data. Nevertheless they act as a useful standalone component, and can potentially operate over other means of interpretation to perform reasoning over more complex features (Angelov and Soares, 2020).

Counterfactual interpretations approach the goal of determining importance by asking what would one change in input to change the decision (Wachter et al., 2017). While these could be considered unique means of interpretation in their own right, a more holistic approach would be to view them as a different style of interpretability that can fit with the other means of interpretations (Lang et al., 2021; Kanehira et al., 2019). Similar to various other modern machine learning ideas, counterfactual reasoning also has old roots in psychology (Roese, 1997). While they are typically not used for global interpretations, they have developed a rich body of literature

spanning most data modalities and interpretability applications (Jacob et al., 2022; Guidotti et al., 2019; De Lara et al., 2021).

2.3.7 Properties for interpretation learning

A central theme to development of many interpretability algorithms, is the idea of converting human understanding about behaviour of interpretations to numerical properties that can be enforced or encouraged through loss functions. It can be considered an important differentiating factor for design of different methods. In our coverage of different methods this idea has been implicitly present at many points in the form of training loss functions and the rationale for their incorporation. For eg. LIME (Ribeiro et al., 2016) encouraged two properties as loss functions. One for local fidelity to output and other for reduced complexity of interpreter.

However, even after recognizing this importance, we have refrained from partitioning the literature from this lens because the challenges and properties are typically specific to the means of interpretation itself and not obvious to compare across different methods. For instance, raw input or simplified input methods impose properties purely to learn relevance generating function $r(x, f)$. The representation for interpretation $\Omega(x)$ is pre-determined. On the other hand prototype/concept-based approaches aim to learn both $r(x, f)$ and $\Omega(x, f)$ through the imposed properties. In case of prototypes this leads to two unique loss functions about dataset coverage of the prototypes. This property is absent from any other design of Ω. In general the underlying notion is that different means of interpretation give rise to different desirable properties with differing goals, making a global comparison non-trivial.

As one might expect, not all methods fit this format. For example, for CAV based methods (Kim et al., 2017; Ghorbani et al., 2019), their understanding about behaviour of interpretations directly reflects in their design rather than as an explicitly imposed loss function.

Nevertheless, it is useful to encapsulate this information for better perspective about the literature. We summarize information about various properties enforced on interpretations for some popular methods in Tab. 2.1, some of which have been discussed previously in this section. Note that the listed properties need not be comparable or implemented in the same way across different methods.

2.4 Image and Audio interpretability

Image interpretability methods : Development of state-of-the-art networks for computer vision tasks has lead efforts for interpretability of these models. Apart from tabular data, images have easily the most attention of interpretability research. Numerous methods have been proposed tackling a diverse set of problems. This includes different methods for local or global, post-hoc and by-design interpretations – saliency

Method	Prediction	Output Fidelity	Complexity	Input fidelity	Stability	Missingness	Consistency
LIME-based		✓	✓				
SHAP		✓				✓	✓
Prototype-based	✓			✓			
Information-bottleneck based		✓	✓				
SENN	✓		✓	✓	✓		

Table 2.1: Popular interpretability methods discussed and the properties imposed. Any single property need not be meaningful for all methods nor be implemented in the same way. Post-hoc methods generally enforce fidelity to output. By-design approaches typically have prediction loss imposed on their representation for interpretation.

map based systems (Simonyan et al., 2013; Selvaraju et al., 2017; Al-Shedivat et al., 2017), perturbation based systems (Ribeiro et al., 2016; Lundberg and Lee, 2017; Fong and Vedaldi, 2017), information bottleneck based systems (Chen et al., 2018; Bang et al., 2021; Schulz et al., 2019), prototypical networks (Li et al., 2018; Chen et al., 2019; Gautam et al., 2022), concept-based systems (Kim et al., 2017; Ghorbani et al., 2019; Yeh et al., 2019b; Alvarez-Melis and Jaakkola, 2018a; Koh et al., 2020; Lang et al., 2021) and language based systems (Hendricks et al., 2016).

Until now, we have clearly highlighted the advantages and disadvantages of various means of interpretation, laying an emphasis on concept-based interpretations. The next chapter will justify our choice of learning unsupervised dictionary of concepts. Interestingly, while there are numerous approaches for them at this point in time, only SENN (Alvarez-Melis and Jaakkola, 2018a) and ACE (Ghorbani et al., 2019) are strictly prior approaches to ours. A clear advantage of opting them for interpretation beyond the typical advantages of using concepts is their complete independence to any external algorithm. However, there are certain limitations to both. ACE, as covered earlier, is limited by its reliance on external algorithms and network representations that can meaningfully cluster superpixels. SENN on the other hand can be improved for its interpretations. The current proposed pipeline by SENN to discover/understand the concepts is to visualize the training samples maximally activating the concept. While certainly insightful, this puts a heavy onus on the user to derive their own meaning about set of images without any tool to understand it deeper. Thus, a similar criticism highlighted before for saliency maps can be levied for this pipeline to some extent, that it does not go deeper in to the question of “what" is the underlying concept captured among the set of maximally activating training samples. This is one of the issues we address for our work on interpretability for image classification.

Audio interpretability methods: Compared to other major modalities, audio interpretability has received sparse consideration. The progression of research has been similar to certain other modalities, such as graphs. We state this because of two reasons. The first being that post-hoc interpretation approaches have received the vast majority of attention compared to by-design interpretation approaches recently. Moreover, the research methods have developed in two directions following a sim-

ilar pattern as graph interpretability research. The first set of methods primarily utilized image interpretability techniques and demonstrated their utility for audio. These include saliency maps (Becker et al., 2018; Muckenhirn et al., 2019) and attention mechanisms (Won et al., 2019). The second are instead methods that specifically try to exploit structures in audio and modify previous proposals to suit audio. This has included extensions of LIME and TCAV algorithms for post-hoc interpretations (Mishra et al., 2017, 2020; Haunschmid et al., 2020; Chowdhury et al., 2021; Foscarin et al., 2022). Critically, these approaches focus more on aspects that make interpretations more comprehensible to users for audio.

In particular, SLIME (Mishra et al., 2017, 2020) proposed to segment the input along time or frequency. The input is perturbed by switching "on/off" the individual segments. AudioLIME (Haunschmid et al., 2020; Chowdhury et al., 2021) proposed to separate the input using predefined sources to create the simplified representation. AudioLIME arguably generates more meaningful interpretations than SLIME as it relies on audio objects readily listenable for end-users. However, it suffers from limited applicability, requiring existence of known and meaningful predefined sources that compose the input audio. More recently, Foscarin et al. (2022) extended the idea of TCAV to represent concepts in music data. The supervised approach requires the overhead of human annotation of concepts, whereas the unsupervised approached based on non-negative tensor decomposition faces the challenge of meaningful learning of concepts. APNet (Zinemanas et al., 2021) extends prototypical networks (Li et al., 2018) for audio input while addressing by-design interpretation by defining a more suitable distance measure for audio prototypes.

2.5 Evaluation of interpretation

To evaluate interpretations generated by an algorithm has been a very challenging task for researchers in this domain and continues to be a topic of active research. There are two major difficulties in evaluating interpretations. Firstly, it is extremely rare to have some ground truth to quantitatively compare an interpretation to. Secondly, there is strong element of subjectivity involved in the evaluation. Interpretations are typically generated for humans to gain insights about a model. Thus, this information needs to be in a form understandable to human. Recall that 'human understandability' is even part of the definition of interpretability we rely on. However, it is extremely hard to objectively quantify human understandability. What can be understandable for one person need not be for another one.

One of the earliest in depth discussion with a hawk-eye view of this topic can be found in the work by Doshi-Velez and Kim (2017). They propose to organize all the possible evaluation strategies in three different categories as shown in fig. 2.5. The three categories in decreasing order of cost are: 'application-grounded evaluation', 'human-grounded evaluation' and 'functionally-grounded evaluation'. The first of them 'application-grounded evaluation' is categorized as involving human experiments with real-world applications. The efficacy in this case is established by showing

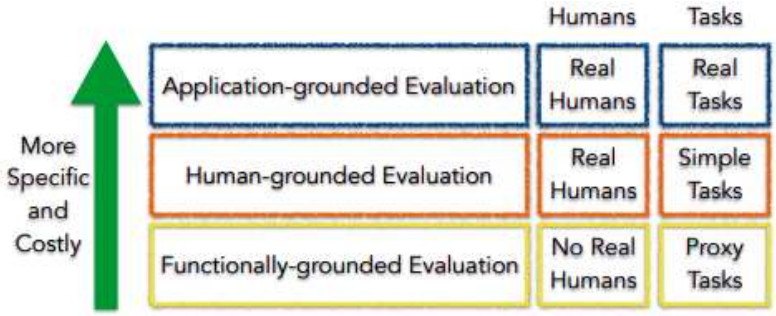


Figure 2.5: Categorization of evaluation strategies proposed by Doshi-Velez and Kim (2017).

improvement on a meaningful end-point for the application compared to a baseline, which can be another established model or a *human baseline*. A popular example for this would be doctors using an interpretability system to help their diagnosis (Koh et al., 2020). While the most effective, this evaluation is expensive to setup and conduct, requiring high standards of safety, experiment design, and potentially genuine implications on human lives.

The second category is that of 'human-grounded evaluation' which considers evaluation with real humans but simplified tasks. These don't directly evaluate performance of the system for an application it would be used for, but are generally used to evaluate certain subjective aspects of an interpretation, such as its understandability. Common examples of this evaluation is choosing between a pair of interpretations based on some qualitative criteria to assess their quality (Selvaraju et al., 2017). While cheaper than the previous category, conducting a human evaluation is always a costly endeavour, requiring care in experiment design and human time and labour cost. The last category is 'functionally-grounded evaluation' which does not rely on any human experiments and instead using a functional proxy to measure quality, such as sparsity or robustness of an interpretation.

Given the high costs involved in 'application-grounded evaluation' and even to a good extent for 'human-grounded evaluation', ML research community has made considerable attempts to advance 'functionally-grounded evaluation'. The most prevalent approach for conducting evaluation is to identify generic or task-specific definition/properties one desires from the interpretations. These properties are quantified through some proposed strategy which forms the basis for evaluation. It is important to emphasize that identifying desirable properties can be used to propose loss functions for training an interpretable model, a very frequent occurrence in the literature. However, at this point we are considering quantitative metrics to evaluate desirable properties.

As of yet, there is no formal definition or particular set of properties with general consensus across various tasks, models, input data or means of interpretation. Researchers have proposed different sets of desired properties of interpretations for different problem settings. Even for a specific property and a problem setting there have

multiple proposed strategies to quantify it. Among the attempts to unify evaluation framework for interpretability the most comprehensive attempt is that of Quantus package (Hedström et al., 2023). Their work is focused on implementing all the proposed properties and evaluation strategies for each property. Their primary focus is on saliency map approaches, and consequently raw input attribution based approaches. However, multiple metrics feature in one form or other for other means of interpretations, tasks and data modalities.

We list the major properties collated under the package and some popular strategies to quantify them.

Faithfulness

The "faithfulness" of a interpretation to the prediction tries to quantify whether the features identified as important by the interpretation are also "important" to the prediction function. There have been various proposed metrics that evaluate this aspect (Bach et al., 2015; Alvarez-Melis and Jaakkola, 2018a; Montavon et al., 2018; Arya et al., 2019). The common theme among most proposals is to quantify change in prediction when simulating "removal" of the most important features in the interpretation. A larger drop in predicted output (logit/probability for classification) indicates better faithfulness.

From a research perspective, the tricky part in any proposal of this metric is the feature removal step. The challenge lies in the fact that a feature removal strategy might push the modified sample out of the data distribution. How the prediction function is affected by this is generally unknown. This introduces an unintended factor that might affect change in prediction. This has led to experimentation with various feature removal strategies. Nevertheless, the core idea of faithfulness is a crucial property one expects from an interpretation.

Complexity

Complexity essentially measures the information content or conciseness of an interpretation provided to a user. All things equal, an interpretation utilizing lower number of features is more preferable. The previously proposed metrics include measuring gini-index of attribution map (Chalasani et al., 2020), entropy of fractional contribution of the features to the total attribution (Bhatt et al., 2020a) and number of features with attribution greater than a threshold (Nguyen and Martínez, 2020).

Robustness

For small perturbations to the input, if the prediction function is largely unchanged, one expects the interpretations to remain same as well. This is quantified by metrics for Robustness/Stability. The common idea behind most proposed metrics (Alvarez-Melis and Jaakkola, 2018b; Montavon et al., 2018; Agarwal et al., 2022; Yeh et al., 2019a) can be summarized as follows: If $S(x, f)$, $S(x', f)$ denote the saliency map or

input attribution for input x and its perturbed version x' respectively, then the various metrics typically aim to maximize the distance between attribution maps $\|S(x,f) - S(x',f)\|_2$ when normalized w.r.t the amount of perturbation $\|x - x'\|_2$. Lower values indicate higher stability. Some of the metrics vary in how they define the "amount" of perturbation by using other quantities like change in prediction output $\|f(x) - f(x')\|_2$ instead of distance between the inputs. Nevertheless, the core idea about computing stability remains the same.

Localization

This is more commonly used for object classification task in images with known bounding box or segmentation mask annotation. It evaluates how well the attribution map aligns with the bounding box or segmentation mask. As with the other metrics, it can be quantified in different ways. Zhang et al. (2018a) for instance check if location of attribution with highest value is inside the target area or not. Kohlbrenner et al. (2020) on the other hand compute the fraction of positive attributions overlapping with the target area.

Axiomatic

These metrics quantify how well do interpretations align with certain axiomatic properties in the literature (Kindermans et al., 2019; Sundararajan et al., 2017; Nguyen and Martínez, 2020). For instance, Kindermans et al. (2019) measure the variation in interpretation when a shift is added to input. Assuming the method is applied on a model invariant to input shift (eg. CNNs), the interpretation is not supposed to be equivariant to shift, but the relevance values are not supposed to change (taking into account the shift).

2.6 Summarizing relevant themes to the thesis

To conclude, this chapter was focused on categorizing the literature from the point of view of context factors of an interpretability application as well as cover details about popular methods related to our framework and its applications to image and audio modality. The context factors themselves were organized in three separate bins. Factors characterizing a problem, characterizing different interpretability algorithms and evaluation of interpretations. We now summarize the themes briefly and highlight where the research in this thesis will operate at the with respect to categorization laid out:

- *Problem characteristics:* We highlighted multiple factors characterizing an interpretability problem but in particular laid strong emphasis on three of them – 'problem type', 'scope' and 'input modality'. In chapter 3, we will develop a framework which can tackle both post-hoc and by-design interpretation problems, and possibly even some recently proposed variants. The framework will

only learn or interpret neural networks and will be capable of generating both local and global interpretations. Chapter 4 will study the application of this framework for image modality. Chapter 5 will study application of this framework for audio modality.

- *Means of Interpretation:* We elaborated different representations of interpretations proposed and used throughout the literature, ranging from low-level representations such as raw-input to human language itself. As will be argued in chapter 3, the means of interpretations for our framework will be closely related to the unsupervised concept-based interpretation methods. However, there is a small note needed to be made regarding strong relationship between 'means of interpretation' and 'input modality'. The restriction of the data modality needed to be processed can impact the choice for representation for interpretation. Using a raw input features as the representation can be regarded lot more meaningful for tabular data, than graphs/audio. The data modality can alter the desired nature of interpretations and thus the representation used for it. This relationship plays an important role in motivation of our choice for representation of interpretation in chapters 4 and 5. While we improve upon the unsupervised concept based interpretations in case of images, for audio modality we are lot more interested in generating listenable concept-based interpretations. This motivation will render the representations employed for images unusable and lead us to novel representation for interpretation for audio.

- *Interpretation Evaluation:* We discussed broad categories of interpretability evaluation with different specificity and cost, as well as strategies of quantitative metrics (part of 'functionally-grounded evaluation') typically used for saliency maps or other raw-input attribution approaches. These strategies and categories will form the basis for how we evaluate interpretations in our framework which we discuss in chapter 3.

Contents

3

Developing a Framework to Learn with Interpretation

Contents

3.1 Introduction

The objective of this chapter is to develop a general framework to address post-hoc and by-design interpretability in the context of supervised classification. To define a single learning problem that can tackle both interpretability problems, we begin by recapping the formulation of a supervised classification problem. We then extend it solve an interpretation task in addition to the prediction task. In order to do this, we define a predictor, its dependent interpreter, each dedicated for one task, and single learning objective to learn them. This leads to the formulation of the central learning problem of our framework, titled Supervised Learning with Interpretation (SLI). Following this, we develop the internal structure of the interpreter. We motivate the representation of interpretation and our method to enable this learning. The representation is based on learning unsupervised concept dictionary, given the discussion about various representations in the previous chapter, and we enable

the interpreter to learn such representations by giving it access to intermediate outputs of predictor. We continue to develop the structure of interpreter by defining properties suitable for its desired function and corresponding loss functions that formulate our learning objective. After laying out the skeleton of the architecture, we sketch an outline to define the interpretation task. It consists of formulating a process to generate local and global importances over our dictionary of concepts based on a novel notion of relevance and discovering/understanding information encoded by each element of the dictionary. We then concretely discuss how the SLI objective can be used to tackle post-hoc and by-design interpretability, and even the recently proposed variant ante-hoc interpretability. Finally, we conclude with discussion on evaluation of interpretations in our framework. We list some potential metrics that quantify various aspects of interpretation, related to previously proposed evaluation strategies discussed in chapter 2. We rely on these metrics for evaluating our interpretations for later chapters when we instantiate our framework for image and audio modalities.

3.2 Moving towards a single learning framework

It is worth spending some time describing the underlying task of supervised classification for which we wish to address the problem of interpretability

A typical instance of any supervised learning task assumes an underlying unknown probability distribution $\mathcal{P}$ over an input space $\mathcal{X}$ and output space $\mathcal{Y}$. We also assume a given training set $\mathcal{S} = \{(x_i, y_i)_{i=1}^N\}$ composed of N independent realizations of a pair of random variables (X, Y) over $\mathcal{P}(\mathcal{X} \times \mathcal{Y})$. On top of this, one defines a hypothesis space $\mathcal{F}$, which denotes a space of predictive models from $\mathcal{X}$ to $\mathcal{Y}$. It is a set of candidate functions $f \in \mathcal{F}, f : \mathcal{X} \to \mathcal{Y}$, for example linear models, decision trees, single layer neural networks etc. over which we wish to select the most "suitable" function for prediction. To determine the suitability of a candidate function from the hypothesis space, a loss function $\mathcal{L}_{pred} : \mathcal{F} \times \mathcal{P}_N \to \mathbb{R}_+$ is defined, where $\mathcal{P}_N$ denotes space of sets of size N that can be independently drawn from $\mathcal{P}$. This loss function is then empirically minimized over the given dataset $\mathcal{S} \in \mathcal{P}_N$ to select the most suitable prediction function:

$$f^* = \underset{f \in \mathcal{F}}{\operatorname{argmin}}\, \mathcal{L}_{pred}(f, S)$$

Given any new input $\hat{x} \in \mathcal{X}$, the selected function f^* can be used for the **prediction task** as $\hat{y} = f^*(\hat{x})$, crucial for any supervised learning problem. For a supervised classification task, $\mathcal{Y} \subset \mathbb{R}^C$ where C denotes the number of classes. In case of multi-class classification, $\mathcal{Y} = \{y \in \{0,1\}^C, \sum_{c=1}^C y^c = 1\}$, which is the set of one-hot encoding vectors of dimension C. On the other hand, for multi-label classification, $\mathcal{Y} = \{y \in \{0,1\}^C\}$.

The choice of hypothesis space $\mathcal{F}$ as deep neural networks (including CNNs, RNNs, Transformers etc.) is arguably the most popular choice among the ML research community in recent years, given the state-of-the-art performance these models have achieved over a host of different classification problems. Thus, we treat this as the

starting point to build our learning framework. Throughout the discussion in this and later chapters, we will concern ourselves with supervised classification as the underlying task and deep neural networks as hypothesis space.

Our goal is now to design a single learning problem that can be seamlessly adapted for multiple interpretability applications. In order to achieve this, note that prediction and interpretation both should be essential tasks of the framework. This leads to the idea of incorporating interpretability as an objective in the learning of the model itself. However, a second key idea we rely on is that the **interpretation** task differs from the **prediction** task and requires a dedicated model that depends on the predictive model to be interpreted. For a given model $f \in \mathcal{F}$, we denote $\mathcal{G}_f$ the family of models $g_f : \mathcal{X} \to \mathcal{Y}$, that depend on f and are devoted to its interpretation. For sake of simplicity, an interpreter $g_f \in \mathcal{G}_f$ is denoted g, omitting the dependency on f. With these assumptions, the empirical loss of supervised learning is revisited to include explicitly an interpretability objective besides the prediction loss, yielding the following definition.

Supervised Learning with Interpretation (SLI):

$$\textbf{Problem 1}: \underset{f\in\mathcal{F}, g\in\mathcal{G}_f}{\operatorname{argmin}} \mathcal{L}_{pred}(f,\mathcal{S}) + \mathcal{L}_{int}(f,g,\mathcal{S}) \tag{3.1}$$

where $\mathcal{L}_{int}(f,g,\mathcal{S})$ measures the ability of g to provide interpretations of predictions by f and $\mathcal{L}_{pred}$ is the prediction loss as before. This tasks the deep neural network f with prediction and its dependent model g with its interpretation. It is worth noting that formulating learning problems with multiple objectives with different models addressing each one is frequent occurrence in machine learning literature. Garcia et al. (2018) for instance, learns two functions, one for structured output prediction and other to predict whether to abstain or not from prediction on the current input. A large number of methods in knowledge distillation literature (Gou et al., 2021) contain two different models, a teacher and a student, both trained to address different objectives.

We now dive deeper into how the interpreter g can be designed, including its dependence on f. This will serve two purposes. Firstly, it will set the foundation for how the interpretations will be generated. Moreover, it will provide a further perspective on specific cases of Pb. 3.1 that can be used for different interpretability applications.

3.3 Developing interpreter structure

3.3.1 Motivation for design

To motivate the structure of interpreter it is essential to start conceptualizing some key parts of its design. Namely, (a) what should be its means of interpretation?, and (b) how should it be dependent on f such that it is in a “suitable" position to interpret output of f.

We start with its means of interpretation, which refers to what representation it relies on to generate an interpretation. The previous chapter delved into detail about various proposed means in literature. One of the interesting observations resulting from the categorization was the use of non-input attribution based representations for interpretation. For almost any data modality, it is preferable to employ a representation that has the potential capacity to capture semantic features, useful for human reasoning and communication. In case of tabular or text data this purpose is well served by the raw input representation itself. However, this does not translate to many other modalities. For high-dimensional modalities like images or audio, the raw/simplified representation of an input is quite different from abstract representations as in human communication. Prototype or concept-based interpretations offer much better prospects at capturing such abstract representation, and provide finer-level details about what features are relevant to the decision. Moreover, they possess better potential to also generate global interpretations. The criticism about interpretability of distance computation in prototype based interpretations however pushes us more towards opting for concept-based representation.

For high flexibility of the framework we would also like to rely on a representation that can be learnt without any additional information. This discourages use of concept representation as proposed in TCAV (Kim et al., 2017). Selecting a unsupervised representation that can encode semantically meaningful information can be modality-driven choice which will be explored further in later chapter. Nevertheless, unsupervised concept-based representation as in SENN (Alvarez-Melis and Jaakkola, 2018a) aligns with the above criteria, and serves as a reasonable general candidate to begin with.

A natural question to then ask is from where and how can this representation be learnt so that it can also simultaneously offer insight about the classifiers decision process. This question connects strongly to second part of the design which asks how should the interpreter be dependent on the classifier. Our solution to this is quite simple. The hidden layers of the classifier, especially the ones close to the output capture bulk of the features that the classifiers relies on for its prediction. Moreover, deeper layers of neural networks have been known to capture more higher-level/abstract features (Bengio et al., 2009; Zeiler and Fergus, 2014). They can act as a great source for the interpreter to not only learn a classifier-specific high-level representation, but also very intimately tie it down to the classifiers decision process.

This offers us tremendous benefits in terms of application, flexibility and zero annotation cost, but it also comes with its own set of challenges. Learning such a representation would put the onus on designing the interpretability objective $\mathcal{L}_{int}$.

3.3.2 Formalizing the structure

Without much loss of generality we assume that f is a deep neural network with J hidden layers of respective dimension $j_1, \ldots, j_J$. Each element $f : \mathcal{X} \to \mathcal{Y}$ of $\mathcal{F}$ satisfies: $f = f_{J+1} \circ f_J \circ \ldots \circ f_1$ where $f_j : \mathbb{R}^{d_{j-1}} \to \mathbb{R}^{d_j}$, $d_0 = d, d_{J+1} = C, j = 1, \ldots, J+1$ is the

function implemented by layer j. As for the interpreter model $g \in \mathcal{G}_f$, we propose the following original architecture which exploits the outputs of T chosen hidden layers of f. Denote $\mathcal{I} = \{i_1, i_2, ..., i_T\} \subset \{1, \ldots, J\}$ the set of indices specifying the intermediate layers of network f to be accessed and chosen for the representation of input. We define $D = \sum_{t=1}^{T} d_{i_t}$. Typically these layers are selected from the latter layers of the network f. The concatenated vector of all intermediate outputs for an input sample x is denoted as $f_{\mathcal{I}}(x) \in \mathbb{R}^D$.

Given f a network to be interpreted and a positive integer $K \in \mathbb{N}^*$, an **interpreter network** g computation can broken down in two parts. First, it extracts a dictionary of attribute functions $\Phi_{\mathcal{I}} : \mathcal{X} \to \mathcal{Z}^K$. by processing the selected intermediate outputs $f_{\mathcal{I}}(x)$, through a function $\Psi : \mathbb{R}^D \to \mathcal{Z}^K$. That is, $\Phi_{\mathcal{I}}(x) = \Psi \circ f_{\mathcal{I}}(x)$. The primary object of interest here is the **attribute dictionary** $\Phi_{\mathcal{I}}$. It is composed of K individual functions $\phi_k : \mathcal{X} \to \mathcal{Z}, k = 1, \ldots, K$ and $\mathcal{Z} = \mathbb{R}_+^{d'}, d' \in \mathbb{N}^*$. Each $\phi_k(x)$ represents presence or activation of a high-level attribute, i.e a "concept" over $\mathcal{X}$. An individual functions output domain $\mathcal{Z}$, is a non-negative orthant and denotes a relevant activation space for a modality. In case of images (chapter 4) d' will be simply set to 1 and in case of audio (chapter 5) it will denote the number of time frames of a spectrogram. The precise meaning of a high-level attribute, and how Ψ needs to be designed is dependent on the modality one is operating on and how $\Phi_{\mathcal{I}}$ will be defined to interact with input space $\mathcal{X}$. One can note that even though the function Ψ is operating over the space of selected hidden layers, the attribute dictionary is defined as operating over the input space. While it can be considered as a superficial detail, it aligns with an important idea of understanding an attribute function through the input space, which is generally interpretable for the user.

The second part in g's computation consists of processing the attributes $\Phi_{\mathcal{I}}(x)$ through a function $\Theta : \mathcal{Z}^K \to \mathcal{Y}$. Ideally, Θ needs to be designed in a way to provide easy mechanisms for quantifying importance of any individual attribute function ϕ_k in the interpreters output. Linear models or any other differentiable interpretable models, for example variations of neural additive models (Agarwal et al., 2020; Radenovic et al., 2022), serve as good candidates for its architecture. The computation of the interpreter can thus be summarized as below:

$$g(x) = \Theta \circ \Phi_{\mathcal{I}}(x) = \Theta \circ \Psi \circ f_{\mathcal{I}}(x) \tag{3.2}$$

The learnable parameters of Ψ, Θ are exclusive to g, which also shares parameters with f upto the last intermediate layer accessed. We summarize the key notation introduced until now in Tab. 3.1

3.3.3 Learning by imposing interpretability properties

Having formalized our initial idea behind designing the interpreter and its representation of interpretation, the follow-up question to consider is how to learn such a representation. Our main mechanism for learning is to encourage various properties relevant for interpretability by incorporating them as loss functions in the object-

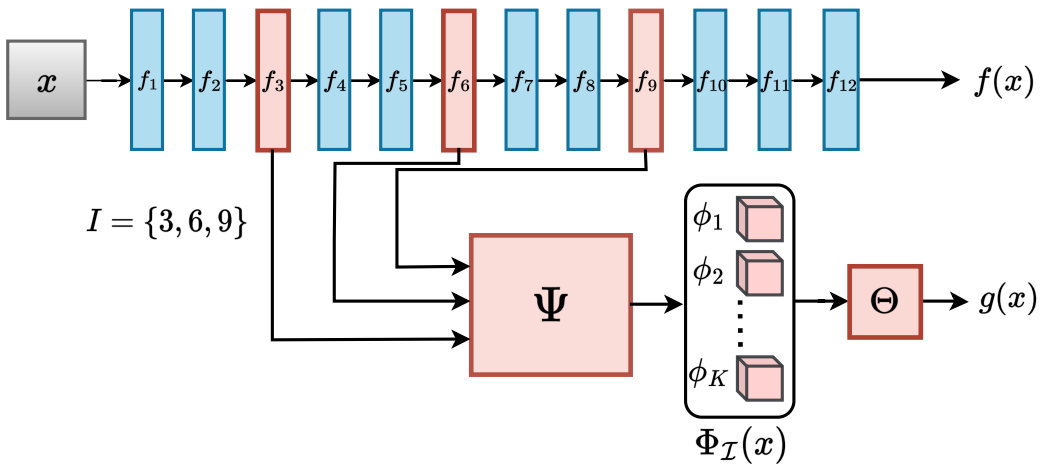


Figure 3.1: Example architecture of predictor and interpreter in our framework.

Symbol	Description
$\mathcal{X}$	Input space
$\mathcal{Y}$	Output space for classification
$f : \mathcal{X} \to \mathcal{Y}$	Predictor deep neural network with J intermediate layers
$g : \mathcal{X} \to \mathcal{Y}$	Interpreter network dependent on f through its intermediate layers
$\mathcal{I}$	Set of selected intermediate layers of f accessed by g
$f_{\mathcal{I}}(x) \in \mathbb{R}^D$	Concatenated output of selected layers of f of size D
$\Phi_{\mathcal{I}} : \mathcal{X} \to \mathcal{Z}^K$	Dictionary of K attribute functions over $\mathcal{X}$
$\phi_k : \mathcal{X} \to \mathcal{Z}$	Individual attribute function, computes activation of a concept over $\mathcal{X}$.
$\Psi : \mathbb{R}^D \to \mathcal{Z}^K$	Part of g. Computes $\Phi_{\mathcal{I}}(x)$ from $f_{\mathcal{I}}(x)$
$\Theta : \mathcal{Z}^K \to \mathcal{Y}$	Computes $g(x)$ from attribute activations $\Phi_{\mathcal{I}}(x)$

Table 3.1: Key notation summarizing computation of interpreter g.

ive $\mathcal{L}_{int}$. This leads to the question of what are the essential properties that should be incorporated in $\Phi_{\mathcal{I}}$. The precise answer to this is again highly dependent on the modality and requirements for what is the desired information to be encoded by $\Phi_{\mathcal{I}}$. Moreover, even among other recent works relying on learning such a representation, there is no clear agreement. However based on our conceptual ideas about $\Phi_{\mathcal{I}}$, and previous unsupervised concept learning systems, we propose a minimal initial set of properties and corresponding loss functions. We primarily attend to three properties that we consider essential to our design and then discuss some other additional properties. For any instance of the framework in the later chapters, the corresponding loss functions of the three properties are linearly combined to form $\mathcal{L}_{int}$.

Note that these properties differ in a crucial way with previous axioms/constraints for interpretability such as ones proposed by Lundberg and Lee (2017); Sundararajan et al. (2017). These methods were not tasked with learning representation for interpretation. They enforced the properties to learn importance values of features, and not for learning the features.

Fidelity to Output

The interpretation is supposed to be generated for the predictors output. It is thus natural to expect the interpreter to be able to approximate the predictors output. We term this as encouraging output fidelity. A typical choice of implementing this as loss function would be to minimize generalized cross-entropy loss between $g(x)$ and $f(x)$. For multi-class classification tasks, this loss can be written as:

$$\mathcal{L}_{of}(f, g, \mathcal{S}) = -\sum_{x \in \mathcal{S}} g(x)^T \log(f(x))$$

For multi-label classification tasks, it writes slightly differently as:

$$\mathcal{L}_{of}(f, g, \mathcal{S}) = -\sum_{x \in \mathcal{S}} g(x) \odot \log(f(x))$$

Remark 3.1. *The learnt representation $\Phi_{\mathcal{I}}(x)$ is tied down to the predictor computation in two ways. First, by construction, it is forced to be generated from certain hidden layers which the predictor relies on for its output. And second, the output fidelity loss encourages that this representation be generated in a way that it can retrieve predictor's output.*

Fidelity to Input

The second essential property is connected to understandability of $\Phi_{\mathcal{I}}$. Every interpretation method proposed until now ultimately makes use of input space for understanding an interpretation. We wish to encode information about activation or presence of a concept over $\mathcal{X}$. This can be viewed as each ϕ_k encoding information about higher order features related to input. A common structure that is used to impose this property, utilized by previous interpretable networks (Alvarez-Melis and Jaakkola, 2018a; Li et al., 2018), is to treat $\Phi_{\mathcal{I}}(x)$ as encoding for an autoencoder. We thus introduce a decoder function $d : \mathcal{Z}^K \to \mathcal{X}$. The primary goal of decoder function d is to reconstruct input x using the dictionary of attribute functions $\Phi_{\mathcal{I}}(x)$. The loss function for this can be implemented in many distinct ways but as a starting point we propose the mean squared error

$$\mathcal{L}_{if}(f, g, d, \mathcal{S}) = \sum_{x \in \mathcal{S}} \|d(\Phi_{\mathcal{I}}(x)) - x\|_2^2 = \sum_{x \in \mathcal{S}} \|d(\Psi(f_{\mathcal{I}}(x))) - x\|_2^2$$

At this point we refrain from going into further details about designing d or the loss function $\mathcal{L}_{if}$. However, it is crucial to emphasize the impact of these choices. The function d heavily affects the type of features encoded in $\Phi_{\mathcal{I}}$ and will be a central theme when we apply the framework for audio classification. Its design is heavily linked to structure of $\Phi_{\mathcal{I}}$ and vice-versa along with particular requirements in a specific problem setting.

Conciseness

For any given sample x, one expects that only a small number of high-level concepts are present. Thus, sparsity of attribute activations is expected to reflect encoding of abstract latent information about the input. The simplest and most generic choice to impose sparsity of activations is penalizing ℓ_1 norm of activations $\Phi_{\mathcal{I}}(x)$.

$$\mathcal{L}_{\text{conc}} = \|\Phi_{\mathcal{I}}(x)\|_1$$

From a learning point of view, penalizing ℓ_1 norm has been a useful regularization term (Bank et al., 2020). However, it also has roots in encouraging interpretability of learnt representations (Lage et al., 2018). In the context of our framework, this benefits learning of $\Phi_{\mathcal{I}}$ in two ways. First, as we already indicated, activation of small number of attributes should reflect their abstract character. Secondly, since one would need to understand concept encoded by an attribute function "relevant" to the decision, conciseness assists in *reducing* number of attributes required to be analyzed during interpretations.

Other properties

It is important to emphasize that the above discussion only sets a rough pathway for what properties and loss functions one can impose to learn $\Phi_{\mathcal{I}}(x)$. There remains significant room for innovation and modification. One can potentially use different ways of imposing the losses. For example, in the next chapter focused on images as input, we present an entropy+ℓ_1 based loss for conciseness with its own advantages and disadvantages. Similarly, one can extend the set of properties and include dedicated losses. We list one such property below as an example:

Stability: It is desirable for the attribute dictionary $\Phi_{\mathcal{I}}$ to remain stable w.r.t any x, i.e. small variations in x should not vary $\Phi_{\mathcal{I}}(x)$ much. A possible approach from representation learning literature is to minimize the squared Frobenius norm of jacobian of $\Phi_{\mathcal{I}}$ w.r.t x, $\|\nabla_x \Phi_{\mathcal{I}}(x)\|_2^2$. Maximizing cosine similarity in a contrastive learning setup is also another possible option.

3.3.4 Detailing the Interpretation Task

As mentioned before, we view interpretation as an additional task besides prediction. Having discussed the design of interpreter and joint learning objective, we need to specify its expected output. We thus build a tentative road map for generating local or global interpretations. Like many other aspects in the design, what functions as the final interpretation is dependent on the specific problem and its requirements. Nonetheless the design discussed until now endows upon the framework a natural pathway to generate an interpretation. We start by elaborating upon what pieces of information give insights about the model's decision process at a local or global scope.

- Note that we want to generate our interpretations through the dictionary of attribute functions $\Phi_{\mathcal{I}}$. Thus, the first crucial piece of information we desire is to understand the relationship between each element of $\Phi_{\mathcal{I}}$ and the interpreters output. For local interpretations this corresponds to which attribute functions are important for the predicted class for a given sample and to what extent. For global interpretation, this corresponds to which attribute functions are generally important in prediction of which classes and to what extent.
- Understand information encoded by any relevant attribute function ϕ_k. This process or pipeline completely depends on the modality, design of $\Phi_{\mathcal{I}}$, and our intended goal with interpretation. We develop two different designs and information understanding procedures for image and audio modalities in the later chapters.

We can illustrate the above ideas as an example to improve the clarity: An image classifier might predict a particular sample as 'Cat' based on detecting 'Pointy-ears' and 'Legs'. However across the dataset other attributes like 'Whiskers' and 'Triangular-nose' might also be useful in predicting 'Cat' class. In this analogy, the goal of learning $\Phi_{\mathcal{I}}$ would be to encode such high-level information in each individual attribute function. Computing local relevance would correspond to computing importance of each attribute functions in predicting 'Cat' for the current sample. The attribute functions corresponding to 'Pointy-ears' and 'Legs' will have high local relevance in the above example. Computing global relevance would correspond to computing importance of each attribute function in prediction of any class. Attributes functions corresponding to 'Pointy-ears', 'Legs', 'Whiskers' and 'Triangular-nose' will all have high global relevance w.r.t 'Cat' class. Understanding information encoded in $\Phi_{\mathcal{I}}$ would correspond to visualizing and establishing the relationship between individual attribute function ϕ_k and their detection or activation for individual concepts, 'Whiskers', 'Pointy-ears' etc. Next, we build our road map to generate local or global relevance. The process of understanding information encoded by attribute functions is design-specific and is handled individually for images and audio in later chapters.

Generating relevances: We define the notion of a local relevance $r_{\text{loc}} : \mathcal{X} \times \mathcal{G}_f \times \mathcal{F} \rightarrow [-1, 1]^K$ and global relevance $r_{\text{glo}} : \mathcal{G}_f \times \mathcal{F} \rightarrow [-1, 1]^{K \times C}$ to store the information about importance of an attribute to interpreter's output. For local relevance, we implicitly assume it is generated specifically for a predicted class. If there are multiple predicted classes as is often a case in multi-label classification, the same process can be repeated for each class separately. To simplify notation, we'll refer to local relevance associated with any attribute k for a sample x as $r_{k,x}$, and $r_{k,c}$ as its global relevance for any class c. These values are computed w.r.t a given interpreter g and predictor f. The subscript and context will make it clear which scope (local or global) we are referring to.

To be able to quantify local relevance $r_{k,x}$ for any attribute k given an input x, we wish to combine the two parts of the decision process. The first is the activation of the attribute, extracted from $\phi_k(x)$. The second is how it affects the output of Θ for some specific class c (typically a predicted class). The generic notion to quantify the effect

would be to use the gradient $\nabla_{\phi_k}\Theta(\Phi_{\mathcal{I}}(x))_c$. However, as done in the later chapters, a simpler Θ makes this process straightforward. For example, if $\phi_k(x) \in \mathbb{R}_+$ and Θ was simply used as a fully connected layer with weight matrix W, then the jacobian $J_{\Phi_{\mathcal{I}}}\Theta(\Phi_{\mathcal{I}}(x)) = W$. Regardless, to quantify local relevance the key idea is to combine the activations $\phi_k(x)$ and their usage for the decision $\nabla_{\phi_k}\Theta(\Phi_{\mathcal{I}}(x))_c$. It is worth noting that this process of quantifying relevance of an attribute is similar to using input times gradient as saliency maps rather than just gradient.

If one could generate a local relevance $r_{k,x}$, it can easily be extended to quantify global relevance of an attribute k to class c, $r_{k,c}$. This can be accomplished by simply averaging the local relevance $r_{k,x}$ across the dataset $\mathcal{S}$ for samples where the predicted class is c. That is, $r_{k,c} = \frac{1}{|\mathcal{S}_c|}\sum_{x\in\mathcal{S}_c} r_{k,x}$, $\mathcal{S}_c = \{x \in \mathcal{S} | \hat{y} = c\}$

We are finally ready to define local and global interpretation.

Definition 3.2 (Global and Local Interpretation)**.** *For a prediction network f, the* **global interpretation** *$G(g, f)$ provided by an interpreter g, is the set of class-attribute pairs (c, ϕ_k) such that their global relevance $r_{k,c}$ is greater than some threshold $\tau, 0 < \tau < 1$. A* **local interpretation** *for a sample x provided by an interpreter g of f denoted $L(x, g, f)$ is the set of attribute functions ϕ_k with local relevance score $r_{k,x}$ greater than some threshold $\tau, 0 < \tau < 1$.*

$$L(x, g, f) = \{\phi_k : r_{k,x} > \tau\}$$
$$G(g, f) = \{(c, \phi_k) : r_{k,c} > \tau\}$$

These definitions essentially help in identifying which attribute functions are important in prediction for a sample x or for any class c by using a threshold $0 < \tau < 1$ for the relevances. Analogous to previous methods, we keep obtaining importances over representation for interpretation as focus of the interpretation task, separate from the process of understanding the underlying representation.

3.4 Variations of SLI objective

The goal of this section is to expound on how the Supervised Learning with Interpretation (SLI) objective described in Eq. 3.1 can be employed to tackle various types of interpretability problems. The previous chapter discussed the various problems, the primary ones being post-hoc interpretation and by-design interpretation. Additionally, we discussed two other problems addressed in the literature: ante-hoc interpretation and black-box regularization. The last problem is not covered by the SLI formulation as it considers the predictor as a black-box and the interpretation is exclusively tasked to g. We now go over each of the other three problems separately, recall them and discuss how the joint learning objective of SLI can be adapted for each. This discussion is visually summarized in Fig. 3.2.

3.4.1 Post-hoc interpretation

A post-hoc interpretation problem assumes a fixed trained predictor for performance $\hat{f}$ and requires to interpret its decisions. A special case of SLI problem can be easily applied to address this problem wherein one fixes $f = \hat{f}$ and optimizes the interpretability objective $\mathcal{L}_{int}$ with respect to the interpreter g only. In particular, the Ψ, d and Θ functions are optimized and all other parameters are kept fixed.

$$g^* = \underset{f=\hat{f}, g \in \mathcal{G}_{\hat{f}}}{\operatorname{argmin}} \mathcal{L}_{int}(\hat{f}, g, \mathcal{S}), \tag{3.3}$$

By default, the predictor $f = \hat{f}$ is tasked with the prediction and interpreter $g = g^*$ is responsible for interpretation task. While this is obvious from the problem definition, this can become slightly contentious for other problems as seen later.

3.4.2 By-design interpretation

The by-design interpretation problem requires to train an interpretable model that can exhibit high performance.The predictor and interpreter both need to be trained and one can simply use the original SLI objective for this.

$$f^*, g^* = \underset{f \in \mathcal{F}, g \in \mathcal{G}_f}{\operatorname{argmin}} \mathcal{L}_{pred}(f, \mathcal{S}) + \mathcal{L}_{int}(f, g, \mathcal{S}) \tag{3.4}$$

However there is an contentious point to be resolved here. The original SLI objective assigns the prediction task to $f = f^*$ and interpretation task to $g = g^*$. The contention arises around if f should be considered as interpretable by-design. One can possibly argue that parameters in f are also regularized by the interpretability objective and given the dependency of g to f, it is reasonable to consider f as by-design interpretable model and g provides its interpretation. However, using the same model for interpretation and prediction can also be argued as a fundamental requirement of the problem statement. In this case the original SLI assignment of tasks is problematic. Instead, using $g = g^*$ for prediction and interpretation is the more preferred choice. Note that this assignment does not affect the original SLI objective.

However, using g for final prediction and interpretation does raise one question regarding formulation of the framework. Does one need f in its entirety? Can we throw away the final layers of f after the last hidden layer accessed by g, i.e $\{f_l : l > \max(\mathcal{I})\}$ and only apply a prediction loss at the output of g. The training loss in this case would be modified by replacing $\mathcal{L}_{pred}(f, S)$ with $\mathcal{L}_{pred}(g, S)$ and not using any output fidelity loss in $\mathcal{L}_{int}$. We will consider this question experimentally in the later chapters. However as it will turn out, training the hidden layers through the prediction loss on output of f still plays an important role in optimal performance of g.

3.4.3 Ante-hoc interpretation

The ante-hoc interpretability problem as proposed by Sarkar et al. (2022) assumes a given trained predictor $\hat{f}$. However, unlike for post-hoc interpretation, the parameters of the predictor can be modified and the task is to learn a dependent interpreter which can tackle both interpretation and prediction. One can again employ the SLI objective for this task but simply initialize f with $\hat{f}$. This can be summarized by the learning objective below:

$$f^*, g^* = \underset{f \in \mathcal{F}, g \in \mathcal{G}_f, init(f)=\hat{f}}{\operatorname{argmin}} \mathcal{L}_{pred}(f, \mathcal{S}) + \mathcal{L}_{int}(f, g, \mathcal{S}) \tag{3.5}$$

Again by problem definition, the responsible models for prediction and interpretation tasks are clearly defined, both handled by the interpreter $g = g^*$.

3.5 Evaluation of interpretations

Having discussed the generic design aspects of the framework and the various learning objective formulations for different problems, the only remaining aspect is the evaluation of interpreter. It is important to identify the similarities and differences in the aims of this section, and discussion about evaluation of interpretations in Chapter 2. The previous chapter considered the most prominent evaluation strategies across the whole literature which were mainly focused on input attribution based methods. While we consider the goals of these strategies as the starting point for evaluation of interpretations in our framework, many of the previous metrics are not applicable as it is. Some of them need to be adapted for concept-based interpretations, while some others are not applicable at all. Our focus here is on the evaluations that can be carried out quantitatively and validate our objectives for interpreter design. Hence, aspects like understandability, which are highly subjective and need subjective evaluation, are not covered in this section.

Prediction performance

For ante-hoc and by-design interpretation, high prediction performance of the interpreter is an important requirement. The prediction performance of g is thus a important metric for these tasks. If one intended to use f as final prediction model, the prediction performance of f could also be essential to measure.

Fidelity of interpreter g to f

For post-hoc interpretation or for cases when f is being used for prediction, it is important to assess how well can the interpreter g approximate or imitate f. The simplest strategy to conduct this assessment is to use classification performance metrics while treating the interpreter output as prediction and classifier output as ground-truth. Such metrics can be seen as measuring effect of output fidelity loss $\mathcal{L}_{of}$, which explicitly encourages output of g be close to f. For multi-class classification tasks this

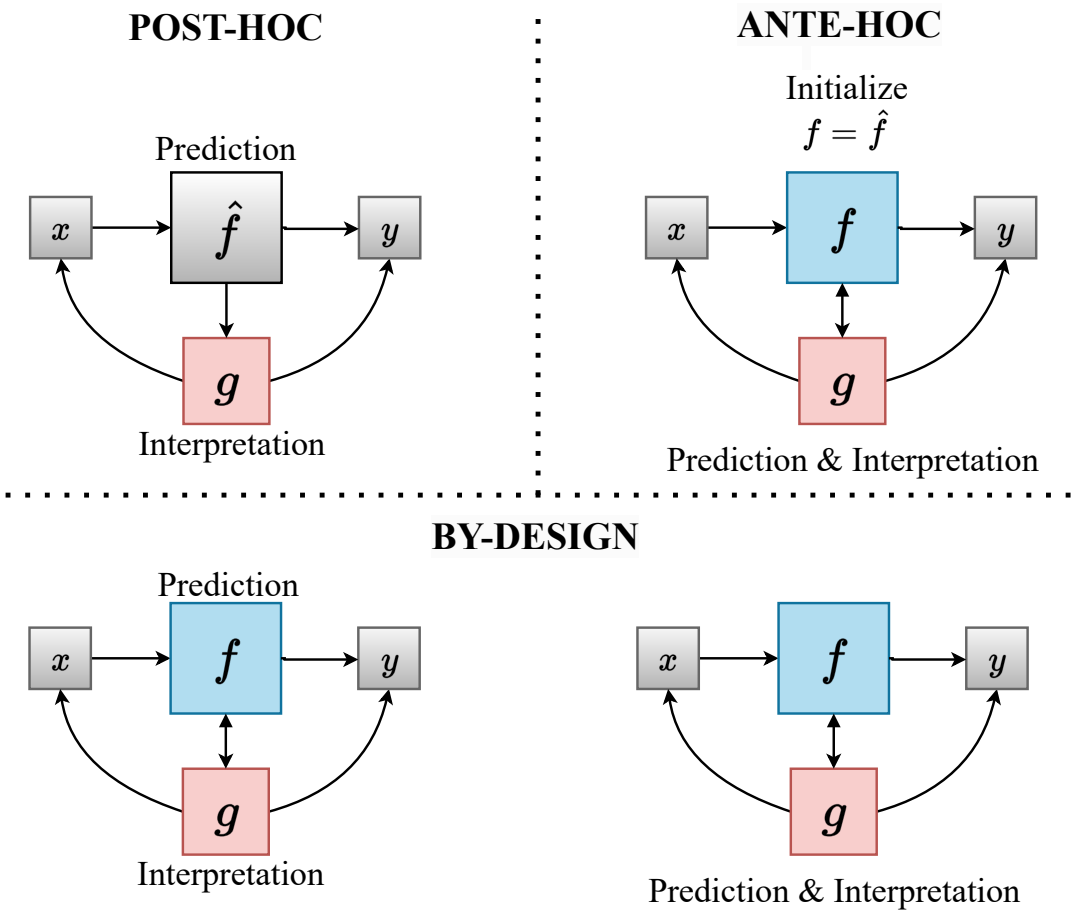


Figure 3.2: Variations of SLI objective for post-hoc, by-design and ante-hoc interpretability problems. For post-hoc, the prediction model $f = \hat{f}$ is fixed and only part of SLI objective is optimized. By-design interpretability can be tackled with proposed SLI objective but the assignment of prediction task can be done to f (left) or g (right). Using g for both prediction and interpretation more accurately aligns with the problem statement. Ante-hoc interpretability can also be addressed with SLI objective but initializing f with a pre-trained $\hat{f}$.

can correspond to "accuracy" of g to predict the top class of f (irrespective of true label). For multi-label classification this can correspond to F1 or AUPRC-based metrics between $g(x)$ and $f(x)$.

There are a few observations one can make about this evaluation w.r.t previous. This can be regarded as a functionally-grounded metric (Doshi-Velez and Kim, 2017), and has been used for evaluation in very similar form in other works (Bang et al., 2021; Lakkaraju et al., 2020) wherein they measure how well do the predictor and interpreter output match.

Conciseness

The complexity axis for evaluation of an attribution map was used to measure the conciseness, that is, how many features were being used for an interpretation. Lower complexity is desired behaviour for interpretations. A similar parallel can be drawn for our means of interpretation. Given a fixed threshold, a typical local interpretation in our framework would always involve selection of a subset of attribute functions that are considered relevant for the decision. Since interpretation requires understanding an individual attribute function, the information provided as interpretation is directly proportional to number of relevant attribute functions. This gives a notion of measuring complexity of local interpretations in our framework. We term this metric conciseness and it is computed as follows:

$$\text{CNS}_{x,g} = |\{k : |r_{k,x}| > \tau\}|$$

Our method of measuring conciseness can be considered similar to effective complexity metric proposed by Nguyen and Martínez (2020). On a separate note, $\text{CNS}_{x,g}$ can also be seen as quantifying the effect of imposing the conciseness loss.

Faithfulness

As discussed in previous chapter, faithfulness of an interpretation considers the question that if the prediction was indeed reliant on the features identified relevant by the interpretation. Note that this evaluation is lot more informative and useful when the model for prediction and interpretation are not same, as is the case with post-hoc interpretation. In principle, any model is faithful to itself by design. As we will discuss below, when using same model for prediction or interpretation in our framework, such as using g in by-design interpretation, it is significantly more straightforward to estimate faithfulness.

The common theme among most proposed metrics for raw attribution based methods is to modify the input according to attribution map and compare the classifiers new output with the original output on unchanged input. The interpretations in our framework are generated over the dictionary $\Phi_{\mathcal{I}}$. For any sample x, computing faithfulness would thus require to measure the effect of "removing" the attribute functions identified relevant in $L(x, g, f)$ on the prediction.

If g is being used for prediction (ante-hoc or by-design), this effect can be easily measured by setting $\phi_k(x) = 0, \forall k \in L(x, g, f)$ and computing output of g with new $\Phi_{\mathcal{I}}(x)$. Moreover, for simple designs of Θ, such as a linear layer with softmax, the relevance function practically guarantees that attribute functions truly utilized by the interpreter for its predicted class have high relevance. However, computing faithfulness for post-hoc interpretation is a challenge in general for concept based approaches. This is because $\Phi_{\mathcal{I}}(x)$ is not a part of computation of $f(x)$ and it is not obvious to measure how modifying $\Phi_{\mathcal{I}}$ affects $f(x)$. We explicitly address this problem for audio modality in chapter 5. One possible solution to measure this is to simulate how

modifying $\Phi_{\mathcal{I}}(x)$ can affect x. The design of decoder d proves essential in this regard as it can precisely be used to simulate this aspect. We specifically explore this aspect in chapter 5.

Interestingly, fidelity as a metric is related to faithfulness of g to f. This is because if both f and g rely on identical features with identical "reasoning" process for their output, it would imply their outputs are same for all inputs. In other words, complete faithfulness implies maximum fidelity. However the converse is not true, i.e. two models can predict the same output for all input samples but have different underlying decision processes. Nevertheless, this relationship indicates that fidelity can be viewed as a weaker form of faithfulness.

3.6 Conclusion

To summarize, in this chapter, we designed a single learning problem that can be utilized to solve post-hoc and by-design interpretation problems at both a local and global scope, in the context of neural networks. To accomplish this, we introduced a novel task Supervised Learning with Interpretation (SLI), which assumes prediction and interpretation as two separate tasks, addressed by dedicated but highly related models f (predictor) and g (interpreter) respectively. We outlined the design of our interpreter which is connected to the predictor through its selected hidden layers and relies on concept-based representations for interpretations. We also described the process of generating local and global interpretations for the interpreter. The precise designs of these modules are dependent on specific modalities they are employed for. We then discussed how the single SLI formulation can be used for different categories of interpretability problems. Finally, we discussed the different ways interpretations in the framework could be evaluated.

This chapter serves as the foundation for the content in the next two chapters wherein we instantiate this framework to tackle interpretability for image classification and audio classification.

Contents

4

Tackling Interpretability for Image Classification Networks

Contents

4.1 Introduction

In the previous chapter, we developed a framework to solve the SLI problem when the model to interpret is a deep neural network classifier. We now label the framework as 'FLINT' (Framework to Learn With INTerpretation). In this chapter, we instantiate FLINT when the input modality is images and the task is supervised image classification. To keep distinction between the image and audio interpretability systems (in Chapter 5), we will only refer to the image interpretability system as FLINT. Acting upon a theme highlighted in previous two chapters, our focus here is on learning a dictionary of concepts in an unsupervised setting as our representation for interpretation. The term 'unsupervised' refers here to not using any additional annotation

or supervision relevant to concept dictionary. The classification task is still solved as a supervised learning problem. Beyond improving upon the objective metrics for the task, we also aim to enhance the understandability of attribute functions by developing a pipeline to gain a deeper understanding what information is encoded by them.

The interpreter in FLINT implements the idea of understanding the prediction through decomposition in terms of attribute functions that encode high-level concepts as other approaches Alvarez-Melis and Jaakkola (2018a); Ghorbani et al. (2019). However, it enjoys two original key features. First the high-level attribute functions leverage the outputs of chosen hidden layers of the neural network. Second, together with expansion coefficients they are jointly learnt with the neural network to enable local and global interpretations. By local interpretation, we mean a subset of attribute functions whose simultaneous activation leads to the model's prediction, while by global interpretation, we refer to the description of each class in terms of a subset of attribute functions whose activation leads to the class prediction. Learning the pair of models involves the minimization of dedicated losses and penalty terms. In particular, local and global interpretability are enforced by imposing a limited number of attribute functions as well as conciseness and diversity among the activation of these attributes for a given input. Additionally, FLINT can be specialized to post-hoc interpretability if a pre-trained deep neural network is available. We summarize our **key contributions** in this chapter below

- We instantiate the framework developed in previous chapter for image classification task. We first propose a model for by-design interpretability based on learning a dictionary of concepts. it provides local and global interpretation using the novel notion of relevance. Eventually, a specialization of FLINT to post-hoc interpretability is presented.
- We propose a novel entropy based criterion for promoting conciseness and diversity in the learnt attribute functions and develop a simple pipeline to visualize the encoded concepts based on previously proposed tools.
- We present extensive experiments on 4 image classification datasets, MNIST, FashionMNIST, CIFAR10, QuickDraw, with a comparison with state-of-the-art approaches and a subjective evaluation study.

4.2 System design

4.2.1 Recap of SLI objective

The previous chapter introduces and discusses the generic task *Supervised Learning with Interpretation* (SLI) in detail. For completeness, we make a brief recap of the learning problem. Denoting $\mathcal{X}$ the input space of color/gray-scale images, and $\mathcal{Y}\{y \in \{0,1\}^C, \sum_{j=1}^{C} y^j = 1\}$ the output space, we assume that the training set $\mathcal{S} = \{(x_i, y_i)_{i=1}^N\}$ of size N is given for our supervised classification problem. SLI refers to the idea that

the **interpretation** task differs from the **prediction** task and must be taken over by a dedicated model that depends on the predictive model to be interpreted. Denoting $\mathcal{F}$ the space of predictive models from $\mathcal{X}$ to $\mathcal{Y}$. For a given model $f \in \mathcal{F}$, we denote $\mathcal{G}_f$ the family of models $g_f : \mathcal{X} \to \mathcal{Y}$, that depend on f and are devoted to its interpretation. For sake of simplicity, an interpreter $g_f \in \mathcal{G}_f$ is denoted g, omitting the dependency on f. The learning problem writes as:

Supervised Learning with Interpretation (SLI):

$$\textbf{Problem 1}: \arg \min_{f \in \mathcal{F}, g \in \mathcal{G}_f} \mathcal{L}_{pred}(f, \mathcal{S}) + \mathcal{L}_{int}(f, g, \mathcal{S}),$$

where $\mathcal{L}_{pred}(f, \mathcal{S})$ is the prediction loss term and $\mathcal{L}_{int}(f, g, \mathcal{S})$ is the interpretation loss term.

As previously, we set $\mathcal{F}$ to the class of deep neural networks with J hidden layers of respective dimension $j_1, \ldots, j_J$. Each element $f : \mathcal{X} \to \mathcal{Y}$ of $\mathcal{F}$ satisfies: $f = f_{J+1} \circ f_J \circ \ldots \circ f_1$ where $f_j : \mathbb{R}^{d_{j-1}} \to \mathbb{R}^{d_j}$, $d_0 = d, d_{J+1} = C, j = 1, \ldots, J+1$ is the function implemented by layer j. A network f in $\mathcal{F}$ is completely identified by its generic parameter V_f.

4.2.2 Interpreter for image modality

The interpreter model $g \in \mathcal{G}_f$, exploits the outputs of chosen hidden layers of f. Identical to chapter 3, the concatenated vector of the chosen intermediate outputs for an input sample x is denoted as $f_{\mathcal{I}}(x) \in \mathbb{R}^D$. Given f a network to be interpreted and a positive integer $K \in \mathbb{N}^*$, an **interpreter network** g computes the composition of a dictionary of attribute functions $\Phi_{\mathcal{I}} : \mathcal{X} \to \mathbb{R}_+^K$ and an interpretable function $\Theta : \mathbb{R}_+^K \to \mathcal{Y}$.

$$\forall x \in \mathcal{X}, g(x) = \Theta \circ \Phi_{\mathcal{I}}(x), \tag{4.1}$$

In this chapter, we take: $\Theta(\Phi_{\mathcal{I}}(x)) := \text{softmax}(W^T \Phi_{\mathcal{I}}(x))$ but other models like decision trees could be eligible. The **attribute dictionary** is composed of functions $\phi_k : \mathcal{X} \to \mathbb{R}^+, k = 1, \ldots K$, that is, $\Phi_{\mathcal{I}}(x) = [\phi_1(x), ..., \phi_K(x)]^T$./ For any element of the dictionary ϕ_k, its non-negative image, $\phi_k(x)$ can be interpreted as the activation of some high level attribute, i.e. a "concept" over $\mathcal{X}$. We propose to represent the encoded concept as a set of visual patterns in the **input space** which highly activate ϕ_k. The key for such learning lies in the fact that the attribute functions ϕ_k (referred to as attribute for simplicity) leverage the outputs of hidden layers of f specified by $\mathcal{I}$:

$$\forall k \in \{1, \ldots, K\}, \phi_k(x) = \psi_k \circ f_{\mathcal{I}}(x) \tag{4.2}$$

where each $\psi_k : \mathbb{R}^D \to \mathbb{R}_+$ operates on the accessed hidden layers. Here, the set of functions $\psi_k, k = 1, \ldots K$ is defined to form a shallow network Ψ (around 3 layers) whose output is $\Psi(f_{\mathcal{I}}(x)) = \Phi_{\mathcal{I}}(x)$ (example architecture in Fig. 4.1). Interestingly, ϕ_j are defined over $\mathcal{X}$ and as a consequence can be interpreted in the input space which is the most meaningful for the user (see Sec. 4.3). For sake of simplicity, we denote $V_g = (V_\Psi, V_\Theta)$ the specific parameters of this model, while the parameters devoted to the computation of $f_{\mathcal{I}}(x)$ are shared with f.

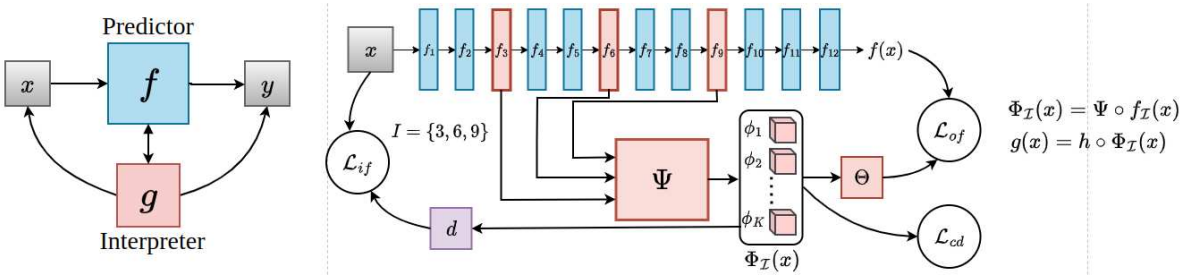


Figure 4.1: **(Left)** General view of SLI objective. **(Right)** Example instantiation of FLINT on a deep architecture for image classification.

4.2.3 Local and global relevances for interpretation

The interpreter being defined, we need to specify its expected role and corresponding interpretability objective. In FLINT, interpretation is seen as an additional task besides prediction. We are interested by two kinds of interpretation, one at the global level that helps to understand which attribute functions are useful to predict a class and the other at the local level, that indicates which attribute functions are involved in prediction of a specific sample. As a preamble, note that, to interpret a local prediction $f(x)$, we require that the interpreter output $g(x)$ matches $f(x)$. When the two models disagree, we provide a way to analyze the conflicting data and possibly raise an issue about the confidence on the prediction $f(x)$ (see appendix 4.4.4). To define local and global interpretation, we rely on the notion of relevance of an attribute.

Given an interpreter with parameter $V_g = (V_\Psi, V_\Theta)$ and some input x, the **relevance score** of an attribute ϕ_j is defined regarding the prediction $g(x) = f(x) = \hat{y}$. Denoting $\hat{y} \in \mathcal{Y}$ the index of the predicted class and $w_{j,\hat{y}} \in W$ the coefficient associated to this class, the contribution of attribute ϕ_j to unnormalized score of class $\hat{y}$ is $\alpha_{j,\hat{y},x} = \phi_j(x).w_{j,\hat{y}}$. The relevance score is computed by normalizing contribution α as $r_{j,x} = \frac{\alpha_{j,\hat{y},x}}{\max_i |\alpha_{i,\hat{y},x}|}$. An attribute ϕ_j is considered as relevant for a local prediction if it is both activated and effectively used in the linear (logistic) model. The notion of relevance of an attribute for a sample is extended to its "overall" importance in the prediction of any class c. This can be done by simply averaging relevance scores from local interpretations over a random subset or whole of the training set $\mathcal{S}$, where predicted class is c. Thus, we have: $r_{j,c} = \frac{1}{|\mathcal{S}_c|} \sum_{x \in \mathcal{S}_c} r_{j,x}$, $\mathcal{S}_c = \{x \in \mathcal{S} | \hat{y} = c\}$. Now, we can introduce the notions of local and global interpretations that the interpreter will provide.

Definition 4.1 (Global and Local Interpretation)**.** *For a prediction network f, the* **global interpretation** *$G(g, f)$ provided by an interpreter g, is the set of class-attribute pairs (c, ϕ_j) such that their global relevance $r_{j,c}$ is greater than some threshold $1/\tau, \tau > 1$. A* **local interpretation** *for a sample x provided by an interpreter g of f denoted $L(x, g, f)$ is the set of attribute functions ϕ_j with local relevance score $r_{j,x}$ greater than some threshold $1/\tau, \tau > 1$.*

It is important to note that these definitions do not prejudge the quality of local and global interpretations. Next, we convert desirable properties of the interpreter into specific loss functions.

4.2.4 Learning by imposing interpretability properties

The previous chapter introduced the minimal set of properties and loss functions we use to formulate the interpretability objective $\mathcal{L}_{int}$. We introduce a novel criterion based on minimizing entropy to impose conciseness. The remaining two properties remain virtually unchanged. The details about each property and loss function are summarized below:

Fidelity to Output. The output of the interpreter $g(x)$ should be "close" to $f(x)$ for any x. This can be imposed through a cross-entropy loss:

$$\mathcal{L}_{of}(f, g, \mathcal{S}) = -\sum_{x \in \mathcal{S}} \Theta(\Psi(f_{\mathcal{I}}(x)))^T \log(f(x))$$

Conciseness and Diversity of Interpretations. For any given sample x, we wish to get a *small* number of attributes in its associated local interpretation. This property of *conciseness* should make the interpretation easier to understand due to fewer attributes to be analyzed and promote the "high-level" character in the encoded concepts. The common approach to impose sparsity is through ℓ_1 regularization. Here instead we opt for incorporating entropy minimization to impose sparsity (Huang and Tran, 2018). However, this can strongly constrain activation of attributes. To encourage better use of available attributes we also expect activation of multiple attributes across many randomly selected samples. We refer to this property as *diversity*. This is also important to avoid the case of attribute functions being learnt as class exclusive (for eg. reshuffled version of class logits). To enforce these conditions we utilize notion of entropy defined for real vectors proposed by Jain et al. (2017) to solve problem of efficient image search. For a real-valued vector v, the entropy is defined as $\mathcal{E}(v) = -\sum_i p_i \log(p_i)$, $p_i = \exp(v_i)/(\sum_i \exp(v_i))$.

Conciseness is promoted by minimizing $\mathcal{E}(\Psi(f_{\mathcal{I}}(x)))$ and diversity is promoted by maximizing entropy of average $\Psi(f_{\mathcal{I}}(x))$ over a mini-batch. Note that this can be seen as encouraging the interpreter to find a sparse and diverse coding of $f_{\mathcal{I}}(x)$ using the function Ψ. Since entropy-based losses have inherent normalization, they do not constrain the magnitude of the attribute activation. This often leads to poor optimization. Thus, we also minimize the ℓ_1 norm $\|\Psi(f_{\mathcal{I}}(x))\|_1$ (with hyperparameter η) to avoid it. Note that ℓ_1-regularization is a common tool to encourage sparsity and thus conciseness, however we show in the experiments that entropy provides a more effective way.

$$\mathcal{L}_{cd}(f, g, \mathcal{S}) = -\mathcal{E}(\bar{\Phi}_{\mathcal{S}}) + \sum_{x \in \mathcal{S}} \mathcal{E}(\Psi(f_{\mathcal{I}}(x))) + \sum_{x \in \mathcal{S}} \eta \|\Psi(f_{\mathcal{I}}(x))\|_1, \quad \bar{\Phi}_{\mathcal{S}} = \frac{1}{|\mathcal{S}|} \sum_{x \in \mathcal{S}} \Psi(f_{\mathcal{I}}(x))$$

Fidelity to Input. To encourage encoding high-level patterns or meaningful factors of variation related to input in $\Phi_{\mathcal{I}}(x)$, we use a decoder network $d : \mathbb{R}_+^K \rightarrow \mathcal{X}$ that takes as input the dictionary of attributes $\Psi(f_{\mathcal{I}}(x))$ and reconstructs x. A similar penalty has previously been applied by Alvarez-Melis and Jaakkola (2018a).

$$\mathcal{L}_{if}(f, g, d, \mathcal{S}) = \sum_{x \in \mathcal{S}} (d(\Psi(f_{\mathcal{I}}(x))) - x)^2$$

Note that one can modify $\mathcal{L}_{if}$ with other reconstruction losses as well (such as ℓ_1-reconstruction).

Given the proposed loss terms, the loss for interpretability writes as follows:

$$\mathcal{L}_{int}(f, g, d, \mathcal{S}) = \beta \mathcal{L}_{of}(f, g, \mathcal{S}) + \gamma \mathcal{L}_{if}(f, g, d, \mathcal{S}) + \delta \mathcal{L}_{cd}(f, g, \mathcal{S})$$

where β, γ, δ are non-negative hyperparameters. The total loss to be minimized $\mathcal{L} = \mathcal{L}_{pred} + \mathcal{L}_{int}$, where the prediction loss, $\mathcal{L}_{pred}$, is the cross-entropy loss.

Let us denote $V = (V_f, V_d, V_\Psi, V_\Theta)$ the parameters of these networks. Learning the models f, Ψ, h and d boils down to learning V. In practice, introducing all the losses at once often leads to very poor optimization. Thus, we follow the procedure described in Alg. 4.1. We train the networks with $\mathcal{L}_{pred}, \mathcal{L}_{if}$ for the first two epochs and gain a reasonable level of accuracy. From the third epoch we introduce $\mathcal{L}_{of}$ and from the fourth epoch we introduce $\mathcal{L}_{cd}$ loss.

Algorithm 4.1 Learning algorithm for FLINT

1: **Input:** $\mathcal{S}$ & parameters $V = (V_f, V_d, V_\Psi, V_\Theta)$ & hyperparameters: $\beta_0, \gamma_0, \delta_0, \eta_0$ & number of batches B & number of training epochs N_{epoch}.
2: Random initialization of parameter V_0
3: $V_1 \leftarrow$ Train $(\mathcal{S}, V_0, \beta = 0, \gamma_0, \delta = 0, \eta = 0, B, 2)$ {% Trains 2 epochs with $\mathcal{L}_{pred}, \mathcal{L}_{if}$}
4: $V_2 \leftarrow$ Train $(\mathcal{S}, V_1, \beta = \beta_0, \gamma_0, \delta = 0, \eta = 0, B, 1)$ {% Trains 1 epoch with $\mathcal{L}_{pred}, \mathcal{L}_{if}, \mathcal{L}_{of}$}
5: $\hat{V} \leftarrow$ Train $(\mathcal{S}, V_2, \beta_0, \gamma_0, \delta_0, \eta_0, B, N_{epoch} - 3)$ {% Trains with all losses}
6: **Output:** $\hat{V} = (\hat{V}_f, \hat{V}_d, \hat{V}_\Psi, \hat{V}_\Theta)$

4.3 Understanding encoded concepts in FLINT

Once the predictor and interpreter are jointly learnt, interpretation can be given at the global and local levels as in Def. 4.1. A key component to grasp the interpretations is to understand/discover the concept encoded by each individual attribute function ϕ_k, previously defined in Eq. 4.2. For images, as previously mentioned, we represent an encoded concept as a set of visual patterns in the **input space** which highly activate ϕ_k. In other words, our aim is to understand/discover the encoded concept by visualizing points from the input space highly activating $\phi_k(x)$. To do so, one can build up

from the generated global relevances and visualize the encoded information by the attribute for each class it is relevant for, that is, we can start by visualizing relevant class-attribute pairs. We present below a pipeline to generate visualizations for global and local interpretation by adapting various previously proposed tools Alvarez-Melis and Jaakkola (2018a); Mahendran and Vedaldi (2016).

Algorithm 4.2 Visualization of global interpretation

1: **Input:** (class,attribute):(c, ϕ_j) & subset size:l & training set:$\mathcal{S}_n$ & AM+PI params:$(\lambda_\phi, \lambda_{tv}, \lambda_{bo})$
2: $\mathcal{S}_c = \{x | (x, c) \in \mathcal{S}_n\}$
3: $\text{MAS}(c, \phi_j, l) \leftarrow \arg\max_{\mathcal{M} \subset \mathcal{S}_c, |\mathcal{M}|=l} \sum_{x_i \in \mathcal{M}} \phi_j(x)$
4: FOR $x_k \in \text{MAS}(c, \phi_j, l)$
5: $x^k_{vis} \leftarrow \text{AM+PI}(x_k, \lambda_\phi, \lambda_{tv}, \lambda_{bo})$
6: ENDFOR
7: **Output:** $\{x^1_{vis}, \ldots, x^l_{vis}\}$, $\text{MAS}(c, \phi_j, l)$

Visualization of global interpretation. Given any class-attribute pair (c, ϕ_k) in the global interpretation $G(g, f)$, we first select a small subset of training samples from class c that maximally activate ϕ_k. This set of samples is referred to as maximum activating samples and denoted $\text{MAS}(c, \phi_k, l)$ where l is the size of the subset (chosen as 3 in the experiments). Although, MAS reveal some information about the encoded concept, it might not be apparent "what" aspect of these samples causes activation of ϕ_k. We thus propose further analyzing each element in MAS through tools that enhance the detected concept. This results in a much better understanding. The primary tool we employ is a modified version of activation maximization Mahendran and Vedaldi (2016), which we refer to as **activation maximization with partial initialization (AM+PI)**.

Given a maximum activating sample $x' \in \text{MAS}(c, \phi_j, l)$, the key idea behind AM+PI is to synthesize appropriate input via optimization, that maximally activates ϕ_j under regularization constraints:

$$\arg\max_x \lambda_\phi \phi_j(x) - \lambda_{tv} \text{TV}(x) - \lambda_{bo} \text{Bo}(x)$$

where $\text{TV}(.), \text{Bo}(.)$ are regularization terms to encourage lower total variation (for smoothness) and better pixel value boundedness respectively. However, we initialize the procedure by low-intensity version of sample x'. This makes the optimization easier with the detected concept weakly present in the input. This also allows the optimization to "fill" the input to enhance the encoded concept. As an output, we obtain a map adapted to x', that strongly activates ϕ_j. As mentioned previously this pipelines is primarily designed to understand "what" concept is encoded by an attribute. Visualization of a class-attribute pair is summarized in Alg. 4.2. There is a potential to utilize alternate tools for visualization such as saliency maps or the decoder. However, for a simpler discussion, we limit to use of AM+PI outputs in this chapter and defer discussion about other tools to appendix G.1.3.

Local analysis. Given any test sample x_0, one can determine its local interpretation $L(x_0, f, g)$, the set of relevant attribute functions accordingly to Def. 4.1. To visualize a relevant attribute $\phi_j \in L(x_0, f, g)$, we can repeat the AM+PI procedure with initialization using low-intensity version of x_0 to enhance concept detected by ϕ_j in x_0. Note that the understanding built about any attribute function ϕ_j via global analysis, although not essential, can still be helpful to understand the generated AM+PI maps during local analysis, as these maps are generally similar.

4.4 Experiments for FLINT

Datasets. We consider 4 primary datasets for experiments, MNIST LeCun et al. (1998), FashionMNIST Xiao et al. (2017), CIFAR-10 Krizhevsky et al. (2009), and a subset of QuickDraw dataset Ha and Eck (2018). We created a subset of QuickDraw from the original dataset Ha and Eck (2018), by selecting 10000 random images from each of 10 classes: 'Ant', 'Apple', 'Banana', 'Carrot', 'Cat', 'Cow', 'Dog', 'Frog', 'Grapes', 'Lion'. We randomly divide each class into 8000 training and 2000 test images. Additional results on CIFAR100 Krizhevsky et al. (2009) (large number of classes) and Caltech-UCSD Birds-200-2011 Wah et al. (2011) (large-scale images and large number of classes) are covered later in section 4.5.

Networks architecture. Our experiments include 2 kinds of architectures for predictor f: (i) LeNet-based LeCun (2015) network for MNIST, FashionMNIST, and (ii) ResNet18-based He et al. (2016) network for QuickDraw, CIFAR. We select one intermediate layer for LeNet based network and two for ResNet based networks, from the last few convolutional layers as they are expected to capture higher-level features. We set the number of attributes $K = 25$ for MNIST, FashionMNIST, $K = 24$ QuickDraw and $K = 36$ for CIFAR. Complete details about architecture and optimization are available in appendix G.1.1.

4.4.1 Quantitative evaluation of FLINT

We evaluate and compare our model with other state-of-the-art systems regarding accuracy and interpretability. The evaluation metrics for interpretability Doshi-Velez and Kim (2017) are defined to measure the effectiveness of the losses proposed in Sec. 3.3.3. Our primary method for comparison, wherever applicable, is SENN, as it is an interpretable network by design with same units for interpretation as FLINT. Other baselines include PrototypeDNN Li et al. (2018) for predictive performance, LIME Ribeiro et al. (2016) and VIBI Bang et al. (2021) for fidelity of interpretations. Implementation of our method is available on Github [1]. Details for implementation of baselines are available in appendix G.1.5.

Predictive performance of FLINT. There are three goals to validate related to performances of models trained with FLINT (denoted FLINT-f and FLINT-g), (i) Jointly training f with g and backpropagating loss term $\mathcal{L}_{int}$ does not negatively impact per-

[1] https://github.com/jayneelparekh/FLINT

	BASE-f	SENN	PrototypeDNN	FLINT-f	FLINT-g	BASE-g
MNIST	98.9±0.1	98.4±0.1	**99.2**	98.9±0.2	98.3±0.2	97.9±0.3
FashionMNIST	90.4±0.1	84.2±0.3	90.0	**90.5±0.2**	86.8±0.4	84.5±0.5
CIFAR10	84.7±0.3	77.8±0.7	–	**84.5±0.2**	84.0±0.4	81.9±0.3
QuickDraw	85.3±0.2	85.5±0.4	–	**85.7±0.3**	85.4±0.1	83.9±0.2

Table 4.1: Accuracy (in %) on different datasets. BASE-f is system trained with just accuracy loss. FLINT-f, FLINT-g denote the predictor and interpreter trained in our framework. Mean and standard deviation of 4 runs for each system are reported

Dataset	LIME	VIBI	FLINT-g
MNIST	95.6±0.4	96.6±0.7	**98.7±0.1**
FashionMNIST	67.3±1.3	88.4±0.3	**91.5±0.1**
CIFAR-10	31.5±0.9	65.5±0.3	**93.2±0.2**
QuickDraw	76.3±0.1	78.6±0.4	**90.8±0.4**

Table 4.2: Results for fidelity of FLINT-g to FLINT-f (in %). Mean and standard deviation of 4 runs for each system are reported.

formance of f, (ii) The achieved performance is comparable with other similar interpretable by-design models, and (iii) Performance of FLINT-g is better compared to training interpreter by directly applying a prediction loss to $g(x)$ and not using $\mathcal{L}_{pred}, \mathcal{L}_{of}$. Regardless of if FLINT-f or FLINT-g is used as the final prediction model, the above three goals cover all the key questions regarding performance. For (i) we compare the accuracy of FLINT-f with same predictor architecture trained just with $\mathcal{L}_{pred}$ (denoted by BASE-f). For (ii), we compare accuracy of FLINT-f with accuracy of SENN and another interpretable network by design PrototypeDNN Li et al. (2018) that does not use input attribution for interpretations. Note that PrototypeDNN requires non-trivial changes to the model for running on more complex datasets, CIFAR10 and QuickDraw. To avoid any unfair comparison we skip these results. Finally for (iii), we compute the performance of the training variant of the interpreter, denoted as BASE-g.

The accuracies are reported in Tab. 4.1. They indicate that training f within FLINT does not result in any significant accuracy loss on any dataset. FLINT-f and FLINT-g are both competitive or better than other interpretable by-design models in terms of performance. The training variant of interpreter BASE-g also performs worse than FLINT-g. This highlights that even when using $g(x)$ for final prediction, updating hidden layers of f with prediction loss on $f(x)$ improves performance of g.

Fidelity of Interpreter. The fraction of samples where prediction of a model and its interpreter agree, i.e predict the same class, is referred to as *fidelity*. It is a commonly used metric to measure how well an interpreter approximates a model Bang et al. (2021); Lakkaraju et al. (2020). Note that, typically, for interpretable by design models, fidelity cannot be measured as they only consider a single model. However,

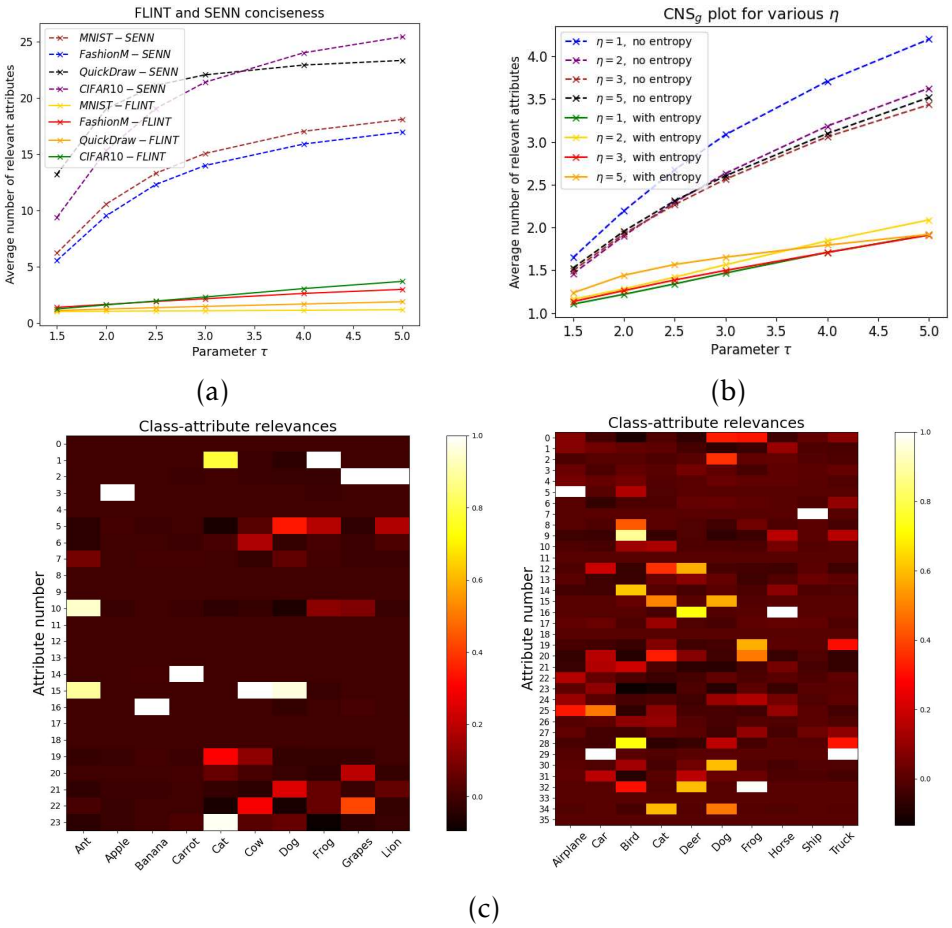


Figure 4.2: (a) Conciseness comparison of FLINT and SENN. (b) Effect of entropy losses on conciseness of ResNet for QuickDraw for various ℓ_1-regularization levels. (c) Global class-attribute relevances $r_{k,c}$ for QuickDraw (Left) and CIFAR10 (Right). 24 class-attribute pairs for QuickDraw and 32 pairs for CIFAR10 have relevance $r_{k,c} > 0.2$.

to validate that the interpreter trained with FLINT (denoted as FLINT-g) achieves a reasonable level of agreement with FLINT-f, we benchmark its fidelity against a state-of-the-art black-box explainer VIBI Bang et al. (2021) and a traditional method LIME Ribeiro et al. (2016). The results for this are provided in Tab. 4.2 (last three columns). FLINT-g consistently achieves higher fidelity. Even though it is not a fair comparison as other systems are black-box interpreters and FLINT-g accesses intermediate layers, they clearly highlight that use of hidden layers is a key for FLINT-g to achieve high fidelity to FLINT-f.

Conciseness of interpretations. We evaluate conciseness by measuring the average number of *important* attributes in generated interpretations. For a given sample x, it can be computed as number of attributes ϕ_j with $r_{k,x}$ greater than a threshold $1/\tau, \tau > 1$, i.e. $\text{CNS}_{g,x} = |\{k : |r_{k,x}| > 1/\tau\}|$. For different thresholds $1/\tau$, we compute the mean of $\text{CNS}_{g,x}$ over test data to estimate conciseness of g, CNS_g. Lower conciseness indicates

need to analyze a lower number of attributes on an average. SENN is the only other system for which this curve can be computed. We thus compare the conciseness of SENN with FLINT on all four datasets. Fig. 4.2a depicts the same. It can be easily observed that FLINT produces lot more concise interpretations compared to SENN. Moreover, SENN even ends up with majority of concepts being considered relevant for lower thresholds (higher τ).

Entropy vs ℓ_1 regularization. We validate the effectiveness of entropy losses by computing conciseness curve at various levels of ℓ_1 regularization strength, with and without entropy, for ResNet with QuickDraw. This is reported in Fig. 4.2b. The figure confirms that using the entropy-based loss is more effective in inducing conciseness of explanations compared to using just ℓ_1-regularization, with the difference being close to use of 1 attribute less when entropy losses are employed.

Importance of attributes. By structure, for both FLINT-g and SENN, the output are generated by combining high level attributes and weights. To test how crucial the learnt attributes are to their predictions, we shuffle the attribute values $\Phi(x)$ for each sample x (this corresponds to shuffling $h(x)$ for SENN with their notations). This is an extreme test: we therefore expect an important drop in accuracy. Tab. 4.3 reports the results for the experiments for our method and SENN. More precisely, we calculate the drop in prediction accuracy of FLINT-g (and SENN), compared to their mean accuracies. For SENN, the very small drop in accuracy indicates its robustness to this shuffling, which highlights the fact that in this model, the weights generated by the model are more crucial for prediction than the attributes. In contrast FLINT-g relies strongly on its attributes for its prediction.

Dataset	SENN	FLINT-g
MNIST	0.5	87.6
FashionMNIST	10.9	76.6
CIFAR-10	17.5	74.4
QuickDraw	0.3	74.9

Table 4.3: FLINT and SENN accuracy drop for shuffled attributes (in %)

4.4.2 Qualitative analysis

Global interpretation. Fig. 4.2c depicts the generated global relevances $r_{k,c}$ for all class-attribute pairs on QuickDraw and CIFAR. Each class-attribute pair with 'high' relevance needs to be analyzed as part of global analysis. Some example class-attribute pairs, with high relevance, are visualized in Fig. 4.3. For each pair we select MAS of size 3 and also show their AM+PI outputs. As mentioned before, simply analyzing MAS reveals useful information about the encoded concept. For instance, based on MAS, ϕ_{15}, ϕ_{19} on MNIST, relevant for class 'One', clearly seem to activate for vertical and diagonal strokes respectively. However, AM+PI outputs give deeper insights about the concept by revealing more clearly what parts of input activate an

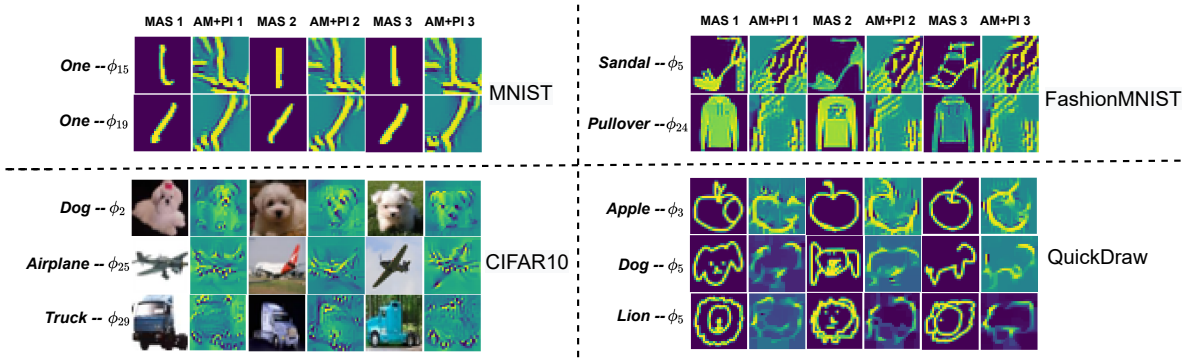


Figure 4.3: Example class-attribute pair analysis on all datasets, with global relevance $r_{k,c} > 0.2$. Each row contains 3 MAS with corresponding AM+PI outputs

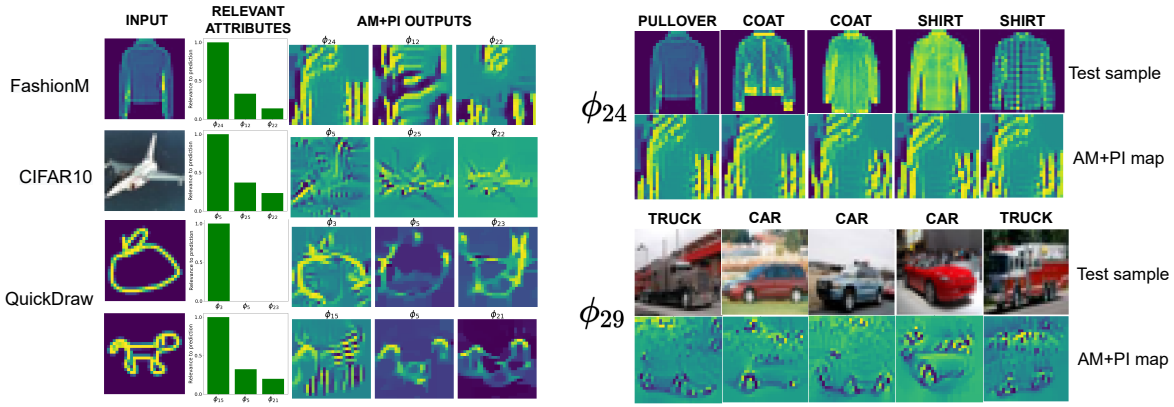


Figure 4.4: **(Left)** Local interpretations for test samples. Top 3 attributes with corresponding AM+PI output are shown. True labels for inputs are: Pullover, Airplane, Apple, Dog. **(Right)** Examples of attribute functions detecting same part across various test samples. For each sample, their relevance is greater than 0.8. True labels of samples indicated above them.

attribute function. For eg., while MAS indicate that ϕ_5 on FashionMNIST activates for heels (one type of 'Sandal'), ϕ_2 on CIFAR10 activates for white dogs, it is not clear what part the attribute focuses on. AM+PI outputs indicate that ϕ_2 focuses on the area around eyes and nose (the most enhanced regions), ϕ_5 primarily detects a thin diagonal stroke of the heel surrounded by empty space. AM+PI outputs generally become even more important for attributes relevant for multiple classes. One such example is the function ϕ_5 on QuickDraw, relevant for both 'Dog' and 'Lion'. It activates for very similar set of strokes for all samples, as indicated by AM+PI maps. For 'Dog' this corresponds to ears and mouth and for 'Lion' it corresponds to the mane. Other such attribute functions in the figure include ϕ_{24} on FashionMNIST, relevant for 'Pullover', 'Coat' and 'Shirt' which detects long sleeves and ϕ_{29} on CIFAR10, relevant for 'Trucks', 'Cars' and primarily detects wheels and parts of upper body. Further visualizations including those of other relevant classes for ϕ_{24}, ϕ_{29} and global relevances are available in supplementary (Sec. S.2).

Local interpretation. Fig. 4.4 (left) displays the local interpretation visualizations for test samples. f and g both predict the true class in all the cases. We show the top 3 relevant attributes to the prediction with their relevances and their corresponding AM+PI outputs. Based on the AM+PI outputs it can be observed that the attribute functions generally activate for patterns corresponding to the same concept as inferred during global analysis. This can be easily seen for attribute functions present in both Fig. 4.3, 4.4 (left). This is further illustrated by Fig. 4.4 (right) where we illustrate AM+PI outputs for two attributes from Fig. 4.3. These functions are relevant for more than one class and detect the same concept across various test samples, namely long sleeves for ϕ_{24} and primarily wheels for ϕ_{29}.

4.4.3 Subjective evaluation

We conducted a *survey based subjective evaluation* with QuickDraw dataset for FLINT with 20 respondents to assess the understandability of our concept discovery pipeline. We selected 10 attributes, covering 17 class-attribute pairs from the QuickDraw dataset. For each attribute we present the respondent with our visualizations (3 MAS and AM+PI outputs) for each of its relevant classes along with a textual description. We ask them if the description meaningfully associates to patterns in the AM+PI outputs. They indicate level of agreement with choices: Strongly Agree (SA), Agree (A), Disagree (D), Strongly Disagree (SD), Don't Know (DK). Descriptions were manually generated by our understanding of encoded concept for each attribute. 40% incorrect descriptions were carefully included to ensure informed responses. These were forcefully related to the classes shown to make them harder to identify.

Results – for correct descriptions: 77.5% – SA/A, 10.0% – DK, 12.5% – D/SD. For incorrect descriptions: 83.7% – D/SD, 7.5% – DK, 8.8% – SA/A. These results clearly indicate that concepts encoded in FLINT's learnt attributes are understandable to humans. The form taken by the participants can be accessed here [2]. Further details about the survey are provided in appendix G.1.6.

4.4.4 Disagreement analysis

In this part, we analyse in detail the "disagreement" between the predictor f and the interpreter g. Note that we already achieve very high fidelity to predictor for all datasets. We limit our analysis to QuickDraw, our dataset with least fidelity. Understanding disagreement can help us improving our framework as well as providing a measure of reliability about predictors output.

For a given sample with disagreement, if the class predicted by f is among the top predicted classes of g, the disagreement is acceptable to some extent as the attributes can still potentially interpret the prediction of f. The worse kind of samples for disagreement are the ones where class predicted by f is not among the top predicted classes of g, and even worse are where, in addition to this, f predicts the true label.

[2] https://forms.gle/PW6DEPZSmXb46Lnv9

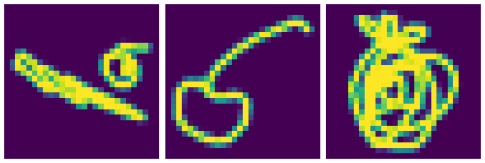


Figure 4.5: The three 'Apple' class samples classified correctly by f but not by g.

We thus compute the top-k fidelity (for $k = 2, 3, 4$) on QuickDraw with ResNet, which for the default parameters described in the main paper, achieves a top-2 fidelity of 94.7%, top-3 fidelity 96.9%, and top-4 fidelity 98.2%. Only on 141 (i.e. 0.7%) samples the class predicted by f, same as true class, is not in top-3 predicted by g classes.

For eg., for the 'Apple' class (in QuickDraw), there only three disagreement samples for which f delivers correct prediction (plotted in Fig. 4.5) are not resembling apples at all. We propose an original analysis approach that consists in calculating a *robust centrality measure*—the projection depth—of these three samples as well as of another 100 training samples w.r.t. the 8000 training 'Apple' samples, plotted in Fig. 4.6. To that purpose, we use the notion of projection depth (Zuo and Serfling, 2000; Mosler, 2013) for a sample $\boldsymbol{x} \in \mathbb{R}^d$ w.r.t. a dataset $\boldsymbol{X}$ which is defined as follows:

$$D(\boldsymbol{x}|\boldsymbol{X}) = \left(1 + \sup_{\boldsymbol{p}\in\mathcal{S}^{d-1}} \frac{|\langle \boldsymbol{p}, \boldsymbol{x}\rangle - \mathrm{med}(\langle \boldsymbol{p}, \boldsymbol{X}\rangle)|}{\mathrm{MAD}(\langle \boldsymbol{p}, \boldsymbol{X}\rangle)}\right)^{-1}, \tag{4.3}$$

with $\langle \cdot, \cdot \rangle$ denoting scalar product (and thus $\langle \boldsymbol{p}, \boldsymbol{X}\rangle$ being a vector of projection of $\boldsymbol{X}$ on $\boldsymbol{p}$) and med and MAD being the univariate median and the median absolute deviation form the median. Fig. 4.6 confirms the visual impression that these 3 disagreement samples are outliers (since their depth in the training class is low).

Fig. 4.7 depicts 26 such cases for 'Cat' class to illustrate their logical dissimilarity. Being a complex model, the ResNet-based predictor f still manages to learn to distinguish these cases (while g does not), but in a way g does not manage at all to explain. Eventually, exploiting disagreement of f and g could be used as a means to measure trustworthiness. Deepening this issue is left for future works.

4.4.5 Ablation studies

Effect of hidden layer selection

In general for any predictor architecture or dataset, the most obvious choice is to select last convolutional layer output. This also helps achieving high fidelity for g. The only problem that might arise when selecting layer(s) very close to the output is that the attribute might be learnt trivially. This is indicated by extremely low entropy and high input fidelity loss. While tuning hyperparameters of interpretability loss could be helpful in tackling this issue (reducing β, increasing γ), choosing an earlier hidden layer can also prove to be very useful. We study the effect of choice of hidden layers with ResNet18 on QuickDraw. We make 3 different choices of single hidden

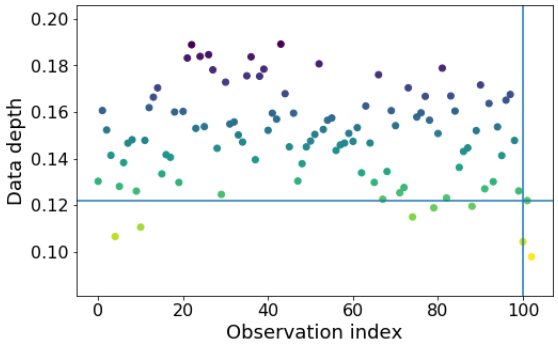


Figure 4.6: Projection data depth calculated with (4.3) w.r.t. the 8000 'Apple' training sample for 100 'Apple' test samples and for the three (observation indices 101–103) 'Apple' class samples classified correctly by f but not by g.

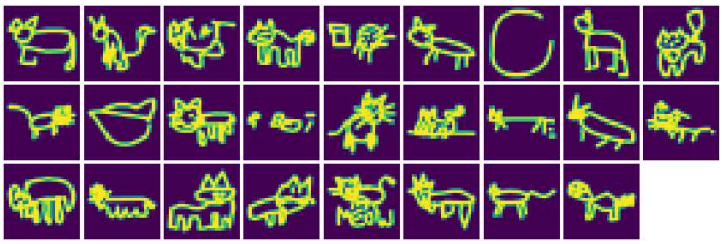


Figure 4.7: 26 samples from 'Cat' class which are not in top3 f-predicted classes.

layers (9th, 13th, 16th conv layers). For each choice we tabulate resulting metrics (accuracy, fidelity of interpreter, reconstruction loss, conciseness for threshold $1/\tau = 0.2$) in Tab. 4.4. All other hyperparameters remain same.

Layer	Accuracy (in %)	Fidelity (in %)	$\mathcal{L}_{if}$	Conciseness $1/\tau = 0.2$
9th conv	85.2	78.0	0.074	1.873
13th conv	85.6	85.6	**0.073**	1.905
16th conv	86.5	**96.0**	0.081	**1.562**

Table 4.4: Effect of different hidden layers for Resnet18 on QuickDraw.

Key observations: (a) Compared to average BASE-f accuracy of 85.3% for ResNet18 on QuickDraw, accuracy of all models are comparable or slightly better. Thus, choice of hidden layers does not strongly affect predictor accuracy. (b) The interpreter fidelity gets considerably better if the layer chosen is closer to the output. (c) The input fidelity/reconstruction loss does not behave as monotonously, but it is not surprising that layers close to the output result in worse input reconstruction. (d) Interpretations are expected to be more concise when chosen layer is very close to the output in the sense that conciseness is an indicator of abstraction level of the interpretation. Thus, a standard choice is to start with a layer close to the output. A small revision may be needed depending upon optimization of input fidelity loss.

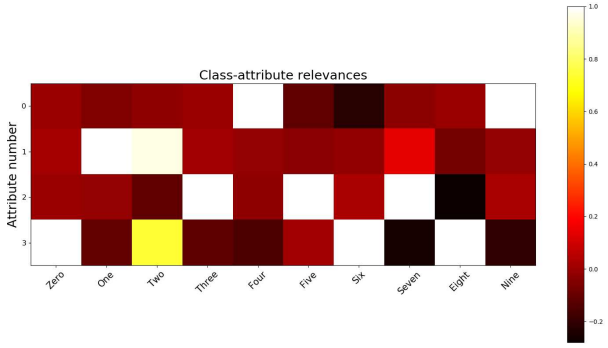


Figure 4.8: Global class attribute relevances for model with $K = 4$ on MNIST.

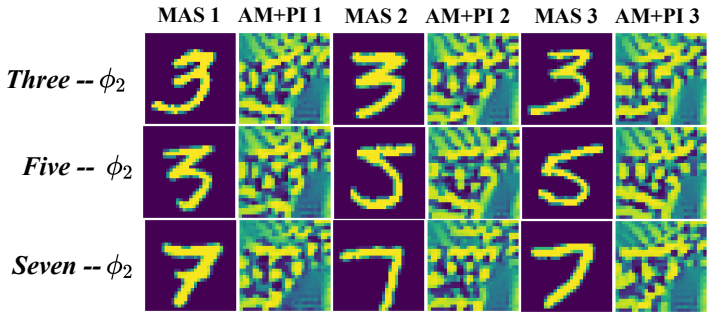


Figure 4.9: Interpretation for attribute ϕ_2 for model learn on MNIST with $K = 4$.

	$\mathcal{L}_{if}$ (train)	$\mathcal{L}_{of}$ (train)	Fidelity (test) (%)
$K = 4$	0.058	0.57	87.4
$K = 8$	0.053	0.23	97.5
$K = 25$	0.029	0.16	98.8

Table 4.5: Effect of K on losses and fidelity for MNIST with LeNet.

Effect of K We study the effect of choosing small values for number of attributes K (keeping all other hyperparameters same). Tab. 4.5 tabulates the values of input fidelity loss $\mathcal{L}_{if}$, output fidelity loss $\mathcal{L}_{of}$ on the training data by the end of training for MNIST and the fidelity of g to f on MNIST test data for different K values. Tab. 4.6 tabulates same values for QuickDraw. The two tables clearly show that using small K can harm the autoencoder and the fidelity of interpreter. Moreover, the system packs more information in each attribute and this makes it hard to understand them, specially for very small K. This is illustrated in Figs. 4.8 and 4.9, which depict part of global interpretations generated on MNIST for $K = 4$ (all the parameters take default values). Fig. 4.8 shows global class-attribute relevances and Fig. 4.9 shows generated interpretation for a sample attribute ϕ_2. It can be clearly seen that the attributes start encoding concepts for too many classes (high number of bright spots). This also causes their AM+PI outputs to be muddled with two many patterns. This adds a lot of difficulty in understandability of these attributes.

	$\mathcal{L}_{if}$ (train)	$\mathcal{L}_{of}$ (train)	Fidelity (test) (%)
$K = 4$	0.094	2.08	19.5
$K = 8$	0.079	1.48	57.6
$K = 24$	0.069	0.34	90.8

Table 4.6: Effect of K on losses and fidelity for QuickDraw with ResNet.

How to choose the number of attributes K Assuming a suitable architecture for decoder d, simply tracking $\mathcal{L}_{if}, \mathcal{L}_{of}$ on training data can help rule out very small values of K as they result in poorly trained decoder and relatively poor fidelity of g. One can also qualitatively analyze the generated explanations from the training data to tune K to a certain extent. Too small values of K can result in attributes encoding concepts for too many classes, which affects negatively their understandability. It is more tricky and subjective to tune K once it becomes large enough so that $\mathcal{L}_{if}, \mathcal{L}_{of}$ are optimized well. The upper threshold of choosing K is subjective and highly affected by how many attributes the user can keep a tab on or what fidelity user considers reasonable enough. It is possible that due to enforcement of conciseness, even for high value of K, only a small subset of attributes are relevant for interpretations. Nevertheless, for high K value, there is a risk of ending up with too many attributes or class-attribute pairs to analyze.

It is important to notice that it is possible to select K from the training set only by using a cross-validation strategy. In practice, it seems reasonable to agree on smallest value of K for which the increase of the cross-validation fidelity estimate drops dramatically, since further increase of K would generate less understandable attributes with very little gain in fidelity.

4.5 Experiments on CIFAR-100 and CUB-200

We also demonstrate the ability of the system to handle more complex datasets by experimenting with CIFAR100 Krizhevsky et al. (2009) and Caltech-UCSD-200 (CUB-200) fine-grained Bird Classification dataset Wah et al. (2011). CIFAR100 contains 100 classes with 500 training and 100 testing samples per class (image size 32×32×3). CUB-200 contains 11,788 images of 200 categories of birds, 5,994 for training and 5,794 for testing. We scale each sample in CUB-200 to size $224 \times 224 \times 3$. We also don't crop using the bounding boxes and use the full images for training and testing.

Compared to our earlier experiments, we make two key changes to the framework, (i) Increase size of dictionary of attribute functions to accommodate larger images/number of classes, (ii) Modify architecture of decoder d with more upsampling and convolutional layers. For CIFAR100, the same architectures for f and g as on CIFAR10 is used, but with $K = 72$. We apply random horizontal flip as additional augmentation and train for 51 epochs. For CUB-200, we use the ResNet18 He et al. (2016)

	Accuracy (in %)			Fidelity (in %)	
Dataset	BASE-f	FLINT-f	FLINT-g	Top-1	Top-5
CIFAR100	70.7	70.8	69.9	85.2	97.3
CUB-200	71.3	71.0	68.7	80.0	96.7

Table 4.7: Results for accuracy (in %) and fidelity to FLINT-f on CIFAR100, CUB-200.

for large-scale images as predictor architecture. We use $K = 180$, and apply random horizontal flip and random cropping of zero-padded image as data augmentation. The predictor is initialized with network pretrained on ImageNet and trained for 50 epochs. For both datasets, we do not vary the other hyperparameters much compared to experiments on CIFAR10. The hidden layers accessed are same for both. The hyperparameters of the interpretability loss remain unchanged for CIFAR100 and for CUB-200 we increase β and γ to 1.0 and 3.0, respectively.

We report the accuracy of BASE-f, FLINT-f and FLINT-g models (single run) and fidelity of FLINT-g to FLINT-f in Tab. 4.7 and conciseness below in Fig. 4.10. It should be noted that due to high number of classes, the disagreements between f and g are more common. The generated interpretations for the class predicted by f can still be useful if it is among top classes predicted by g (for a more detailed discussion, see Sec. 4.4.4). Thus we report below top-k fidelity of g to f for $k = 1, 5$ (the default fidelity of interpreter metric corresponds to $k = 1$). We also illustrate visualizations of sample relevant class-attribute pairs with global relevance $r_{k,c} > 0.5$ in Fig. 4.11 for CIFAR100, and for CUB-200 in Figs. 4.12, 4.14, 4.13 , 4.15.

Key observations: FLINT-f achieves almost the same accuracy as BASE-f model for both datasets, competitive for models of this size. Given the large number of classes, it achieves high fidelity of interpretations with top-1 fidelity of more than 80% and top-5 fidelity around 97% for both datasets. The effect of increased number of classes and complexity of datasets is also seen in comparatively higher conciseness of FLINT. However, relative to the total number of attributes, the interpretations still utilize small fraction of them, similar to results on other datasets. We also showcase the ability of attributes learnt in FLINT to capture interesting concepts. For eg. on CUB-200, we visualize various attributes which encode concepts like 'yellow-headed birds' (Fig. 4.12), 'red-headed birds' (Fig. 4.14), 'blue-faced birds' (Fig. 4.13) and 'long orange/red legs' (Fig. 4.15). The AM+PI procedure emphasizes these patterns and "imprints" them multiple times over any given input image. While this certainly provides greater insight into encoded information of an attribute, the lack of clear localization of these patterns in the visualization negatively affects understandability. We talk at length about this issue when discussing limitations in chapter 6.

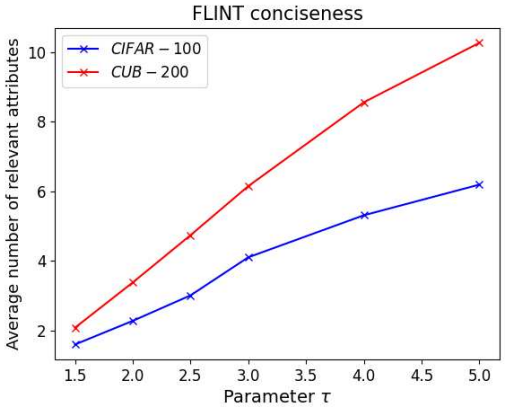


Figure 4.10: Conciseness curve of FLINT-g interpretations on CIFAR100 and CUB-200

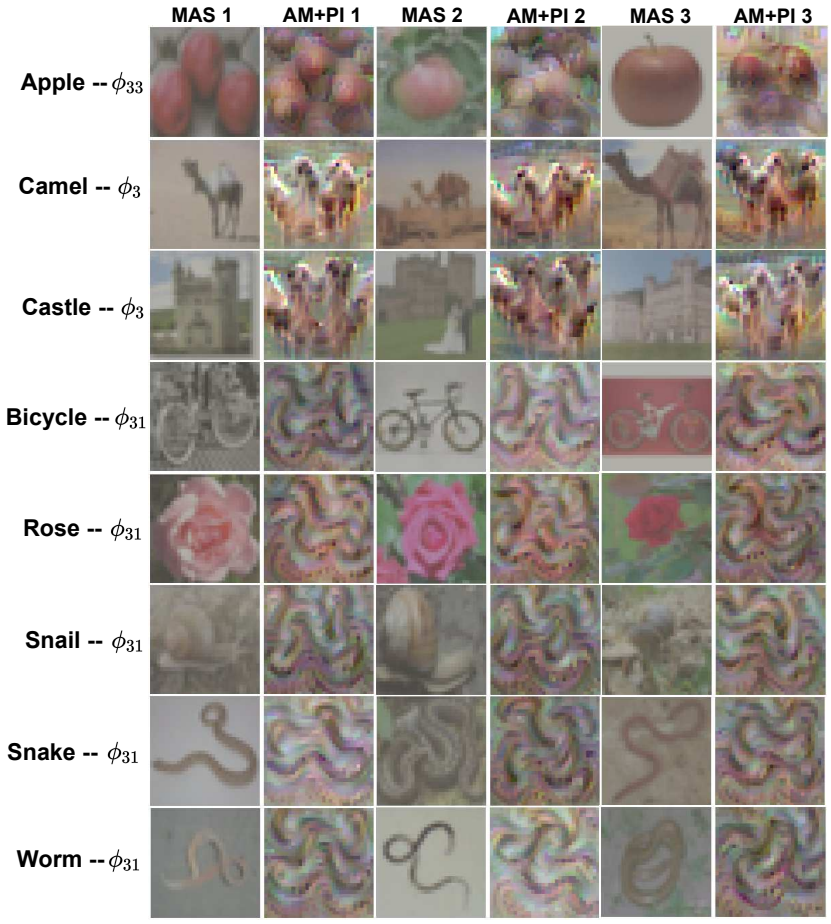


Figure 4.11: Sample class-attribute visualizations for CIFAR100. Three MAS and their corresponding AM+PI outputs are shown.

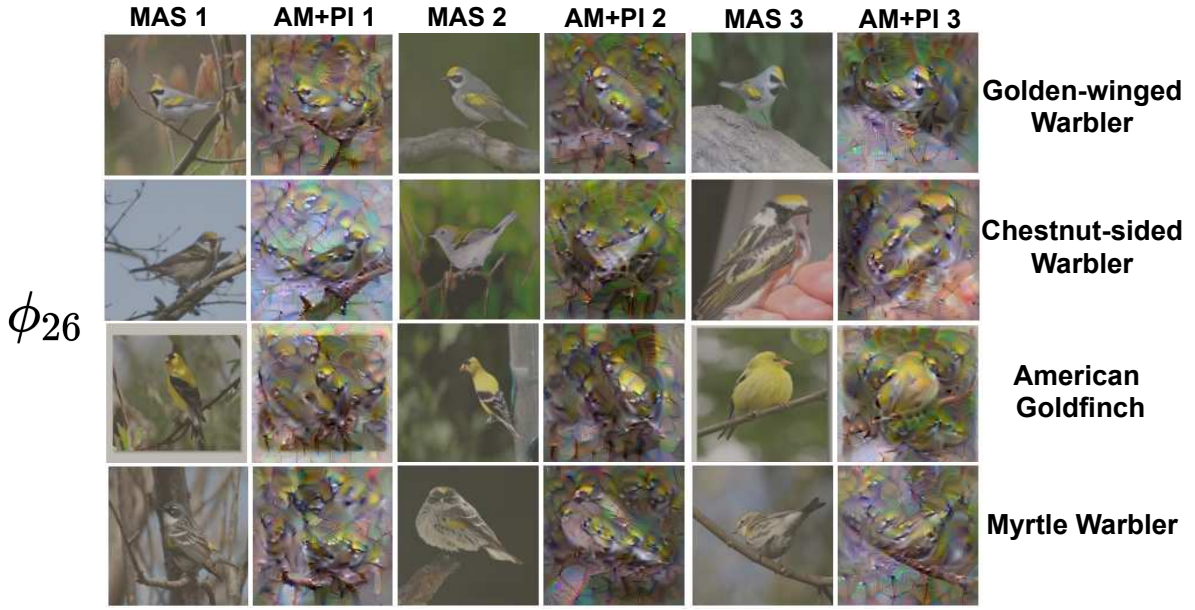


Figure 4.12: Relevant class-attribute pairs on CUB-200 with attribute ϕ_{26}. Each row gives visualization for a relevant class of the attribute with three MAS and corresponding AM+PI outputs.

4.6 Specialization to post-hoc interpretability

While interpretability by design has been the primary focus in this chapter, the SLI objective can easily be specialized to provide a *post-hoc* interpretation when a classifier $\hat{f}$ is already available. We quickly recap the learning problem in this case. Given a classifier $\hat{f} \in \mathcal{F}$ and a training set $\mathcal{S}$, the goal is to build an interpreter of $\hat{f}$ by solving:

$$\textbf{Problem 2: } \arg\min_{g \in \mathcal{G}_{\hat{f}}} \mathcal{L}_{int}(\hat{f}, g, \mathcal{S}).$$

With FLINT, we have $g(x) = h \circ \Phi(x)$ and $\Phi(x) = \Psi \circ \hat{f}_{\mathcal{I}}(x)$ for a given set of accessible hidden layers $\mathcal{I}$ and a attribute dictionary size J. Learning can be performed by specializing Alg. 4.1 with slight modification of replacing Θ as $\Theta = (\theta_\Psi, \theta_h, \theta_d)$ while $\theta_{\hat{f}}$ is fixed and eliminating $\mathcal{L}_{pred}$ from training loss $\mathcal{L}$. Note that the model architecture and loss functions directly affecting the intermediate embedding remain precisely the same as before.

Experimental results for post-hoc FLINT: We validate this ability of our framework by interpreting fixed models trained only for accuracy, i.e, BASE-f models from section 4.4.1. Even after not tuning the internal layers of f, the system is still able to generate high-fidelity, concise and meaningful interpretations. We report these metrics Fidelity benchmarked against VIBI is tabulated in Tab. 4.8 and conciseness curves for post-hoc interpretations are shown in Fig. 4.16. They clearly indicate that FLINT can yield high fidelity and highly concise *post-hoc* interpretations. Sample visualization of the attributes is provided in appendix G.2.1. We also qualitatively study application of a post-hoc CAV-based method in appendix G.2.2

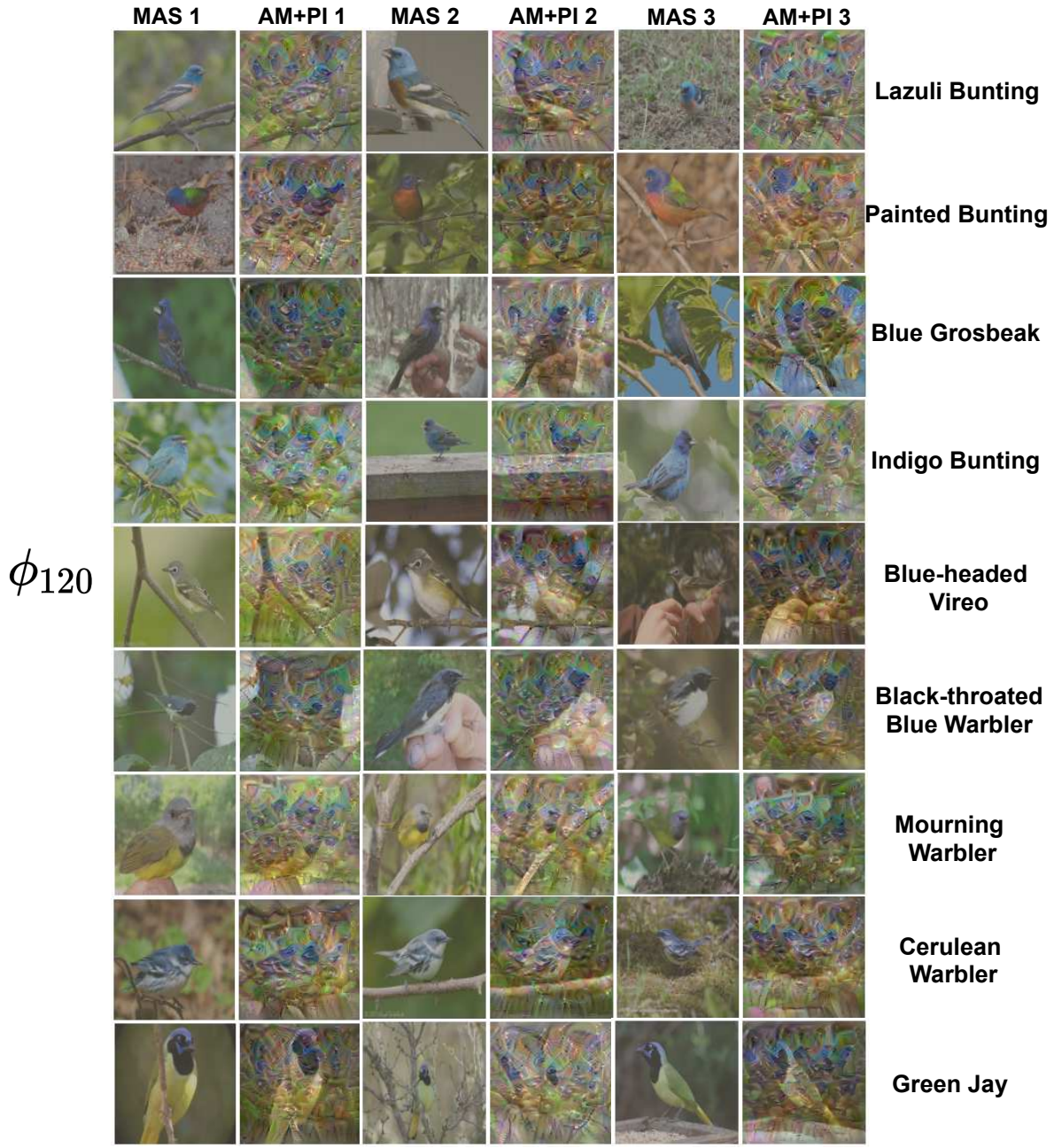


Figure 4.13: Relevant class-attribute pairs on CUB-200 with attribute ϕ_{120}. Each row gives visualization for a relevant class of the attribute with three MAS and corresponding AM+PI outputs.

4.7 Conclusion

FLINT is a novel framework for learning a predictor network and its interpreter network with dedicated losses. It provides local and global interpretations in terms of high-level learnt attributes/concepts by relying on (some) hidden layers of the prediction network. We demonstrate interpretations with high fidelity, predictive per-

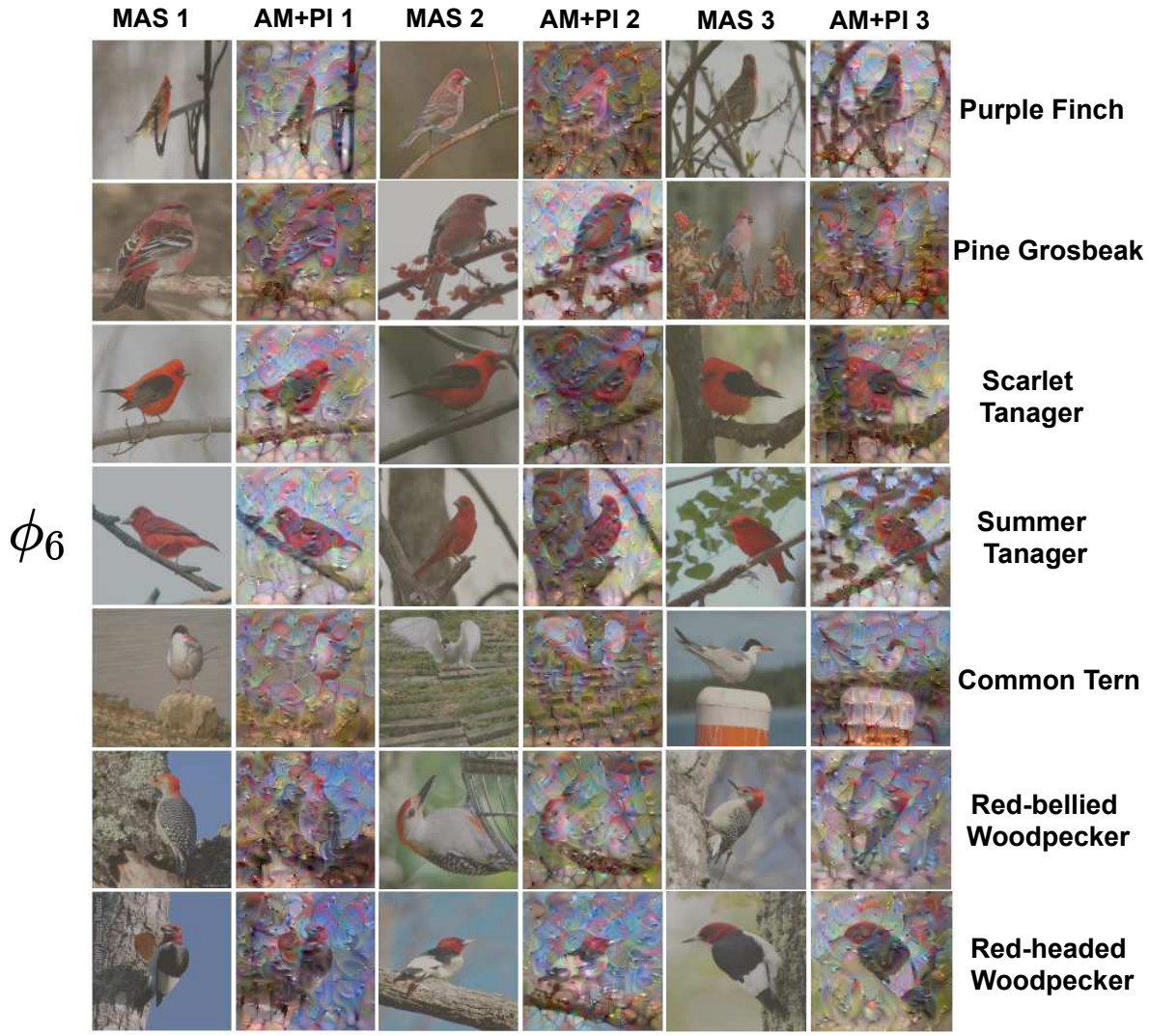


Figure 4.14: Relevant class-attribute pairs on CUB-200 with attribute ϕ_6. Each row gives visualization for a relevant class of the attribute with three MAS and corresponding AM+PI outputs.

formance and conciseness (due to entropy-based criterion), validated against some state-of-the-art models on multiple image classification benchmarks. We proposed a novel pipeline to understand concepts encoded in attribute functions, which we qualitatively evaluated for understandability.

This chapter however leaves some under-explored questions about *faithfulness* of interpretations to f. Computing faithfulness of an interpretation in the case of post-hoc interpretability or when the two models, predictor and interpreter, differ (Yin et al., 2021) is not trivial as there isn't any way to measure impact of any relevant attribute. Even though generating interpretations based on hidden layers of predictor ensures high level of faithfulness of the interpreter to the predictor, a complete faithfulness cannot be guaranteed since predictor and interpreter differ in their last part. How-

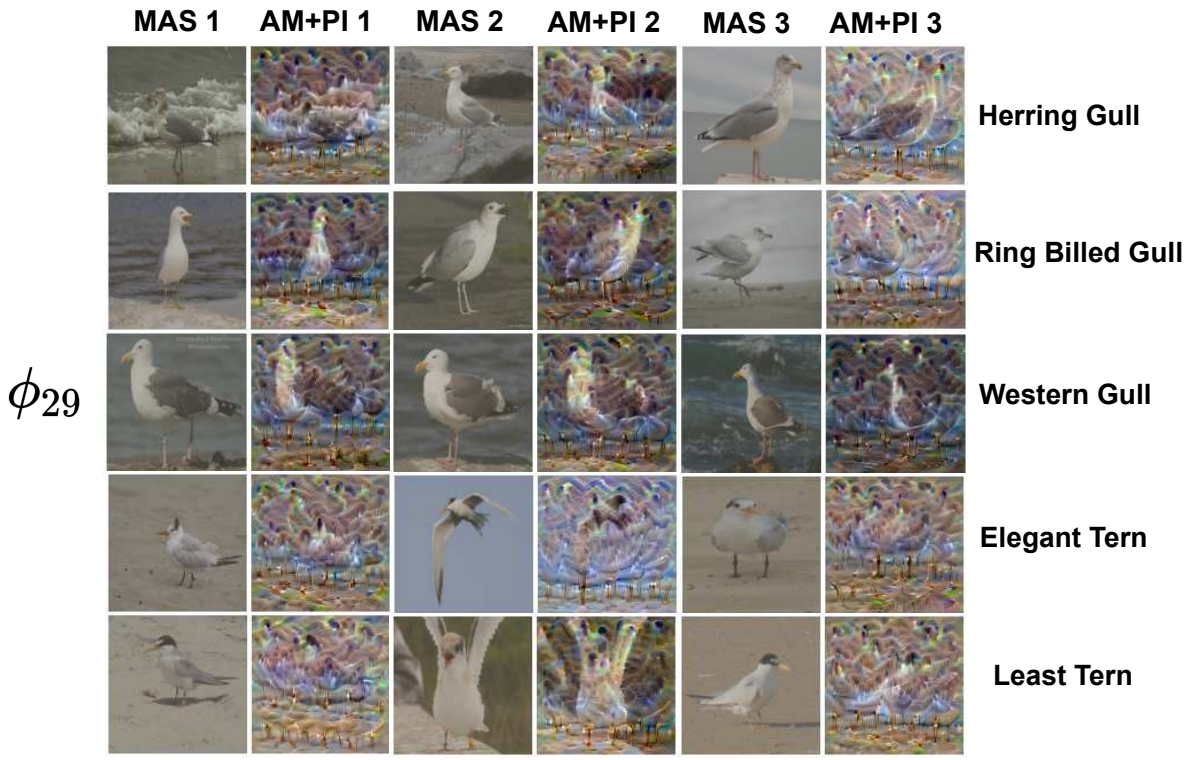


Figure 4.15: Relevant class-attribute pairs on CUB-200 with attribute ϕ_{29}. Each row gives visualization for a relevant class of the attribute with three MAS and corresponding AM+PI outputs.

Dataset	VIBI	FLINT-g
MNIST	95.8±0.2	**98.6±0.2**
FashionMNIST	88.4±0.2	**92.8±0.3**
CIFAR10	64.2±0.3	**89.1±0.5**
QuickDraw	78.0±0.4	**90.5±0.3**

Table 4.8: Fidelity for post-hoc interpretations of BASE-f (in %)

ever if ensuring faithfulness by design is regarded as the primary objective, nothing stops the use of interpreter FLINT-g as the final decision-making network. In this case, there is only one network and the so-called prediction network has only played the useful role of providing relevant hidden layers.

Contents

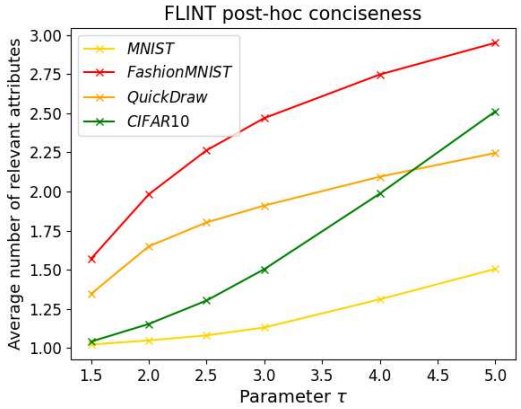


Figure 4.16: Conciseness curve of post-hoc interpretations generated using FLINT

5

Tackling Interpretability for Audio Classification Networks

5.1 Introduction

In this chapter, our aim is to address both problems for audio classification networks while proposing a system more suited for understanding interpretations for audio modality beyond the common methods in literature.

An ideal interpreter is supposed to offer insights about a model's decision in an understandable fashion to humans. In the case of audio classification, there are certain desirable traits for an interpreter that effectively help to fulfil this purpose. Firstly, we advocate that the interpretations should be generated in terms of **high-level audio objects**. Even more importantly, the interpretation should be **listenable** for an end-user. The rationale behind posing these traits as desirable is as follows: Audio scenes are often composed of multiple high-level audio objects (Bregman, 1994). Moreover, understanding events/scenes through the notion of audio objects also aligns with cognitive development in human and animals (Griffiths and Warren, 2004; Dyson and Alain, 2004). Listenability is essential since it is significantly more intuitive and easier to listen to an interpretation rather than visualizing it in its time-frequency representation (eg. spectrogram). Usefulness for both the traits can be reinforced through an example. Imagine an audio-based surveillance system for a house raising an alarm for break-in. An interpreter can be expected to be able to localize the event among a host of concurrent events that triggered the alarm. If for example 'glass-breaking' is the triggering event that the interpreter recognizes in the input, a human would find it easier to understand, if they can hear the interpretation rather than visualize it on a spectrogram. Saliency maps or FLINT (chapter 4) face non-trivial issues to be usable for generating listenable interpretations (see appendix H.2). Other approaches tailored for audio (Mishra et al., 2020; Haunschmid et al., 2020; Zinemanas et al., 2021) are either limited in applicability or do not utilize concept-based representations.

To this end, we propose an interpreter that relies on processing selected hidden representations of the classifier by a neural network to extract an intermediate embedding. This intermediate encoding is regularized in multiple ways, the two essential ones being: (i) Mimicking the classifier output to be able to interpret its decisions, and (ii) Reconstruct the input through the help of a dictionary of spectral patterns. The latter loss and its design is strongly inspired by the structure in Non-negative

Matrix Factorization (NMF, (Lee and Seung, 2001)), known to provide part-based decompositions. The loss is crucial in imposing a highly understandable meaning of "time activations" on the intermediate embedding. This decomposition structure also allows the interpreter to benefit from filtering information from the input. It's worth emphasizing that audio interpretability is not the same as classical tasks of separation or denoising. These tasks involve recovering complete object of interest in the output audio. On the other hand, a classifier network might focus more on salient regions. When interpreting its decision and making it listenable we expect to uncover such regions and not necessarily the complete object of interest.

In summary, we make the following contributions:

- We build a holistic approach that generates listenable concept-based interpretations to tackle post-hoc and by-design interpretability for audio classification networks.
- We present an original formulation that constrains the interpreter encoding through two loss functions, one for input reconstruction through NMF dictionary and the other for fidelity to the network's decision. From a learning perspective, we show a new way to link NMF with deep neural networks, especially for generating interpretations.
- We extensively evaluate on three popular audio event analysis benchmarks, tackling both multi–class and multi–label classification tasks. The dataset for the latter is very challenging due to its collection in noisy real–world settings. Our method's design allows us to simulate feature removal and perform *faithfulness* evaluation.

5.2 A primer on NMF

We briefly go through the traditional definition of NMF and its applications for audio signals before we describe our system.

5.2.1 NMF basics

NMF is a data decomposition technique popularized by Lee and Seung (2001) as a method to learn "parts of an object". It is a popular technique for unsupervised decomposition of audio signals (Badeau and Virtanen, 2018). Given any positive time–frequency representation $\mathbf{X} \in \mathbb{R}_+^{F \times T}$ consisting of F frequency bins and T time frames, NMF decomposes it into a product of two non-negative matrices, such that,

$$\mathbf{X} \approx \mathbf{W}\mathbf{H}$$

Here, $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_K] \in \mathbb{R}_+^{F \times K}$ is interpreted as the spectral pattern or dictionary matrix containing K components and $\mathbf{H} = [\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_K]^\intercal \in \mathbb{R}_+^{K \times T}$ a matrix containing the corresponding time activations. Typically, a β-divergence measure between $\mathbf{X}$ and

WH is minimized and multiplicative updates are used for estimating **W** and **H** (Lee and Seung, 2001). Note that it is possible to reconstruct signal corresponding to each or a group of spectral components. This is typically done using a procedure called soft–masking. For a single component k, this is written as,

$$\mathbf{X}_k = \frac{\mathbf{w}_k \mathbf{h}_k{}^\intercal}{\mathbf{WH}} \odot \mathbf{X}$$

Both ./. and $\odot$ are element-wise operations. If **X** is an invertible representation of the magnitude spectrogram, time domain signal for $\mathbf{X}_k$ is easily recovered using the inverse STFT operation. We extensively utilize this procedure for generating listenable interpretations. NMF can also be used for dictionary learning, by estimating **W** on a training dataset matrix $\mathbf{X}_{\text{train}}$. As discussed later, we use a variant of NMF called Sparse-NMF (Le Roux et al., 2015b) to pre-learn dictionary for subsequent usage in the interpretation module.

5.2.2 NMF applications for audio

NMF has since been used widely within the audio community to tackle source separation (Smaragdis, 2004), denoising (Wilson et al., 2008), inpainting (Le Roux et al., 2011) and transcription (Dittmar and Gärtner, 2014; Bertin et al., 2007). Bisot et al. (2017) couple NMF-based features with neural networks to boost performance of acoustic scene classification. NMF has also been successfully employed with audio–visual deep learning models for separation (Gao et al., 2018) and classification (Parekh et al., 2019).

Iterations of NMF optimization algorithms can be unfolded as novel deep neural networks. This observation has led to development of "Deep NMF" methods. In particular, Le Roux et al. (2015a) unfold the multiplicative updates of NMF parameters into a deep network for speech separation. Wisdom et al. (2017) apply this strategy to iterative soft thesholding algorithm to propose deep recurrent NMF.

While these works share with us the high-level idea of combining neural networks and NMF, there is no overlap between our goals and methodologies. Unlike aforementioned studies, we wish to investigate a classifier's decision using NMF as a regularizer. Furthermore, to our best knowledge, attempting to regress temporal activations of a fixed NMF dictionary by accessing intermediate layers of an audio classification network is novel even within the NMF literature.

5.3 System design

We organize this section as follows: We start with a brief note on notation used throughout the chapter. We describe the setup of our framework to address post-hoc interpretation in section 5.3.2. This is extended to address by-design interpretation in section 5.3.3. We expound on the specific architectural details common to both problem settings in section 5.3.4 and conclude the section by detailing how we generate interpretations with our design in section 5.3.5.

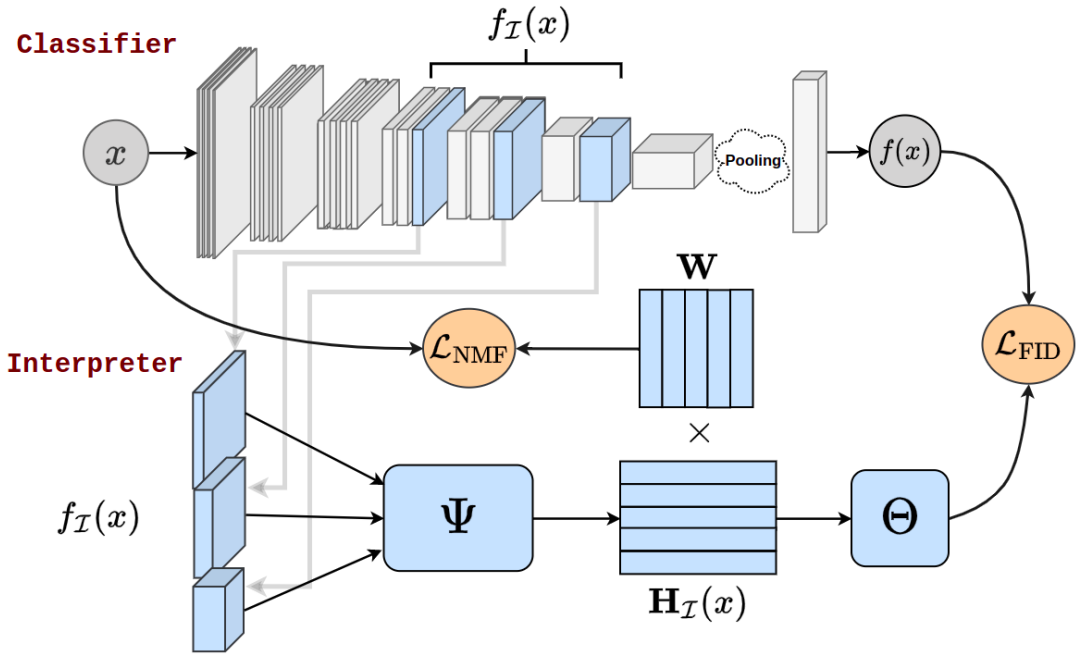


Figure 5.1: **System overview**: The core design common to both post-hoc and by-design interpretation. The interpreter (indicated in blue) accesses hidden layer outputs of the classifier. These are used to predict an intermediate encoding. Through regularization terms, we encourage this encoding to both mimic the classifier's output and also serve as the time activations of a pre-learnt NMF dictionary. In post-hoc interpretation, the classifier is pre-trained and fixed, and only the interpreter is trained. For by-design interpretation we train both jointly and make final predictions using output of interpreter.

As an additional note, we will use slightly different notations for representations and loss functions compared to the ones introduced in Chapter 3, in order to (a) follow NMF-like notations for parts motivated by it, and (b) make distinctions between our designs for FLINT and the current system. The skeleton of this system is exactly same as in Chapter 3 and we will clearly highlight the equivalence between the two at a later stage.

5.3.1 Data Notation

We denote a training dataset by $\mathcal{S} := (x, y)_{i=1}^{N}$, where $x \in \mathcal{X}$ is the time domain audio signal and $y \in \mathcal{Y}$, a label vector. The label vector could be a one-hot or binary encoding depending upon a multi-class or multi-label dataset, respectively. For listenable interpretations through NMF, we favor a representation of x that can be easily inverted back to the time-domain and use a log–magnitude spectrogram $\mathbf{X} \in \mathbb{R}^{F \times T}$ that is computed by applying an element-wise transformation $x_0 \rightarrow \log(1 + x_0)$ on the magnitude spectrogram with F frequency bins and T time frames. This is preferred over using magnitude spectrograms as it corresponds more closely to human perception

of sound intensity Goldstein (1967). A deep neural network classifier for post-hoc interpretations is denoted as $f : \mathcal{X} \to \mathcal{Y}$.

5.3.2 Post-hoc interpretation

When addressing the problem of post-hoc interpretation, the classifier f will be pre-trained and then fixed throughout. We describe now the components of the interpreter and what are its inputs and outputs.

Overview The system design is illustrated in figure 5.1. The interpreter is designed to have access to hidden representations of the classifier and is tasked to produce an intermediate embedding through the function Ψ. This embedding is placed under certain constraints via function Θ and a pre-learnt dictionary of NMF components $\mathbf{W}$. These constraints impose a highly meaningful structure on the embedding and help in interpreting the decision $f(x)$. We discuss the constrains, which form the core of our approach, in this subsection. The precise architectures of Ψ and Θ and optimization problem used to pre-learn $\mathbf{W}$ are covered later in Sec. 5.3.4.

Specifically, hidden layer outputs of the classifier f, taken as input by the interpreter, are denoted as $f_{\mathcal{I}}(x) \in \mathcal{Z}$. They are processed through the function $\Psi : \mathcal{Z} \to \mathbb{R}_{+}^{K \times T}$, modelled as a neural network. This produces an intermediate encoding. For simplicity, we will denote this encoding generated from hidden layers as $\mathbf{H}_{\mathcal{I}}(x) = \Psi \circ f_{\mathcal{I}}(x)$, a function over input x. The constraints on this encoding, implemented as loss functions are as follows:

Loss 1 (Fidelity loss): To be able to identify the relevant signal for interpretation, we constrain $\mathbf{H}_{\mathcal{I}}(x)$ to approximate classifiers output probabilities $f(x)$ through the function $\Theta : \mathbb{R}_{+}^{K \times T} \to \mathcal{Y}$. The term $\Theta(\mathbf{H}_{\mathcal{I}}(x))$ is also referred to as interpreter's output. We implement this constraint as a loss function by minimizing the generalized cross-entropy loss between $\Theta(\mathbf{H}_{\mathcal{I}}(x))$ and $f(x)$. We refer to it as the fidelity loss $\mathcal{L}_{\text{FID}}$. Denoting the parameters of Ψ, Θ as V_{Ψ}, V_{Θ}, for multi-class classification the loss can be written as,

$$\mathcal{L}_{\text{FID}}(x, V_{\Psi}, V_{\Theta}) = -f(x)^{\intercal} \log(\Theta(\mathbf{H}_{\mathcal{I}}(x))) \tag{5.1}$$

On the other hand, for multi-label classification this loss reads,

$$\begin{aligned} \mathcal{L}_{\text{FID}}(x, V_{\Psi}, V_{\Theta}) = &- \sum f(x) \odot \log(\Theta(\mathbf{H}_{\mathcal{I}}(x))) \\ &+ (1 - f(x)) \odot \log(1 - \Theta(\mathbf{H}_{\mathcal{I}}(x))). \end{aligned} \tag{5.2}$$

Here $\odot$ denotes element-wise multiplication.

Loss 2 (Reconstruction loss): We additionally constrain $\mathbf{H}_{\mathcal{I}}(x)$ to be able to reconstruct the input audio using pre-learnt dictionary $\mathbf{W} \in \mathbb{R}_{+}^{F \times K}$. This constraint asks to decompose input log-magnitude spectrogram as $\mathbf{X} \approx \mathbf{W}\mathbf{H}_{\mathcal{I}}(x)$, that is, a product of two non-negative matrices. This loss is based on popular non-negative matrix factorization. Crucially, this allows us to consider $\mathbf{H}_{\mathcal{I}}(x)$ as a time activation matrix for $\mathbf{W}$. We refer to this as the reconstruction loss, denoted as $\mathcal{L}_{\text{NMF}}$.

$$\mathcal{L}_{\text{NMF}}(x, V_{\Psi}) = \|\mathbf{X} - \mathbf{W}\mathbf{H}_{\mathcal{I}}(x)\|_2^2. \tag{5.3}$$

Loss 3 (Sparsity loss): In addition to $\mathcal{L}_{\text{FID}}$ and $\mathcal{L}_{\text{NMF}}$, we impose ℓ_1 regularization on $\mathbf{H}_{\mathcal{I}}(x)$ to encourage well-behavedness, especially for large dictionary sizes Le Roux et al. (2015b).

Training optimization. The complete loss function over our training dataset $\mathcal{S}$ can thus be given as:

$$\mathcal{L}(V_\Psi, V_\Theta) = \sum_{x \in \mathcal{S}} \mathcal{L}_{\text{FID}}(x, V_\Psi, V_\Theta) + \alpha \mathcal{L}_{\text{NMF}}(x, V_\Psi) + \beta \|\mathbf{H}_{\mathcal{I}}(x)\|_1 \tag{5.4}$$

where $\alpha, \beta \geq 0$ are loss hyperparameters. All the parameters of the system are constituted in the functions Ψ, Θ and dictionary $\mathbf{W}$. Since $\mathbf{W}$ is pre-learnt and fixed, the training loss $\mathcal{L}$ is optimized only w.r.t V_Ψ, V_Θ. As a reminder, when training the interpreter for post-hoc analysis, the classifier network is kept fixed. The final optimization problem addressed for post-hoc interpretation writes as follows:

$$\hat{\Psi}, \hat{\Theta} = \arg\min_{\Psi, \Theta} \mathcal{L}(V_\Psi, V_\Theta) \tag{5.5}$$

A reader should be able to clearly draw the parallels between the framework design introduced in Chapter 3 and the system explained above. In particular losses for interpretability $\mathcal{L}_{of}, \mathcal{L}_{if}, \mathcal{L}_{\text{conc}}$ are equivalent to $\mathcal{L}_{\text{FID}}, \mathcal{L}_{\text{NMF}}, \|\mathbf{H}_{\mathcal{I}}(x)\|_1$ respectively. Furthermore, the representation of interpretation $\Phi_{\mathcal{I}}(x)$ is same as $\mathbf{H}_{\mathcal{I}}(x)$ and the decoder d is implemented as dictionary $\mathbf{W}$.

5.3.3 By-design interpretation

Interestingly, the same framework can also be utilized to train an inherently interpretable model. As a first step, we propose the following function to be used for making final predictions

$$g : \mathcal{X} \to \mathcal{Y}, g(x) = \Theta \circ \Psi \circ f_{\mathcal{I}}(x)$$

which is a mixture of interpreter and classifier layers. One might be tempted to employ the same training mechanism for by-design problem as done for post-hoc interpretation. However, there is a difference in the problem setting we need to adapt for. Namely, the classifier layers are not trained for prediction as before. This implies that we cannot simply aim to generate meaningful representations from it as is.

To remedy the difficulty, we modify the training in two different ways: (i) Layers of f are now modified by backpropagating all interpreter losses to them, and thus are now jointly trained with the interpreter. (ii) We modify the loss function to include an additional prediction loss on the output $f(x)$ to train all the layers in f. The training loss function and optimization problem write as following:

$$\begin{aligned}
&\mathcal{L}_f(x, V_f) = -y^\intercal \log(f(x)) \\
&\mathcal{L}_{\text{NMF}}(x, V_\Psi, V_f) = \|\mathbf{X} - \mathbf{W}\mathbf{H}_{\mathcal{I}}(x)\|_2^2 \\
&\mathcal{L}_{\text{FID}}(x, V_\Psi, V_\Theta, V_f) = -f(x)^\intercal \log(\Theta(\mathbf{H}_{\mathcal{I}}(x)))
\end{aligned}$$

$$\mathcal{L}(V_\Psi, V_\Theta, V_f) = \sum_{x \in \mathcal{S}} \mathcal{L}_f(x, V_f) + \gamma \mathcal{L}_{\text{FID}}(x, V_\Psi, V_\Theta, V_f) + \alpha \mathcal{L}_{\text{NMF}}(x, V_\Psi, V_f) + \beta \|\mathbf{H}_\mathcal{I}(x)\|_1$$

$$\hat{\Psi}, \hat{\Theta}, \hat{f} = \arg\min_{\Psi,\Theta,f} \mathcal{L}(V_\Psi, V_\Theta, V_f)$$

A reader might question the need for applying a prediction loss at output of f when the function g described above is proposed to make final predictions. This is indeed a reasonable variant of our current choice and we resolve this issue by comparing the performance of both systems in experiments in section 5.5.4

5.3.4 Filling the gaps

It should be noted that the network architectures and other implementation details remain the same in both problem settings. We now cover the remaining architectural details of Ψ, Θ and the algorithm for pre-learning $\mathbf{W}$.

Design of Ψ. The network Ψ is tasked with producing the encoding $\mathbf{H}_\mathcal{I}(x) \in \mathbb{R}_+^{K \times T}$ from the set of convolutional feature maps of the classifier, given by $f_\mathcal{I}(x)$. These feature maps potentially originate from different layers and thus can be of different resolutions. To perform joint processing on them, each one is first appropriately transformed to ensure same width and height dimensions. The subsequent layers process these maps through some convolutional (with ReLU activation) and resampling layers. However, this composition is based on certain important aspects. Firstly, audio feature maps of CNNs with spectrogram-like inputs contain the notion of time and frequency along the width and height dimensions. Secondly, our goal with this network is to process a 3D representation of feature patterns across time and frequency, and convert it to a 2D intermediate encoding that can serve as time activation matrix of size $K \times T$. To achieve this, the subsequent convolutional layers continuously decrease resolution on the frequency axis and increase resolution the time axis to T frames. Furthermore, the input axis for number of feature maps corresponds to the axis of number of components K in output of Ψ, equal to the number of components in dictionary $\mathbf{W}$.

Design of Θ. The goal of this network is to mimic the output $f(x)$ by processing $\mathbf{H}_\mathcal{I}(x)$ This directly helps in shaping $\mathbf{H}_\mathcal{I}(x)$ to interpret $f(x)$. An important consideration for designing Θ was to keep its operations on $\mathbf{H}_\mathcal{I}(x)$ interpretable. This helps during the interpretation phase in easily quantifying how different parts of $\mathbf{H}_\mathcal{I}(x)$ influence the interpreters output. It is thus composed of two parts. The first part pools activations $\mathbf{H}_\mathcal{I}(x)$ across time. This pooling can be implemented in multiple ways, for eg. max or average pooling. However, we opt for an intermediate style of attention–based pooling Ilse et al. (2018), *i.e.*, $\mathbf{z} = \sum_{t=1}^{T} \mathbf{H}_\mathcal{I}(x)\mathbf{a}$, where $\mathbf{a} \in \mathbb{R}^T$ are the attention weights and $\mathbf{z} \in \mathbb{R}^K$ is the pooled vector. The pooled representation vector is passed through a linear layer. This is followed by an appropriate activation function to convert its output to probabilities, that is, softmax for multi-class classification and sigmoïd for multi-label classification.

Pre-learning W. The non-negative matrix $\mathbf{W}$ forms an integral part of the interpreter design. It is pre-learnt from the input data, and essential in formulating the reconstruction loss $\mathcal{L}_{\text{NMF}}$. We employ Sparse-NMF Le Roux et al. (2015b) for the pre-learning. The following optimization problem is solved through multiplicative updates to pre-learn $\mathbf{W}$:

$$\begin{aligned} \min \quad & D(\mathbf{X}_{\text{train}}|\mathbf{WH}) + \mu\|\mathbf{H}\|_1 \\ \text{subject to} \quad & \mathbf{W} \geq 0, \mathbf{H} \geq 0, \\ & \|\mathbf{w}_k\| = 1, \ \forall k. \end{aligned} \tag{5.6}$$

where $\mathbf{X}_{\text{train}}$ is a subset of the training data $\mathcal{S}$. Note that its construction is dataset dependent and will be covered in experiments. Here $D(.|.)$ is a divergence cost function. In practice, euclidean distance is used. Training audio files are converted into log-magnitude spectrogram space for factorization.

5.3.5 Generating Interpretations

Having described the goals and details of all components of our framework, we finally discuss how the interpretations are generated. To generate audio that interprets the classifier's decision for a sample x and a predicted class c, we follow a two-step procedure: The first step consists of identifying the components which are considered "important" for the prediction. This is determined by estimating their relevance using the pooled time activations in Θ and the weights for linear layer. Precisely, given a sample x, the pooled activations are computed as $\mathbf{z} = \mathbf{H}_{\mathcal{I}}(x)\mathbf{a}$. Denoting the weights for class c in the linear layer as θ^w_c, the relevance of component k is estimated as $r_{k,c,x} = \frac{(\mathbf{z}_k\theta^w_{c,k})}{\max_l |\mathbf{z}_l\theta^w_{c,l}|}$. This is essentially the normalized contribution of component k in the output logit for class c. To select the "important" components, we simply threshold the relevance via a parameter $\tau \in (0,1)$ as, $L_{c,x} = \{k : r_{k,c,x} > \tau\}$.

The second step consists of estimating a time domain signal for each relevant component $k \in L_{c,x}$ and also for set $L_{c,x}$ as a whole. In this paper, we refer to the latter as the generated interpretation audio, x_{int}. For certain classes, it may also be meaningful to listen to each individual component, x_k. As discussed earlier under NMF basics, estimating time domain signals from spectral patterns and their activations typically involves a soft–masking and inverse STFT procedure. We detail this step with appropriate equations in Algorithm 5.1.

5.4 Experimental design

Most of the experimental settings remain the same for post-hoc and by-design interpretations since the underlying architecture and the loss functions directly affecting interpreter are identical. Thus, the datasets, audio representation used by the network and the learnt dictionaries remain unchanged. However, there are some differences in training and evaluation that will be discussed explicitly. We start by

Algorithm 5.1 Audio interpretation generation

Input: log-magnitude spectrogram $\mathbf{X}$, input phase $\mathbf{P}_x$ components $\mathbf{W} = \{\mathbf{w}_1, \ldots, \mathbf{w}_K\}$, time activations $\mathbf{H}_{\mathcal{I}}(x) = [\mathbf{h}_1^{\mathcal{I}}(x), \ldots, \mathbf{h}_K^{\mathcal{I}}(x)]^\top$, set of selected components $L_{c,x} = \{k_1, \ldots, k_B\}$.
for all $k \in L_{c,x}$ **do**
  $\mathbf{X}_k \leftarrow \frac{\mathbf{w}_k \mathbf{h}_k^{\mathcal{I}}(x)^\top}{\sum_{l=1}^{K} \mathbf{w}_l \mathbf{h}_l^{\mathcal{I}}(x)^\top} \odot \mathbf{X}$ {// Soft masking}
  $x_k = \text{INV}(\mathbf{X}_k, \mathbf{P}_x)$ {// Inverse STFT}
end for
$\mathbf{X}_{\text{int}} \leftarrow \sum_{k \in L_{c,x}} \mathbf{X}_k$
$x_{\text{int}} = \text{INV}(\mathbf{X}_{\text{int}}, \mathbf{P}_x)$
Output: $\{x_{k_1}, \ldots, x_{k_B}\}$, x_{int}

covering the above details in section 5.4.1-5.4.2. We discuss the interpretation evaluation strategies relevant for both problems in section 5.4.3, including all the systems evaluated.

5.4.1 Datasets

We experiment with three datasets covering different types of learning tasks, source data etc. We discuss each of them in greater detail below.

ESC50: ESC-50 (Piczak, 2015) is a popular benchmark for environmental sound classification task. It is a multi-class dataset that contains 2000 audio recordings of 50 different environmental sounds. The classes are broadly arranged in five categories namely, animals, natural soundscapes/water sounds, human/non-speech sounds, interior/domestic sounds, exterior/urban noises. Each clip is five-seconds long and extracted from publicly available recordings on the `freesound.org` project. The dataset is prearranged into 5 folds.

SONYC-UST: The DCASE task used a very challenging real-world dataset called Sounds of New York City-Urban Sound Tagging (SONYC-UST) (Cartwright et al., 2019). It contains audio collected from multiple sensors placed in the New York City to monitor noise pollution. It consists of eight coarse-level and 20 fine-level labels. We opt for the coarse-level labeling task that involves multi-label classification into: 'engine', 'machinery-impact', 'non-machinery-impact', 'powered-saw', 'alert-signals', 'music', 'human-voice', 'dog'. This task is highly challenging for several reasons: (i) since it is real-world audio, the samples contain a very high level of background noise, (ii) the audio sources corresponding to the classes are often weak in intensity, as they are not necessarily close to the sensors, (iii) some classes may also be highly localized in time and more challenging to detect, (iv) lastly, noisy audio also makes it difficult to annotate, leading to labeling noise. This is especially true for training data labeled by volunteers.

OpenMIC-2018: The OpenMIC-2018 dataset (Humphrey et al., 2018) is composed of 20000 polyphonic audio recordings annotated with weak labels from among 20 instrument classes. The dataset was created by querying the content available on Free Music Archive under the Creatives Commons license with AudioSet concept ontology and using a multi-instrument estimator model trained on AudioSet data to suggest candidates for annotation. Each recording/clip is 10 seconds long. A single sample generally consists of weak labels of only a small subset of classes. Each instrument class has at least 500 positive and 1500 total annotated samples. Compared to SONYC-UST, the number of positive samples intra class and inter class are considerably more balanced. It is currently the only large publicly available dataset with multi-label annotation for polyphonic audio.

5.4.2 Implementation details

Classification network

We interpret a VGG-style convolutional neural network proposed by Kumar et al. Kumar et al. (2018). This network was chosen due to its popularity and applicability for various audio scene and event classification tasks. It can process variable length audio and has been pretrained on AudioSet (Gemmeke et al., 2017), a large-scale weakly labeled dataset for sound events. It takes as input a log-mel spectrogram. The architecture broadly consists of six convolutional blocks (B1–B6) followed by a convolutional layer with pooling for final prediction. Most convolutional blocks consist of two sets of conv2D + batch norm + ReLU layers followed by a max pooling layer. We fine-tune this network on each dataset separately before training our system for any post-hoc interpretations. For ESC-50, we modify only fully-connected layers after the convolutional blocks while for SONYC-UST and OpenMIC-2018, we modify all the layers during fine-tuning.

Classifier performance. On ESC-50, the classifier is evaluated using 5-fold cross-validation. It achieves an accuracy of $82.5 \pm 1.9\%$ over the 5 folds, higher than the average human accuracy of 81.3%. SONYC-UST is an unbalanced multi-label dataset. The evaluation is done using AUPRC based metrics. Our fine-tuned classifier achieves a macro-AUPRC (official metric for DCASE 2020 challenge) of 0.601. This is higher than the DCASE baseline performance of 0.510 and comparable to the best performing system macro-AUPRC of 0.649 (Arnault and Riche, 2020). Note that it is obtained without use of data augmentation or additional strategies to improve performance. OpenMIC-2018 is a relatively balanced multi-label dataset. To evaluate our trained classifier, we use the weighted average F1-score metric, proposed in the original paper. The metric computes for each class a weighted average of F1-scores over the positive and negative samples. The final score is the average over 20 classes. Our classifier achieves final score of 0.83, better than the VGGish based baseline score of 0.78 and competitive with other recent models. These details are tabulated in Tab. 5.1. As noted earlier, the pre-training is only executed for post-hoc interpretations.

System	ESC-50 (in %) top-1	SONYC-UST macro-AUPRC	OpenMIC-2018 avg-weighted-F1
Human accuracy Piczak (2015)	81.3	×	×
ESC50-CNN baseline Piczak (2015)	64.5	×	×
Arnault et al. Arnault and Riche (2020)	×	0.649	×
Koutini et al. Koutini et al. (2020)	×	×	0.822
VGGish Cartwright et al. (2020); Humphrey et al. (2018)	×	0.510	0.785
Current-f	82.5	0.601	0.831

Table 5.1: Benchmarking performance of pre-trained classifier f for post-hoc interpretation.

Audio time-frequency representation

For both the tasks, we perform the same audio pre-processing steps. All audio files are sampled at 44.1kHz. STFT is computed with a 1024-pt FFT and 512 sample hop size, which corresponds to about 23ms window size and 11.5ms hopsize. The log-mel spectrogram is extracted using 128 mel-bands.

Dictionary learning

The matrix on which we apply sparse-NMF to learn $\mathbf{W}$, $\mathbf{X}_{\text{train}}$, is constructed differently for each dataset due to their specific properties. For ESC-50, $\mathbf{X}_{\text{train}}$ is constructed by concatenating the log–magnitude spectrograms corresponding to each sample in the training data of the cross-validation fold (1600 samples for each fold). SONYC-UST however, is an imbalanced multilabel dataset with very strong presence of background noise. A procedure to learn components, as for ESC-50, yields many components capturing significant background noise, affecting understandability of interpretations. Hence, we process this dataset differently. We first learn $\mathbf{W}_{\text{noise}}$, that is, a set of 10 components to model noise using training samples with no positive label. Then, for each class, we randomly select 700 positively-labeled samples from all training data and learn 10 new components (per class) with $\mathbf{W}_{\text{noise}}$ held fixed for noise modeling. All $10\times8 = 80$ components are stacked column-wise to build our dictionary $\mathbf{W}$. While this strategy helps us reduce the number of noise-like components in the final dictionary, it does not completely avoid it. OpenMIC is instead a balanced multilabel dataset for rare noise presence. We simply select random 500 positively labeled samples for each of the 20 classes and learn 15 components. All of them are stacked together to create $\mathbf{X}_{\text{train}}$.

Hyperparameters

The hidden layers input to the interpreter module are selected from the convolutional block outputs. As is often the case with CNNs, the latter layers are expected to capture higher-order features. We thus select the last three convolutional block outputs as input to the network Ψ. The loss weights and number of components used for post-hoc interpretation are summarized in table 5.2. Ablation studies about all

Dataset	α	β	K	# of epochs
ESC-50	10.0	0.8	100	35
SONYC-UST	10.0	0.8	80	21
OpenMIC-2018	5.0	0.2	300	21

Table 5.2: Hyperparameters for all datasets for post-hoc interpretation

Dataset	γ	α	β	K	# of epochs
ESC-50	1.0	3.0	0.2	100	51
SONYC-UST	1.0	4.0	0.2	80	21
OpenMIC-2018	1.0	3.0	0.2	300	21

Table 5.3: Hyperparameters for all datasets for by-design interpretation

the hyperparameters and justification of their choices will be presented in the next section. The hyperparameters for by-design interpretation are guided by choices in post-hoc interpretation and are tabulated in table 5.3.

Optimization

All the networks are optimized using Adam Kingma and Ba (2014) with learning rate 2×10^{-4}.

5.4.3 Evaluating interpretations

Quantifying different aspects of interpretability has been a challenging research question recently. This challenge stems from the inherent subjectivity involved in its definition. Our unique style of "concept-like" basis for interpretation and global approximation of the base model results in a testing situation to conduct its evaluation, wherein no other method can be directly compared to it. We resolve this hurdle by evaluating different aspects of the interpretation separately. We first discuss quantitative metrics for post-hoc and by-design interpretation along with their goals, followed by discussion on subjective evaluation of interpretations.

Metrics and baselines (Post-hoc). The simplest aspect to evaluate is how well does the interpreter agree with the classifier's output. We refer to this metric as the *fidelity* metric. To do so for any given task, we utilize the same metric used to evaluate the classifier performance but instead treat classifiers output as ground truth and evaluate the interpreter's approximation $\Theta(\mathbf{H}_{\mathcal{I}}(x))$ w.r.t to it. Thus, for multi-class classification, this is done by computing fraction of samples where the class predicted by f is among the top-k classes predicted by the interpreter, referred to as *top-k fidelity*. For multi-label classification tasks with unbalanced number of positive samples of classes, we compute Area Under Precision-Recall Curve (AUPRC) based metrics. In case of balanced classes, we compute F1-score based metrics. We denote our pro-

posed Listen to Interpret (L2I) system, with attention based pooling in Θ by L2I w/ Θ_{ATT}. The most suitable baselines to benchmark its fidelity are *post-hoc* methods that approximate the classifier over input space with a single surrogate model. We select two state-of-the-art systems, FLINT Parekh et al. (2021) and VIBI Bang et al. (2021). A variant of our own proposed method, L2I w/ Θ_{MAX}, is also evaluated. Herein, attention is replaced with 1D max-pooling operation.

We also conduct a *faithfulness* evaluation for our interpretations. In general for any interpretability method, *faithfulness* tries to assess if the features identified to be of high relevance are *truly* important in classifier's prediction (Alvarez-Melis and Jaakkola, 2018a). Since a "ground-truth" importance measure for features is rarely available, attribution based methods evaluate faithfulness by performing feature removal (generally by setting feature value to 0) and observing the change in classifier's output (Alvarez-Melis and Jaakkola, 2018a). However, it is hard to conduct such evaluation for non-attribution or concept based interpretation methods on data modalities like image/audio, as simulating feature removal from input is not evident in these cases.

Interestingly, our interpretation module design allows us to simulate removal of a set of components from the input. Given any sample x with predicted class c, we remove the set of relevant components $L_{c,x} = \{k : r_{k,c,x} > \tau\}$ by creating a new time domain signal $x_2 = \text{INV}(\mathbf{X}_2, \mathbf{P}_x)$, where $\mathbf{X}_2 = \mathbf{X} - \sum_{l \in L_{c,x}} \mathbf{X}_l$. We define faithfulness of the interpretation to classifier f for sample x with:

$$\text{FF}_x = f(x)_c - f(x_2)_c \tag{5.7}$$

where $f(x)_c, f(x_2)_c$ denote the output probabilities for class c. It should be noted that this strategy to simulate removal may introduce artifacts in the input that can affect the classifier's output unpredictably. Also, interpretations on samples with poor fidelity can lead to negative FF_x. Both of these observations point to the potential instability and outlying values for this metric. Thus, we report the final faithfulness of the system as median of FF_x over test set, denoted by $\text{FF}_{\text{median}}$. A positive $\text{FF}_{\text{median}}$ would signify that interpretations generally tend to be faithful to the classifier.

As already discussed, it is not possible to measure faithfulness for concept-based *post-hoc* interpretability approaches. While measurement for input attribution based approaches is possible, the interpretations themselves and the feature removal strategies are different, making comparisons with our system significantly less meaningful. We thus compare our faithfulness against a *Random Baseline*, wherein the less-important components, those not present in $L_{c,x}$, are randomly removed. To compare fairly, we remove the same number of components that are present in $L_{c,x}$ on average. This would validate that, if the interpreter selects *truly* important components for the classifier's decision, then randomly removing the less important ones should not cause a drop in the predicted class probability.

We also emphasize at this point that works related to audio interpretability are not suitable for comparison on these metrics. Particularly, APNet (Zinemanas et al., 2021) is not designed for *post-hoc* interpretations. AudioLIME (Haunschmid et al., 2020) is not applicable on our tasks as it requires known predefined audio sources. Moreover,

SLIME (Mishra et al., 2020) and AudioLIME still rely on LIME (Ribeiro et al., 2016) for interpretations. It is a feature-attribution method that approximates a classifier for *each* sample separately. As discussed before, these characteristics are not suitable for comparison on our metrics.

Separate from the quantitative metrics, we conducted a subjective evaluation to evaluate quality and understandability of interpretations. Our design for the same was based on qualitative understanding of saliency maps for images. Attribution maps in images are qualitatively judged by observing the visual overlap in input with the given class being interpreted. In similar spirit, our design was based on providing the user with input and class being interpreted and asking them to rate auditory overlap of the interpretation and part of input audio corresponding to the class. Further details and results are covered in the next section. Apart from evaluating understandability, we also extensively analyze our interpretations qualitatively.

Metrics and baselines (By-design). For by-design interpretation, the faithfulness metric is much less significant. This is because the final classification output is generated by the interpreter itself and thus faithfulness is ensured by-design. The classification performance of the interpreter is the primary metric, similar in spirit to fidelity evaluation for post-hoc interpretations. We compare this with several baselines to (i) benchmark performance of our by-design interpretable network, and (ii) to evaluate the two key modifications introduced in the learning problem while extending from post-hoc to by-design interpretation (section 5.3.3). Specifically, the hidden layers of f are not pre-trained on the given dataset in by-design problem and updated jointly with interpreter layers. And secondly, applying an additional classification loss on $f(x)$ to affect the hidden layers. The various baselines and the reasons to include them are the following:

- Audio prototypical networks (APNet) (Zinemanas et al., 2021) act as a primary baseline from literature. It is an audio processing by-design interpretable network. While it generates interpretation differently from us, it is the only system in the literature addressing by-design interpretation for audio modality. Note that the dedicated post-hoc interpretation systems VIBI and SLIME are not relevant for this problem. For fair comparison, we use the same number of prototypes in their network as our number of components.

- In order to ascertain that using a CNN based representation for NMF offer advantage over typical NMF based representations in terms of prediction performance, we also evaluate performance of two NMF variants: Unsupervised NMF based classification and the Task-driven Dictionary Learning (TDL)-NMF model (Bisot et al., 2017). The unsupervised NMF model simply learns a dictionary on training data, computes average time activations on test samples and makes predictions using a linear model. The TDL-NMF model instead updates the initial learnt dictionary with classification loss from the linear model and thus learns them jointly. For both the systems, we experiment with use of two data types to learn NMF-dictionaries. The first is log-magnitude spectrograms and second is

power mel-spectrogram (with a square root transformation). We vary dictionary sizes from 64 to 512 components and report results for best performance.

- Given the framework level similarities between FLINT and L2I, we also evaluate the performance of the FLINT interpreter when trained for by-design interpretation. As before, we again emphasize that FLINT is not suitable for audio interpretations, but provides a interpretable network design for comparison of performance.
- Variants of L2I: We denote our proposed version of L2I for by-design interpretation as L2I_{BD} w/ Θ_{ATT}. We further evaluate two variants of our proposed classification network $g(x)$. The first variant "L2I_{BD}-NoPred" does not include a classification loss applied to $f(x)$ and instead applies it directly to $g(x)$. The second variant "L2I-PostHoc" is simply the interpreter trained for post-hoc interpretation. We compare with these variants to gain perspective on effect of differences between our formulations of post-hoc and by-design problems.

Details about the baseline implementations for both post-hoc and by-design, can be found in appendix H.2.3.

5.5 Results and discussion

5.5.1 Post-hoc interpretation

Fidelity

As discussed previously, to quantify fidelity, we use the same respective metrics as done to benchmark classifier performance but evaluate them for interpreter output w.r.t classifier output. For ESC-50, mean and standard deviation of top-k fidelity is calculated over the 5 folds. We show these results for $k = 1, 5$. For SONYC-UST, we report the macro-AUPRC, micro-AUPRC and max-F1 for the interpreter output w.r.t classifier. For fairness, we ignore the class 'non-machinery impact' from all class-wise evaluations involved in fidelity (*i.e.* macro-AUPRC) or faithfulness. This is because the classifier predicts only one sample in test set with positive label for this class, causing AUPRC scores to vary widely for different interpreters. For OpenMIC-2018, we report the Fidelity weighted F1-score for each system. All the above results are available in Tab. 5.4.

Among the four systems, VIBI performs the worst in terms of fidelity. This is very likely because it treats the classifier as a black-box, while the other three systems access its hidden representations. This strongly indicates that accessing hidden layers can be beneficial for fidelity of interpreters. While on ESC50, FLINT achieves the best fidelity, L2I w/ Θ_{ATT} outperforms all systems on the other datasets. It should be noted that our system variants distinctly hold the advantage of generating listenable interpretations over FLINT and VIBI. Nevertheless, these systems form strong baselines for fidelity and the results demonstrate that our interpreter can generate

	ESC-50 (in %)		SONYC-UST		OpenMIC-2018
System	top-1	top-5	macro-AUPRC	micro-AUPRC	avg-weighted-F1
L2I w/ Θ_{ATT}	65.7 ± 2.8	88.2 ± 1.7	**0.909 ± 0.011**	**0.917 ± 0.008**	**0.920 ± 0.004**
L2I w/ Θ_{MAX}	73.3 ± 2.3	92.7 ± 1.2	0.866 ± 0.014	0.913 ± 0.012	0.906 ± 0.004
FLINT	**73.5 ± 2.3**	**93.4 ± 0.9**	0.816 ± 0.013	0.907 ± 0.011	0.907 ± 0.004
VIBI (Bang et al., 2021)	27.7 ± 2.3	53.0 ± 1.8	0.608 ± 0.027	0.575 ± 0.019	0.581 ± 0.037

Table 5.4: Fidelity results for the interpreter w.r.t classifier's output on all datasets. We report top-1 and top-5 fidelity (in %) for ESC-50 (all five folds), AUPRC-based metrics for SONYC-UST and weighted F1-score averaged over all classes for OpenMIC-2018. All results contain mean and variance over three runs. Values in bold indicate maximum of the metric among all the evaluated systems (incl. baselines).

high-fidelity *post-hoc* interpretations. Moreover, its design is flexible w.r.t different pooling functions.

Faithfulness

In Table 5.5, we report median faithfulness $\text{FF}_{\text{median}}$ on ESC-50 for our primary system L2I w/ Θ_{ATT} at different thresholds τ averaged over the five folds. Smaller τ corresponds to higher $|L_{c,x}|$, which denotes the number of components being used for generating interpretations. Thus, for Random Baseline, we report $\text{FF}_{\text{median}}$ at the lowest threshold $\tau = 0.1$, to ensure removal of maximal number of components. To recall the definition of Random Baseline, please refer to Sec. 5.4.3. $\text{FF}_{\text{median}}$ for L2I w/ Θ_{ATT} is positive for all thresholds. It is also significantly higher than the Random Baseline, indicating faithfulness of interpretations.

The results for class-wise faithfulness on SONYC-UST and OpenMIC are illustrated in Fig. 5.2 and 5.3 respectively. We show $\text{FF}_{\text{median}}$ (absolute drop in probability) for our system and the Random Baseline. For most classes, interpretations can be considered faithful, with a significantly positive median compared to random baseline results, which are very close to 0.

Subjective evaluation

The test was conducted with 15 participants. Each participant was provided with 10 input samples, a predicted class by the classifier for each sample and the corresponding interpretation audios from SLIME and L2I. They were asked to rate the interpretations on a scale of 0-100 for the following question: *"How well does the interpretation correspond to the part of input audio associated with the given class?*". The 10 samples were randomly selected from a set of 36 (5-6 random test examples per class). For each sample, we ensured that the predicted class was both, present in the ground-truth and audible in input. Class-wise preference results and average ratings are shown in Fig. 5.4. L2I is preferred for 'music', 'dog' & 'alert-signal', SLIME is pre-

System	Threshold τ	$\text{FF}_{\text{median}}$
L2I w/ Θ_{ATT}	$\tau = 0.9$	0.002
	$\tau = 0.7$	0.004
	$\tau = 0.5$	0.012
	$\tau = 0.3$	0.040
	$\tau = 0.1$	0.113
Random Baseline	$\tau = 0.1$	$< 10^{-4}$

Table 5.5: Faithfulness results on ESC-50 for different thresholds, τ. We report $\text{FF}_{\text{median}}$ for proposed L2I w/ Θ_{ATT} and the Random Baseline.

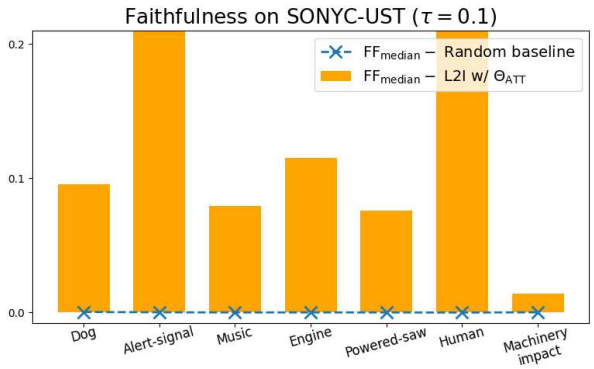


Figure 5.2: Faithfulness (absolute drop in probability value) results for SONYC-UST arranged class-wise for threshold, $\tau = 0.1$

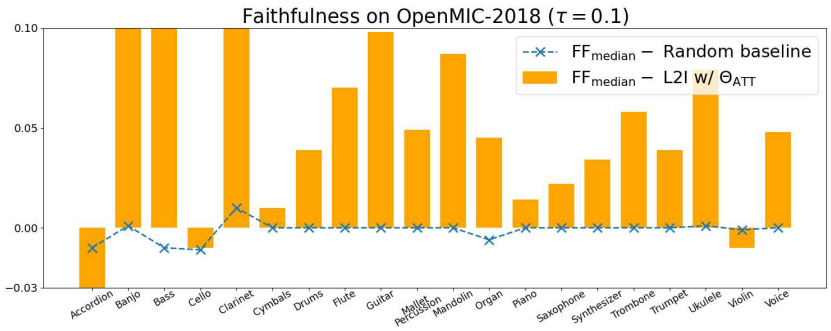


Figure 5.3: Faithfulness (absolute drop in probability value) results for OpenMIC-2018 arranged class-wise for threshold, $\tau = 0.1$

ferred for 'machinery-impact', no clear preference for others. Further details about the subjective evaluation are available in appendix H.2.4.

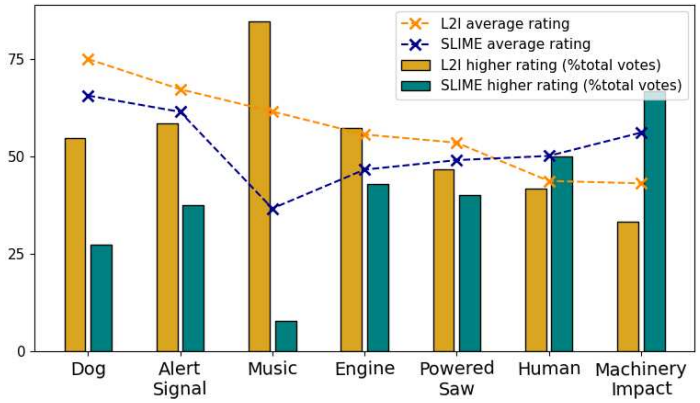


Figure 5.4: Subjective evaluation results. Average scores for L2I and SLIME and fraction of votes in favour of each system

5.5.2 Qualitative analysis of interpretations

Qualitatively we observe that our interpretations are capable of emphasizing the object of interest and are insightful for an end-user to understand the classifier's prediction. We share multiple examples on our companion website.[1] Samples in case of SONYC-UST and OpenMIC are often already challenging with the presence of other sources of audio. In case of ESC50, to create more interesting and challenging scenarios we devise an experiment described below

Audio corruption experiment: interpretability illustration. For ESC50, we generate interpretations after corrupting the testing data for fold–1 in two different ways (i) either with white noise at 0dB SNR (signal-to-noise ratio), (ii) or mixing it with a sample of a different class. It should be noted that in both these cases the system is exactly the same as before and **not** trained with corrupted samples. Some examples, covering both types of corruptions are shared on our companion website.[1]. Regardless, of the corruption audio, in most cases the system is able to clearly emphasize the object of interest. A detailed qualitative analysis of this experiment can be found in appendix H.1, including examples of cases where interpreter provides insights for misclassified samples. We also discuss about interpretations from a two saliency map methods and FLINT in appendix H.2 and highlight why they are not suitable in their current form for listenable interpretations.

For SONYC-UST, we observe good interpretations for classes 'alert-signal', 'dog' and 'music'. For them, the background noise is significantly suppressed and the interpretations mainly focus on the object of interest. Interpretations for class 'human' are also able to suppress noise to a certain extent and focus on parts of human voices. However, for this class, we found presence of some signal from other audio sources too. For the remaining classes, namely 'Engine', 'Powered-saw' and 'Machinery-impact' the quality of the interpretation is more sample dependent. This is due to their acoustic similarity with the background noise. We provide example interpretations for

[1] https://jayneelparekh.github.io/listen2interpretV2/

SONYC-UST on our companion website.[1]

The third dataset OpenMIC-2018, offers challenges under unique scenarios. Unlike SONYC-UST while it does not face issue of noise in data, it faces the hurdle of a strong overlap between instruments. This is because their onsets are often aligned by beats of the musical piece. This increases difficulty of filtering the signal of interest. Even with the greater complexity, the interpretations in many cases are able to emphasize the class of interest. Classes with relatively unique sounds such as 'Bass' or 'Mallet-percussion' are very well extracted. String like instruments including Violin and Guitar are also generally emphasized well.

Coherence of interpretations. We visualize interpretations generated on the test set for SONYC-UST and OpenMIC-2018 by clustering relevance vectors. Specifically, we compute the vector $r_{c,x} \in \mathbb{R}^K$ which contains relevances of all components in prediction for class c for sample x. The relevance vectors are collected for each test sample x and its predicted class c. We then apply a t-SNE (LJPvd and Hinton, 2008) transformation to 2D for visualization. This is shown in Fig. 5.5. Each point is labeled/colored according to the class for which we generate the interpretation. Interpretations for any single class are coherent and similar to each other. This is to some extent a positive consequence of global weight matrix in Θ. Moreover, globally it can be observed that classes like 'Machinery-impact' and 'Powered-Saw' have similar relevances which are to some extent close to 'Engine'. This is to be expected as these classes are acoustically similar. 'Dog' and 'Music' are also close in this space, likely due to the often periodic nature of barks or beats. The visualization for OpenMIC is arguably even more interesting because of larger number of classes and several inter-class relationships. Various sets of similar instruments end-up as clusters in proximity of each other. The examples include 'Cello-Violin', 'Drums-Cymbals', 'Clarinet-Flute', 'Ukulele-Mandolin-Banjo', 'Trombone-Trumpet-Saxophone'. Moreover, the meaningfulness of clustering also extends to higher-level of grouping. For example, the data is partitioned so as the string-based, wind-based, or percussion instruments are close to each other within their respective groups. This indicates that the interpreter's representations of what constitutes sound of an instrument aligns to some extent to human understanding.

5.5.3 Ablation studies

Tab. 5.6 and Tab. 5.7 present ablation studies for loss hyperparameters and choice of hidden layers. The values in bold indicate our current choices for post-hoc interpretation. The metrics and loss values given here are for a single run.

Selecting the **hidden layers** of the classifier that should be accessed by the interpreter is an important choice. At first glance, this model selection task might appear to be computationally too expensive as total possible choices is exponential in number of hidden layers. However, practical considerations can heavily reduce the search space. An upper bound to the number of layers could be set depending upon the desired size of interpreter. In our experiments throughout the paper, we limited ourselves to

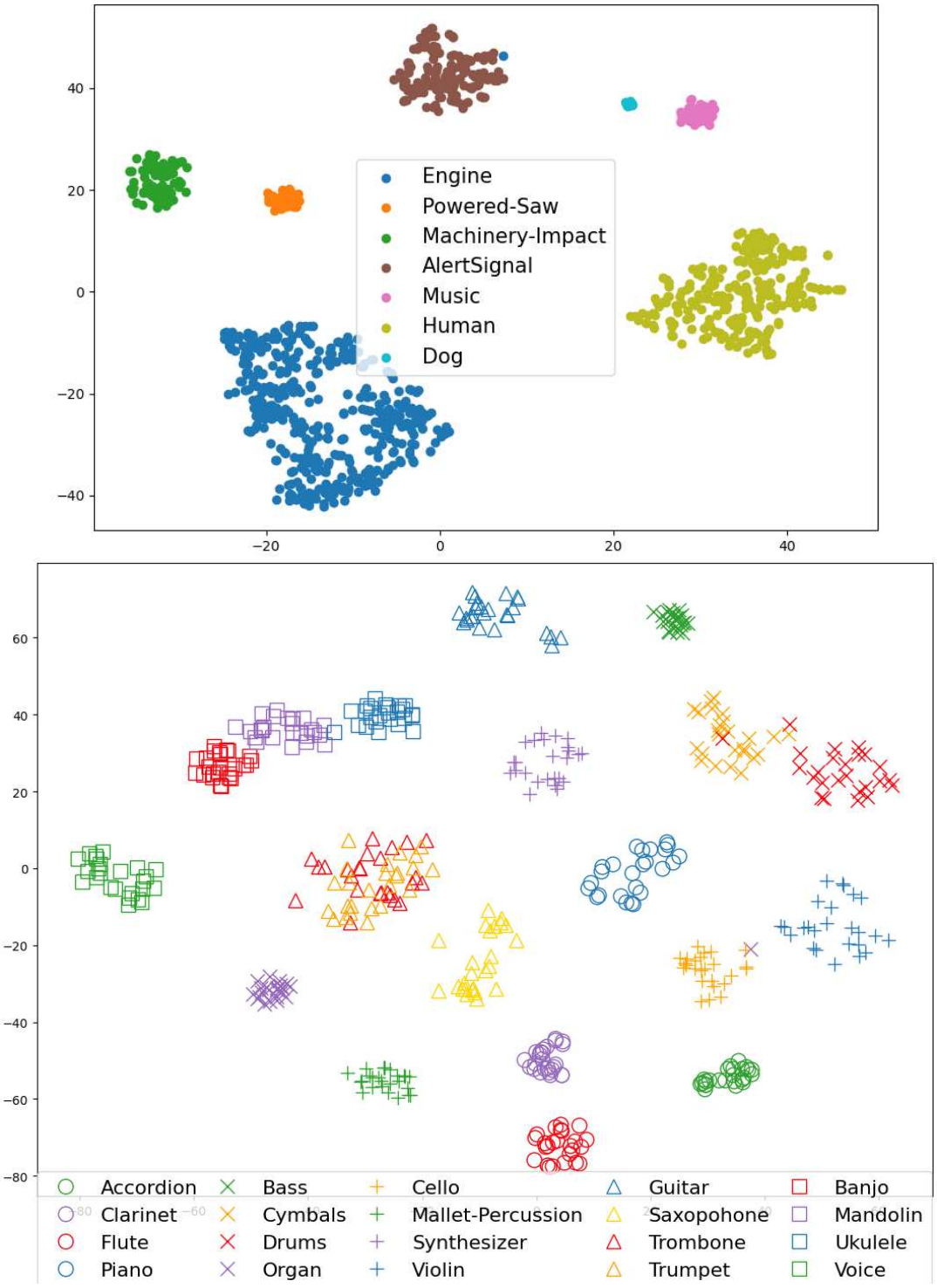


Figure 5.5: Visualized relevances (following a t-SNE transformation) of generated interpretations on test sets of SONYC-UST (top) and OpenMIC-2018 (bottom), colour-coded according to interpreted class. For clarity in case of OpenMIC, we show up to random 25 interpretations of a class.

ConvBlocks	$\mathcal{L}_{\text{NMF}}$	$\mathcal{L}_{\text{FID}}$	top-1
B3	0.104	1.788	53.0
B6	0.118	1.698	57.8
B2+B3	0.093	1.966	51.8
B5+B6	0.103	1.572	61.5
B4+B5+B6	**0.079**	**1.546**	**65.5**
Input	0.102	2.384	34.5

Table 5.6: Ablation study for hidden layers: loss values on ESC50 (fold 1) test set for different subsets of hidden layers. Current choice indicated in bold.

α	β	$\mathcal{L}_{\text{NMF}}$	$\mathcal{L}_{\text{FID}}$	macro-AUPRC
10.0	**0.8**	**0.028**	**0.386**	**0.900**
10.0	8.0	0.048	0.386	0.879
10.0	0.08	0.028	0.388	0.876
1.0	0.8	0.045	0.375	0.921
100.0	0.8	0.027	0.445	0.612

Table 5.7: Ablation study for loss hyperparameters: loss values on SONCY-UST test set for different weights of loss functions. Current choice indicated in bold.

at most 3 layers. Crucially, layers close to the output are more favourable, for multiple reasons. They generally result in better fidelity and inherently tie the interpreter much closer to the output of classifier. Moreover, the latter layers are also expected to capture higher level features. We illustrate how selecting different subsets of hidden layers affects optimization of our fidelity and reconstruction loss by doing an ablation study. It's results are reported in table 5.6. The classifier consists of 6 major convolutional blocks (B1–B6).

Loss weights. We illustrate the effect of varying loss weights on optimization in table 5.7. Too high emphasis on $\mathcal{L}_{\text{NMF}}$, that is, high α can hurt the fidelity of interpreter while a high β (sparsity loss) can result in poorer reconstruction. Importantly, there is a good range of values wherein the system can be regarded as operating reasonably.

Number of components. Choosing K, also known as order estimation, is typically data and application dependent. It controls the granularity of the discovered audio spectral patterns. Determining the optimal value has been a long standing problem within the NMF community (Tan and Févotte, 2012). Our choice for this parameter was guided by three main factors:

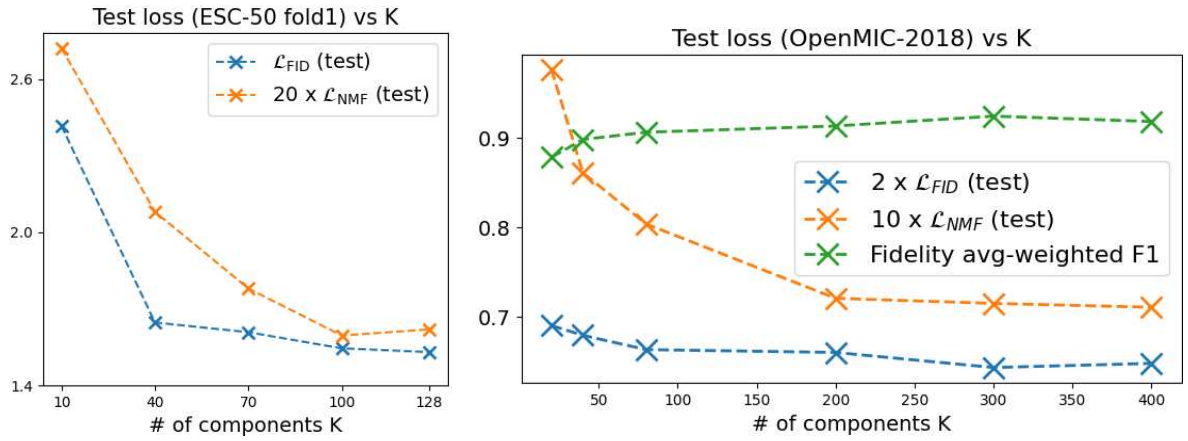


Figure 5.6: Ablation study for number of components. Loss values on test data for ESC-50 and OpenMIC-2018.

- Choices made previously in literature for similar pre-learning of **W** (Bisot et al., 2017), who demonstrated reasonable acoustic scene classification results with a dictionary size of $K = 128$. We used this as a reference to guide our choice.

- Dataset specific details which include number of classes, samples for each class, variability of recordings etc. For eg. acoustic variability of ESC-50 (larger number of classes), prompted us to use a dictionary of larger size compared to SONYC-UST. We use highest number of components for OpenMIC, which has largest dataset size among the three and reasonably high acoustic variability.

- When tracking loss values for different K, we observed a plateauing effect for larger dictionary sizes as illustrated in Fig. 5.6 for ESC-50 and OpenMIC-2018. In case of OpenMIC, this effect is prominent for reconstruction loss $\mathcal{L}_{\text{NMF}}$. The fidelity remains high even for small K .

5.5.4 By-design interpretation

The performance of all systems is given in Tab. 5.8. We compute the same metrics as used to evaluate the classifiers for each dataset. Mean performance along with variance across 3 runs is reported. We make the following key observations:

- Among the **interpretable neural networks for audio**, L2I_{BD} w/ Θ_{ATT} clearly outperforms APNet. The size of the models plays an important role in this. L2I learns with the help of a network architecture that feeds it with higher quality representations for prediction compared to architecture in APNet. It is generally able to sustain a comparable performance w.r.t the base network BASE-f while imposing an interpretable structure for final prediction model.

System	ESC-50 (in %) accuracy	SONYC-UST macro-AUPRC	OpenMIC-2018 avg-weighted-F1
$L2I_{BD}$ w/ Θ_{ATT}	**70.1 ± 1.5**	**0.581 + 0.008**	**0.825 ± 0.005**
APNet (Zinemanas et al., 2021)	63.6 ± 1.7	0.422 ± 0.012	0.563 ± 0.025
Unsupervised-NMF	39.4 ± 2.3	0.373 ± 0.006	0.659 ± 0.018
TD-NMF (Bisot et al., 2017)	46.7 ± 2.7	0.431 ± 0.018	0.699 ± 0.012
L2I–Posthoc	65.4 ± 3.4	0.567 ± 0.007	**0.825 ± 0.003**
$L2I_{BD}$–NoPred	64.4 ± 1.1	0.563 ± 0.004	0.746 ± 0.006
FLINT	75.3 ± 3.6	0.556 ± 0.008	0.827 ± 0.002
BASE-f	82.5	0.601	0.831

Table 5.8: Classification performance for by-design interpretation. The evaluated systems include our proposed by-design interpretable network, denoted as $L2I_{BD}$ w/ Θ_{ATT}, its variant with modified loss function ($L2I_{BD}$–NoPred), interpreter trained for post-hoc interpretation (L2I–Posthoc), classification models based on traditional NMF (unsupervised NMF and TD-NMF) and audio prototypical network APNet. The base classification network used for post-hoc interpretation (BASE-f) and FLINT are used as references for high performance networks not suitable for audio interpretation.

- **Comparison with NMF baselines**. While TDL-NMF performs better than unsupervised-NMF, L2I variants are noticeably better than both. This highlights a unique advantage of combining NMF representations with deep neural network representations, wherein, the NMF structure leads to interpretability and using deep networks as source provides higher prediction performance compared to directly using NMF activations generated from input.

- We also validate our design of training procedure for by-design interpretable network $g(x)$, by comparing it with the two variants of proposed system, L2I-PostHoc and L2I-NoPred. The performance of $L2I_{BD}$ w/ Θ_{ATT} compared to L2I-Posthoc highlights that $g(x)$ tends to perform better when hidden layers of f are trained jointly with interpreter. L2I-NoPred performs the worst among the three, emphasizing the benefits of updating the hidden layers of f with classification loss imposed on $f(x)$ rather than on $g(x)$.

5.6 Conclusion

We have presented a framework to tackle both post-hoc and by-design for audio classification networks. To this end, a novel interpreter is designed with the key idea of using an NMF-inspired regularizer. This enables listenable concept-based interpretations. We motivate listenability as an important attribute for audio interpretability. Efficacy of the proposed framework is established through extensive qualitative and quantitative experimentation. In particular, we quantitatively evaluate both post-hoc

and by-design interpretations on three popular datasets pertaining to audio event and music instrument recognition tasks. We perform a user-study to confirm usefulness of our interpretations. In addition, through a visualization of the generated interpretations, we show that they are coherent across samples from different classes and cluster in a fashion that aligns well with human understanding of sound. Further works concern the extension of this framework to other machine learning audio-based tasks.

Contents

6 Perspectives

Contents

The goal of this chapter is to gain perspectives about the research presented in this thesis. We first analyze the research with a broad outlook and identify the contributions. We then discuss the limitations and research directions that can deepen and broaden the abilities of the developed framework. A specific section is devoted to discuss ongoing work regarding extending the decoder for the image interpretability system.

6.1 Discussion and contributions

6.1.1 Summary of contributions

We first designed a framework in Chapter 3, that can be used to address both post-hoc and by-design interpretation problems. The framework is based on the idea of considering prediction and interpretation as two separate tasks solved by different but related models, a predictor f and an interpreter g. The interpreter is dependent on f through its selected hidden layers. We proposed a single learning objective

composed of a prediction loss term and an interpretability loss term. To expand upon the structure of g, first we motivated the idea of using a dictionary of concepts as representation of interpretation, learnt without any additional supervision. Then we proposed a minimal set of properties which formulate the interpretability loss for training. We finished outlining details about g by proposing a novel notion of local and global relevance based on activation of attribute functions and how they affect the output of g.

We presented two instantiations of the framework, for image classification networks in Chapter 4 (titled FLINT) and audio classification networks in Chapter 5 (titled L2I). For both tasks, we demonstrated the applicability to post-hoc and by-design interpretation. We showed improved prediction performance of g with respect to comparable interpretable networks on multiple popular classification benchmarks in both cases. We also illustrated better fidelity of our interpreter compared to black-box interpreters, highlighting the advantages of utilizing hidden layers for interpreter's representation. Additionally, we thoroughly analyzed the interpretations qualitatively, along with a subjective evaluation to assess their understandability in both instances.

Modality specific contributions: For image modality, we proposed an entropy based loss to improve the conciseness of interpretations compared to other unsupervised concept-based approaches and propose a novel pipeline to discover the encoded concepts. The pipeline was designed to delve deeper into understanding what visual patterns activated an element of concept dictionary. For audio modality, additionally motivated by listenability of interpretations, we instead proposed a completely novel method for interpretation, based on popular non-negative matrix factorization (NMF). This method also presented a novel way to link NMF and deep neural network representations. We also proposed a mechanism to compute faithfulness of interpretation for post-hoc interpretations in this setting which is non-trivial for unsupervised concept-based methods. Furthermore, we illustrated that our system in most cases remains faithful to the predictor.

6.1.2 Contrasting between FLINT and L2I

It is quite fascinating to study the differences between the two instantiations, especially since the underlying framework is identical for FLINT and L2I. Our motivations in both applications are slightly different. Consequently, the systems are realized through different representations for interpretation, even though both learn unsupervised dictionaries which capture higher-level features. These differences ultimately lead to distinct information given to the user as interpretation even though the process of relevance computation is similar and central to both. The focus of evaluation also slightly differs in the two applications. We now elaborate on these points below:

- **Listenability as motivation**: A critical desideratum that set the foundation of L2I was listenability of interpretations. This was linked to the observation that any visualization is significantly less meaningful for audio. The representation

used in FLINT was ruled out as the activation maximization based visualizations were poor for generating listenable audio.

- **Representation for interpretation**: There are a few intriguing differences one can observe between the two representations and the interpreters for FLINT and L2I. The representation in FLINT, $\Phi_{\mathcal{I}}(x)$, is completely learnt from scratch whereas the representation in L2I, $\mathbf{H}_{\mathcal{I}}(x)$, has a separate component tied to it, the underlying dictionary $\mathbf{W}$, which is learnt separately on the input data and independent of the networks. The hidden layers of the predictor are then processed to learn which of the components of $\mathbf{W}$ need to be activated, in line with all interpretability losses. In this sense, the use of Sparse-NMF to learn $\mathbf{W}$ is similar to the use of external algorithms in concept extraction of ACE (Ghorbani et al., 2019). However, unlike in ACE, the dictionary learning in L2I is not limited by the representation power of a pre-trained network. The dictionary is learnt independently of any neural network. Another intriguing difference is the completely different designs of decoder. L2I employs the pre-learnt dictionary $\mathbf{W}$ as decoder whereas FLINT employs a neural network as decoder. The operation of L2I-decoder is well defined and interpretable in itself. This design correspondingly imposes a very clear meaning of time activations on the learnt representation $\mathbf{H}_{\mathcal{I}}(x)$. Conversely in FLINT we develop a separate pipeline to discover the encoded concepts in the attribute dictionary $\Phi_{\mathcal{I}}$. The decoder design of L2I plays a central role in allowing the possibility of a filtering procedure on input audio. Moreover, the generative nature also allowed the possibility to simulate removal of components, essential to compute faithfulness for post-hoc interpretations.

- **Interpretation generation**: Despite the clear meaning enforced on the learnt representation in L2I, one can notice that we do not explicitly attempt to derive any semantic understanding for each individual component in $\mathbf{W}$. This is mainly because the components in L2I, generally do not encode the same level of abstract information as in FLINT, or as one would require for high-level concepts. This does not imply that a global interpretation cannot be generated. One could still meaningfully compute global relevances in L2I. The visualization of relevances in some sense also is a type of global interpretation. However, it is hard to expect one single component to capture an abstract concept. Interestingly though there is a possibility that a group of components collectively represent a concept. We leave the research in this direction as a potential future work. Lastly, the interpretation generation algorithm in L2I, while more straightforward and simpler than FLINT due to decoder design, doesn't provide the same level of information or detail. Similar to other interpretability algorithms, the information served as interpretation in L2I focuses to highlight relevant regions of input, albeit with a completely different methodology. On the other hand the pipeline in FLINT segregates finer details about the input as part of interpretation rather than highlighting region of relevance.

6.2 Limitations and future work

We divide the discussion about limitations and interesting research directions for future in five parts. The first three discuss them pertaining to our general framework design, and specific systems for image or audio. The ideas elaborated in these parts aim to deepen the framework developed. In the fourth part we discuss directions to broaden the scope of our framework. In the final part we conclude the discussion by covering key themes of this thesis with implications beyond our scope of research.

6.2.1 Framework design prospects

Hyperparameter selection: Invariably any instance of the framework would require selection of hidden layers for the interpreter to access, size of dictionary $\Phi_{\mathcal{I}}$ and the loss hyperparameters. While we do study and provide empirical and practical guidelines to make a smart choice of these parameters, they do require certain background knowledge about the networks and behaviour of the losses to reach a balance.

Interpretability loss function: As of yet, we only propose a minimal set of properties and corresponding penalties to learn $\Phi_{\mathcal{I}}(x)$. Flexibility in using different loss functions to impose other properties offers an attractive direction to improve the learning. We already invoked the possibility of imposing stability of activations with respect to changes in the input. Orthogonality of components/attribute functions also poses an interesting direction in this regard. It can be a useful constraint for data decomposition (Asteris et al., 2015) having found applications for both audio and image processing. Its frequently used in vision for encoding disentangled information (Sarhan et al., 2020; Liu et al., 2020), and in audio processing for regularizing NMF dictionaries (Sobieraj et al., 2018; Kitamura et al., 2014).

Another compelling direction to explore is that of imposing invariance of activations to transformations. Invariances in general have natural links to our understanding of the physical world. For example, our brains can identify an image of a dog even if it's upside down, and our reasons for identifying it remain the same. In other words our "internal concept representation" of the image remains invariant to orientation of the image. Imposing such properties can guide the learning of the attribute functions to improve interpretability aspects. While some invariances are imposed by-design through an architectures (for eg. translation invariance in CNNs), depending upon the transformation, the invariance can result in better semantic structure (Rieger et al., 2020).

Innovating with architectures: Our framework employs three different functions with three different roles: (a) Ψ with role of processing the hidden layers and predicting the representation for interpretation, (b) Θ tasked with processing output of Ψ and predicting the final output of interpreter g, and (c) Decoder d tasked with processing output of Ψ and reconstructing the input to encourage the representation to encode relevant patterns of the input space. Each of them should be designed according to a given application. However, Θ has more general possibilities of improvement.

There are generic designs (Agarwal et al., 2020; Radenovic et al., 2022) that can replace the use of linear layer in current versions of Θ to improve fidelity/performance without giving up much on interpretability, or introduce a novel usage of attributes in interpreter's prediction that benefits certain use-cases, such as logical reasoning over the attribute functions (Kusters et al., 2022).

6.2.2 Image interpretability directions

Decoder limitations and extensions. Even though we demonstrate that FLINT can provide useful insights for large scale and diverse images, the learnt attributes and concept discovery pipeline to understand them have limitations that stand out more for complex problems. For the concept discovery pipeline, one key limitation is the lack of interactive nature. While activation-maximization based outputs can assist in answering what concept is encoded in an attribute function, they do not necessarily localize the activating visual patterns. This makes it more challenging for the user to identify relevant information in the visualization. Interactive tools to link an attribute to input space give the user a better handle to simultaneously localize while emphasizing an encoded concept.

The reduction of semantic meaningfulness of the learnt Φ in complex scenarios (for eg. in CUB-200) is another key limitation. Note that understandability of features captured by an attribute does not exactly equate to it being semantically meaningful. An attribute capturing multiple semantic features can still be deemed understandable even though its not semantically meaningful. However undoubtedly both notions are related as human understanding about high-level concepts is an underlying factor for both. Both of the aforementioned limitations can potentially be addressed through the use of a decoder based on a generative model like Variational-Autoencoder (VAE) (Kingma and Welling, 2013) or Generative Adversarial Networks (GAN) (Goodfellow et al., 2020). We devote the next section to it, as it is one of the ongoing contributions.

Incorporating prior knowledge: Currently there is no mechanism to integrate any prior knowledge in the learning of attributes. Recent work by Sarkar et al. (2022) considers the case of full supervision of concept dictionary with complete annotations for each sample, which we believe could be adapted to FLINT. There are however no systems to have demonstrated concept dictionary learning in a semi-supervised setting i.e. annotations for a subset of concepts. Work in this direction could enhance the flexibility and use-cases of the method.

6.2.3 Audio interpretability directions

As previously mentioned, the aspect of global interpretations and possibilities of encoding abstract concepts have not been studied to full extent in L2I currently. While it is unlikely that a single component encodes high-level properties about the input, but it is possible that a subset of components do. Moreover, the current components in $\mathbf{W}$ only capture frequency fingerprints for single time-frames. There are frequency

patterns that can exist over multiple time frames which the current **W** is unable to capture.

The above discussion suggests that the current L2I interpretation need not be the most favourable for all types of audio tasks. Our motivation indicates the scenarios where L2I is more suitable, i.e. when the input scene can be decomposed into multiple audio sources. Emphasizing one or a subset of sources for listenable audio is insightful in these cases. For tasks where the mapping between categories and underlying audio object/events is not as clear, L2I's representation for interpretation is probably not optimal. Thus, extending **W** using the notions of convolutive-NMF (Bisot et al., 2017), or non-negative tensor decomposition (NTD) is a promising direction (Foscarin et al., 2022). Studying the possibilities of a set of components encoding an abstract concept is possibly worth pursuing. Furthermore, devising new ways to build a dictionary with other ways of sample selection procedures or loss functions to impart some prior knowledge are also interesting directions.

6.2.4 Breadth in current framework

The current research can be extended and studied in many different directions. The framework offers flexibility in its ability to address both post-hoc and by-design interpretation and scope of interpretability (local and global). This in itself incentivizes application to novel problem settings. The other vital ingredient of the framework is the non input attribution based representation for interpretability. Similar to our application on audio, developing and analyzing novel representations in new domains offers a wide range of interesting possibilities in broadening the framework's applicability. We go through some of these below:

- **Data modalities**: This manuscript primarily concerns with image and audio signals. Nevertheless, there are multitude of other data modalities that benefit in performance of classification systems by employing deep neural networks. Moreover, there might be potential advantages in relying on concept-based interpretations on these modalities. Graphs, time-series data, multi-modal data (including videos) are some prominent options for the same. Research on interpretability in graph processing models have already invested efforts in proposing novel means of interpretations more suited for graphs (Yuan et al., 2021) as covered in Chapter 2. Multi-modal data includes two popular pairs of modalities, image–text and image–audio. There is a strong potential to propose more convenient means of interpretation for them, for eg. by using higher-order objects with simultaneous grounding in both domains.

- **Learning tasks and predictor architectures**: Our framework was developed for supervised classification as the underlying task. Our experiments included 3 different types of CNNs across the two modalities, given their popularity for image and audio classification. However, our design has generic elements which we believe could be adapted to novel tasks such as supervised regression, rein-

forcement learning, structured prediction. Instead the architectures and losses can also be adapted to specific networks including transformers, structured prediction energy networks (SPEN) or other energy based models.

- **Prototypical representation**: A commonality between methods for prototype based and concept-based interpretations is their reliance on latent representations. This leads us to believe that the framework could be potentially enhanced even more by incorporating prototypes as means of interpretation. For instance, our framework could offer a pathway to apply prototypical networks for post-hoc interpretations.

6.2.5 General research directions for interpretability

Evaluation and Comparisons. A key theme in this research has been about the choice of means of interpretation for any method. It directly impacts what information humans need to assess and understand. Given the different possible choices, this raises the question of how can the systems using different representations for interpretation can be compared. A possibility which we ourselves opted for was human evaluation for audio interpretation (L2I vs SLIME). However, in terms of quantitative metrics or functionally-grounded evaluation, it is not a given that two different methods are comparable. For example, saliency maps and prototype-based interpretation methods would measure conciseness in different ways. Even if they measured it in the same way, it is not obvious that comparison is sensible. Thus, proposal of metrics or platforms that can cost-effectively compare between systems with different means of interpretation is an important aspect of research for near future.

Non-technical context factors. When listing out the various context factors for an interpretability application in chapter 2, we only considered the technical details. With the close relation of interpretability to social, cognitive and legal spheres (Chatila et al., 2021; Bertrand et al., 2022; Brand, 2022), there are a host of other constraints that we haven't considered. And unavoidably all the context factors (including technical ones) interact with each other to form a complex web of relationships. For any real-world application of interpretability the influence of all these relationships need to be considered and studied to ensure ethical and beneficial use of these systems

Interpretability as intermediate task. Throughout the thesis we consider interpretability as an end goal, in line with spirit of most of the recent literature. Nonetheless, it is very intriguing to consider it as an intermediary for other downstream tasks such as model debugging, active learning, anomaly detection etc. This is to some extent a natural progression for future as mastering a skill is typically followed by its use to solve other problems. Research maturity in interpretability and its understanding among the community will inevitably push the frontiers in this direction.

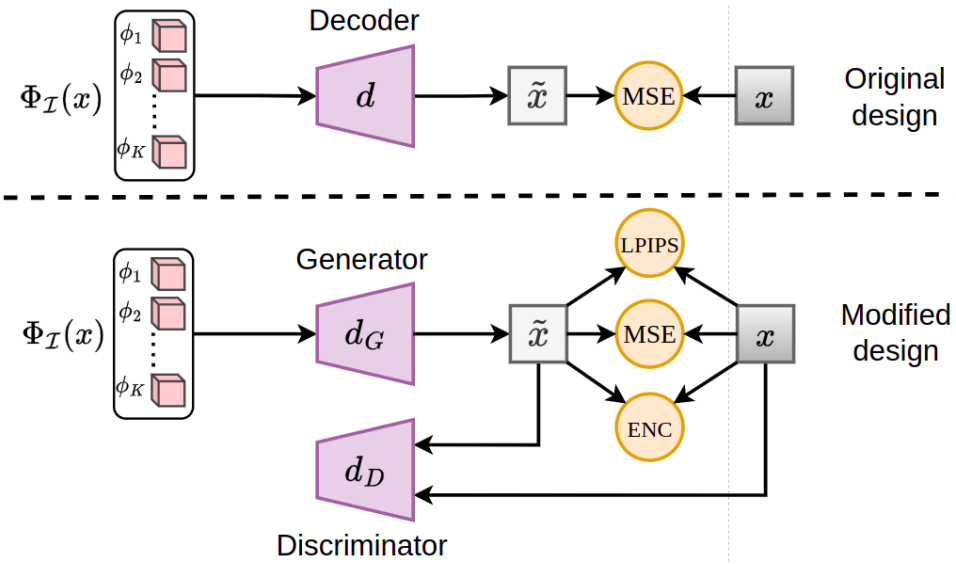


Figure 6.1: Modification of decoder architecture and input fidelity loss terms. The indicated 'MSE', 'LPIPS', 'ENC' losses all promote $\tilde{x}$ to be close to x (Eq. 6.3.1)

6.3 Extending FLINT decoder with generative models

As mentioned previously, there is ongoing work to incorporate generative modelling in the decoder d for two goals, to increase semantic meaningfulness in learning of dictionary $\Phi_{\mathcal{I}}$ (while maintaining relatively small dictionary size) and adding interactivity in the visualization for better understanding. A secondary advantage of generative modelling is the possibility to measure faithfulness, as witnessed for L2I in Chapter 5. There are multiple possibilities for popular generative models in machine learning research, which includes variational autoencoders (VAE) (Kingma and Welling, 2013), generative adversarial networks (GAN) (Goodfellow et al., 2020), invertible neural networks (INN) or flow-based models (Rombach et al., 2020) and diffusion models (Ho et al., 2020). However, given the much smaller dimensionality of latent space compared to input and desirability of fast generation time, VAE or GAN are currently more viable options. There are some recent works that learn unsupervised concept dictionaries using a VAE (Taeb et al., 2022) and StyleGAN (Lang et al., 2021). However, due to GAN's common reputation to generate sharper images at high resolutions, we gravitate more towards opting them, even though training VAE's is relatively much easier.

We would like to emphasize that this research direction is currently being investigated. We share some preliminary qualitative results that seem promising.

6.3.1 System design

We intend to replace the decoder part with a generator as shown in Fig. 6.1. However, as expected, this results in a discriminator also being incorporated to help train the generator. We denote the generator and discriminator being subsumed under the

decoder as d_G, d_D respectively. Given that in the recent work StyleGAN and its variants are essentially among the state-of-the-art models for image generation and disentangled representation learning (Karras et al., 2020b, 2021), we opt for its generators and discriminators architectures for d_G and d_D respectively. The key loss function modified is the input fidelity loss $\mathcal{L}_{if}$. Earlier it was simply implemented as mean-squared error reconstruction. However, taking inspiration from Lang et al. (2021), we incorporate three separate terms to promote reconstruction. The new $\mathcal{L}_{if}$ writes as the following:

$$
\begin{aligned}
&\tilde{x} = d_G(\Phi_{\mathcal{I}}(x)) = d_G \circ \Psi \circ f_{\mathcal{I}}(x) \\
&\mathcal{L}_{mse}(\tilde{x}, x) = \|\tilde{x} - x\|_2^2 \\
&\mathcal{L}_{enc}(\tilde{x}, x) = \|\Phi_{\mathcal{I}}(\tilde{x}) - \Phi_{\mathcal{I}}(x)\|_1 \\
&\mathcal{L}_{if}(x, g, d) = \mathcal{L}_{mse}(\tilde{x}, x) + \mathcal{L}_{\text{LPIPS}}(\tilde{x}, x) + \mathcal{L}_{enc}(\tilde{x}, x) \\
&\mathcal{L}_{int}(x, g, d) = \mathcal{L}_{of}(x, g) + \beta \mathcal{L}_{if}(x, g, d) + \gamma \|\Phi_{\mathcal{I}}(x)\|_1
\end{aligned}
$$

The $\mathcal{L}_{\text{LPIPS}}$ computes the LPIPS distance (for perceptual similarity) between the images, based on pretrained network embeddings (Zhang et al., 2018c). Among other loss functions for training, there is an adversarial loss to train the generator and discriminator (Arjovsky et al., 2017). We also drop the entropy terms for conciseness and simply impose an ℓ_1 regularization. We take this step for the time being since improving conciseness is not our primary objective here. Moreover, it gets rid of an additional loss hyperparameter. The fidelity loss remains unchanged.

StyleGAN methodological challenge: A unique aspect about the StyleGAN generator architecture is that it employs two sub-components, a mapping network to map a vector from noise distribution to a vector in 'Style-space' (also known as $\mathcal{W}$-space), and a synthesis network to synthesize the image from the 'style vector'. The $\mathcal{W}$ space essentially captures the latent factors of variation. Ideally, one would like to learn $\Phi_{\mathcal{I}}$ to be identical to the $\mathcal{W}$-space but it is an unreasonable task for two reasons. $\Phi_{\mathcal{I}}(x)$ is generated from hidden layers of a predictor also trained for classification. Thus, $f_{\mathcal{I}}(x)$ and $\Phi_{\mathcal{I}}(x)$ are both influenced to preserve features beneficial for classification. It is then hard to expect $\Phi_{\mathcal{I}}(x)$ to capture as rich a set of features as $\mathcal{W}$ even though it can encode some of them. The second reason is that the synthesis network uses a separate style-vector for each resolution. This again does not fit with our design as $\Phi_{\mathcal{I}}(x)$ is a single vector.

The methodological challenge for us is to determine a suitable way to map $\Phi_{\mathcal{I}}(x)$ to $\mathcal{W}$-space. For the moment we simply employ 2 fully-connected layers for this mapping. The original mapping network consists of 8 fully-connected layers to transform a noise vector to $\mathcal{W}$-space. Considering the desired similarity between $\Phi_{\mathcal{I}}$ and $\mathcal{W}$, we will consider a more structured way of this transformation. This might also require us to exclusively work with bigger architectures or access more hidden layers for a powerful backbone to predict $\Phi_{\mathcal{I}}(x)$.

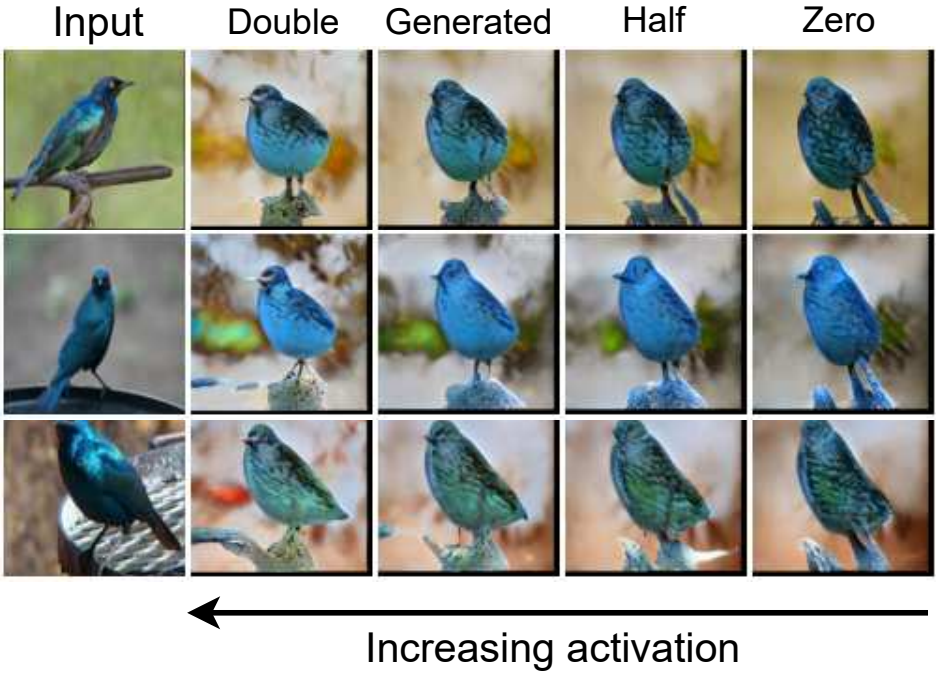


Figure 6.2: Interactive visualization of a relevant class-attribute pair ('Cape glossy starling', ϕ_{22}). The first column denotes the input images (3 maximum activating samples). The 2^{nd} to 5^{th} columns are all generated using d_G with different activation values of ϕ_{22} to clearly visualize the changes. 3^{rd} column contains the normal generated image with computed activation $\alpha = \phi_{22}(x)$. The $2^{nd}, 4^{th}, 5^{th}$ columns denote generated images with modified activations $\phi_{22}(x) = 2\alpha, \alpha/2, 0$ respectively.

6.3.2 Preliminary experiments

Our preliminary experiments are with CUB-200 birds classification dataset Wah et al. (2011) in a post-hoc interpretability setting. Starting with a post-hoc problem allows us to ascertain the quality of predictor representations for classification before training the interpreter. The dataset contains 11,788 images of 200 categories of birds, 5,994 for training and 5,794 for testing. We operate on a 256×256 resolution and use $J = 256$. We use ResNet18 as predictor architecture. The architectures of Ψ and Θ remain unchanged. Knowing the difficulty in training a GAN from scratch on limited data with relatively limited training resources compared to popular works, we initialize d_G and d_D using pre-trained weights of a StyleGAN2, trained on ImageNet. This has been empirically shown to help converge training faster (Grigoryev et al., 2022), although not specifically on CUB-200.

Our main goal here is to qualitatively illustrate the advantages of interactive visualization and identify the critical limitation for the current iteration of the system.

Qualitative illustration: We calculate the global relevances as before and select relevant class-attribute pairs with relevance $r_{k,c} > 0.5$. As a reminder, the relevance computation and selection procedure practically guarantees that the attribute ϕ_k generally

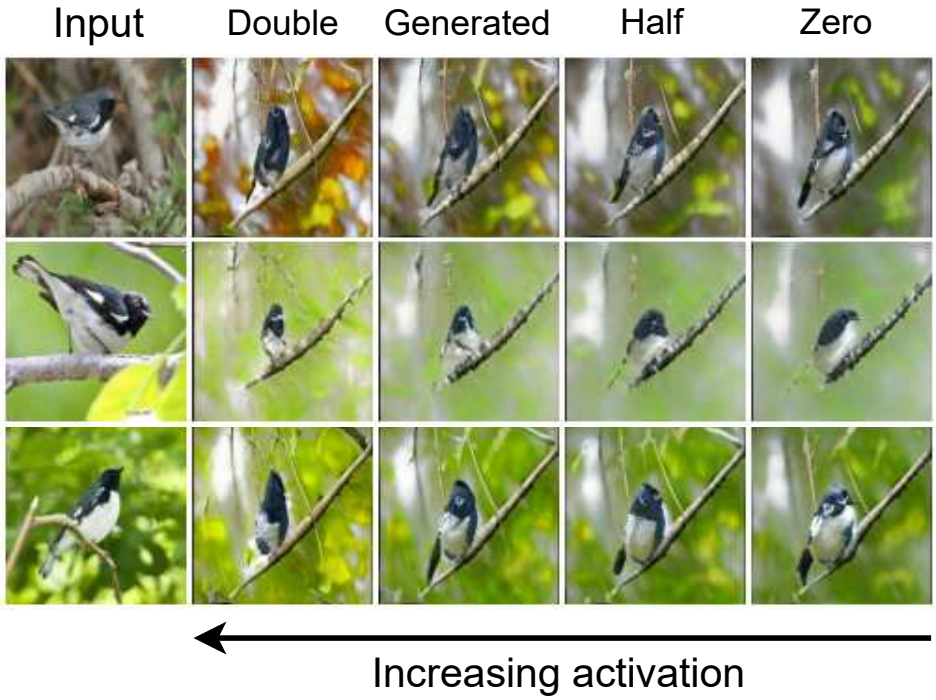


Figure 6.3: Interactive visualization of a globally relevant class-attribute pair ('Black throated blue warbler', ϕ_{12}). The first column denotes the input images (3 maximum activating samples). The 2^{nd} to 5^{th} columns are generated using d_G with different activation values of ϕ_{12} to clearly visualize the changes. 3^{rd} column contains the normal generated image with computed activation $\alpha = \phi_{12}(x)$. The $2^{nd}, 4^{th}, 5^{th}$ columns are generated with modified activations $\phi_{12}(x) = 2\alpha, \alpha/2, 0$ respectively.

plays an important role in interpreter's output for class c.

We deliberately select two specific class-attribute pairs which can illustrate the advantages of new pipeline to understand the encoded concepts. For each class-attribute pair we first its three maximum activating samples. The goal of the new pipeline is to specifically vary activation of the respective attribute for a given sample and view the generated samples to identify the visual changes in the image. The visualizations are given in Fig. 6.2 and Fig. 6.3. In both figures, the first column denotes the MAS of the attribute. The 3^{rd} column denotes the generated output for the given sample without modifying $\Phi_{\mathcal{I}}(x)$. For the remaining three samples, we generate output of d_G by first manually modifying the activation to twice, half and zero times its original value respectively. This results in increasing emphasis of the encoded concept (from right to left). For attribute ϕ_{22} in Fig. 6.2, the most visual change is the addition of blue color on the central/side part of the belly, indicating that the attribute detects this concept. For attribute ϕ_{12} in Fig. 6.3, the most visual change is the addition of blue colour in the neck region (just below the head and above the belly). This indicates that "blue neck color" is the primary encoded concept in ϕ_{12}.

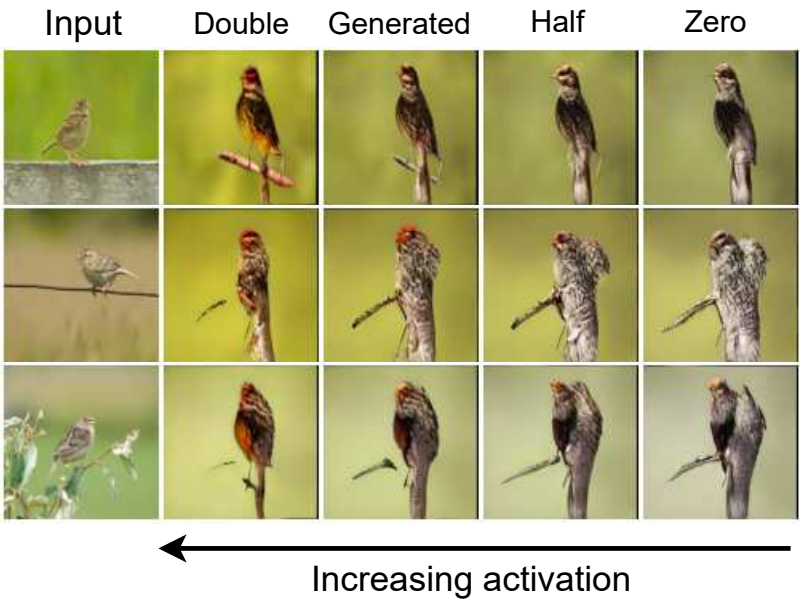


Figure 6.4: Issues in current system: Interactive visualization of a relevant class-attribute pair (ϕ_{28}, class 'Grasshopper sparrow'). The poor reconstruction does not ground the variations to a concept about the input.

Note this scheme is indeed similar to image editing schemes (Yao et al., 2021). However the key difference is that these attributes are specific to the underlying predictor/interpreter and determined to be important for prediction. Moreover, the user is discovering the encoded concept through the visualization rather than searching for features which encode a desired concept.

Limitations: However, the current iteration of this system is facing limitations and isn't properly functioning. One might notice that for both the attributes, the body shapes of generated birds also change. We believe this can be attributed to the current mapping of $\Phi_{\mathcal{I}}$ to $\mathcal{W}$ space wherein a single ϕ_k can influence multiple style-vector elements. Nevertheless, if the generated images can consistently reflect multiple changes, there is a case to be made that the attribute is understandable (even if entangled).

The much more critical limitation for the current system is the poor fidelity to input. This is already visible to some extent in the above two examples but we illustrate this clearly in Fig. 6.4 for a randomly selected class-attribute pair. The attribute ϕ_{28} for example consistently adds yellow and red colours to feathers and head with increasing activation in the generated outputs, but because the generated images are too different from the input image, these changes cannot be grounded w.r.t to the input image and thus the concept cannot be associated well. We quantitatively assess this through the average test LPIPS loss, which is also commonly used as a metric for perceptual similarity for GAN inversion (Yao et al., 2022). The current LPIPS-test loss is around 0.65, dropping from around 1.01 at the start of training. This is con-

siderably worse than values one typically encounters in the GAN inversion literature (upto 0.2). While our system does not target to achieve reconstruction quality close to state-of-the-art inversion models, high-fidelity reconstruction should significantly improve understandability of attributes.

Directions to address the limitations:

- **Updating architectures of f:** It is likely that improving the source representation to learn $\Phi_{\mathcal{I}}(x)$, $f_{\mathcal{I}}(x)$ would significantly help in tackling the poor reconstruction.
- **Updating training:** We currently face stability issues in training the GAN, wherein the generator frequently diverges around 50K-80K iterations. It is likely due to the relatively small size of the dataset, compared to ones typically used to train GANs. This has been a known issue with prior efforts to address it (Karras et al., 2020a). Path regularization loss (Karras et al., 2020b) or increased augmentation could partially address this problem but this needs to be assessed further
- **GAN inversion:** Another interesting option is to actually fix the pre-trained decoder to a version we know can model the data well and then learn to predict a $\Phi_{\mathcal{I}}$ and its mapping to $\mathcal{W}$ space that can reconstruct the input well. This would be similar to L2I (Chapter 5) where we trained a dictionary $\mathbf{W}$ on our data and used it as a fixed decoder. The ImageNet pretrained StyleGAN2 we use to initialize poses an attractive option in this regard. However, GAN inversion is not a straightforward problem and would require modifications to the current architecture (both f and Ψ) (Yao et al., 2022). This is a very interesting possibility as it can significantly ease the training resource requirements with no need to train d_G or d_D.

Conclusion

To summarize, this thesis operates within the field of interpretable machine learning, which revolves around generating human-understandable insights about decision process of machine learning models. In particular, we developed a single flexible framework in chapter 3 to address two most common classes of interpretability problems, post-hoc and by-design interpretation, specifically for neural networks. The former assumes a given predictive model and searches for algorithms to best interpret its decisions, while the latter is tasked to learn a single predictive model that is inherently interpretable. We accomplished our goal by formulating a single learning problem, that attends to both prediction and interpretation. We learn a predictor and a related interpreter to tackle the respective tasks via a single loss function consisting of a prediction and interpretation loss. This formulation is based on extending the traditional empirical risk minimization formulation for supervised learning and is termed as supervised learning with interpretation (SLI). The interpreter is related to the predictor by accessing its selected hidden layers. We developed the interpreters structure by opting for dictionary of concepts as its representation for interpretation, proposing a minimal set of properties with corresponding loss functions for its training and defining notion of local and global relevance for providing local and global interpretations.

We applied our framework for post-hoc and by-design interpretation in the context of image and audio classification in chapters 4 and 5 respectively. Owing to the common underlying framework, both systems exhibit improvements in predictive performance of interpreter and fidelity of interpretations. However, the two systems are designed with slightly different motivations leading to some separate individual contributions. The system for image interpretability proposes a novel pipeline to improve understandability of encoded concepts, which is qualitatively evaluated. Moreover, the system is trained with a novel entropy based criterion to lower conciseness/complexity of interpretations. The audio interpretability system is designed with a novel means of interpretation inspired from non-negative matrix factorization (NMF), to enable generation of listenable interpretations. It proposes a novel way of linking deep neural network with NMF that additionally grants it the capability to evaluate faithfulness for post-hoc interpretations.

We analyzed the contributions and noted the limitations for our proposed systems and framework in chapter 6. Furthermore, we examined directions to deepen and broaden their research expanse. This includes discussion on ongoing work to extend our image interpretability system using generative models. The extension aims to achieve greater semantic meaningfulness of learnt dictionary and an interactive concept visualization pipeline.

Appendices

G

Appendix for Chapter 4

Contents

G.1 By-design interpretation: Details and further analysis

G.1.1 Design details

Network architectures

Predictor Fig. G.1 and G.2 depict the architectures used for experiments with predictor architecture based on LeNet LeCun (2015) (on MNIST, Fashion-MNIST) and ResNet18 (on CIFAR10, QuickDraw) He et al. (2016) respectively.

Interpreter The architecture of interpreter $g = h \circ \Phi$ and decoder d for MNIST, FashionMNIST are shown in Fig. G.1. Corresponding architectures for QuickDraw are in Fig. G.2. For CIFAR-10, the interpreter architecture is almost exactly the same as QuickDraw, with only difference being output layer for $\Phi(x)$, which contains 36 attributes instead of 24. The decoder d also contains corresponding changes to input and output FC layers, with 36 dimensional input in first FC layer and 3072 dimensional output in last FC layer.

The choice of selection of intermediate layers is an interesting part of designing the interpreter. In case of LeNet, we select the output of final convolutional layer. For ResNet, while we tend to select the intermediate layers from the latter convolutional layers, we do not select the last convolutional block (CBlock 8) output. This is mainly because empirically, when selecting the output of CBlock 8, the attributes were trivially learnt, with only one attribute activating for any sample and attributes exclusively activating for a single class. The hyperparameters are much harder to tune to avoid this scenario. Thus we selected two outputs from CBlock 6, CBlock 7 as intermediate layers. The layers in the interpreter itself were chosen fairly straightforwardly with 1-2 conv layers followed by a pooling and fully-connected layer.

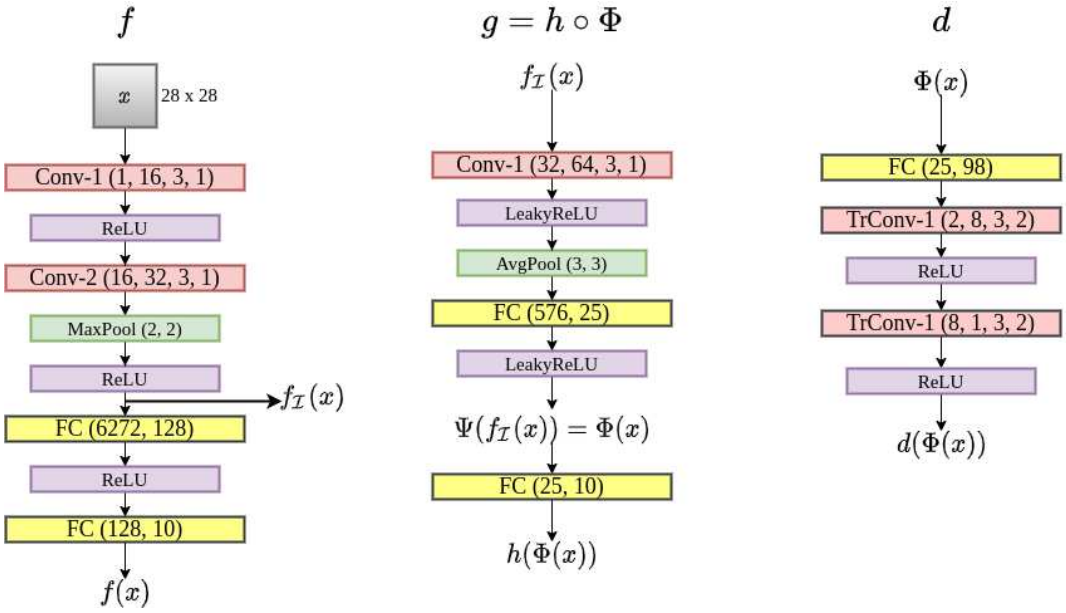


Figure G.1: Architecture of networks based on LeNet LeCun (2015). Conv (a, b, c, d) and TrConv (a, b, c, d) denote a convolutional, transposed convolutional layer respectively with number of input maps a, number of output maps b, kernel size c × c and stride size d. FC(a, b) denotes a fully-connected layer with number of input neurons a and output neurons b. MaxPool(a, a) denotes window size a × a for the max operation. AvgPool(a, a) denotes the output shape a × a for each input map

Optimization

The models are trained for 12 epochs on all datasets. We use Adam Kingma and Ba (2014) as the optimizer with fixed learning rate 0.0001 and train on a single NVIDIA-Tesla P100 GPU. Quantitative metrics on QuickDraw with ResNet are averaged across 3 runs for each set of parameters. Implementations are done using PyTorch Paszke et al. (2019).

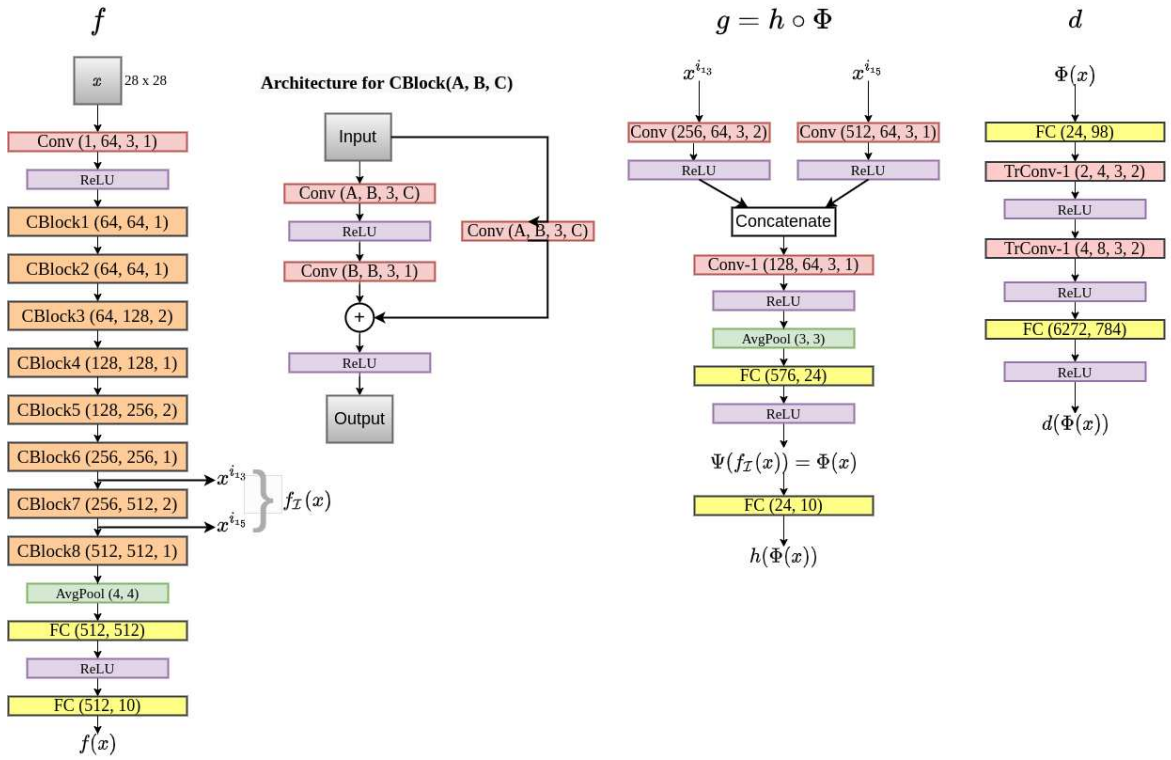


Figure G.2: Architecture of networks for experiments on QuickDraw with network based on ResNet He et al. (2016). Conv (a, b, c, d) and TrConv (a, b, c, d) denote a convolutional, transposed convolutional layer respectively with number of input maps a, number of output maps b, kernel size c × c and stride size d. FC(a, b) denotes a fully-connected layer with number of input neurons a and output neurons b. AvgPool(a, a) denotes the output shape a × a for each input map. Notation for CBlock is explained in the figure.

Hyperparameter tuning

For our experiments we set the number of attributes to $K = 25, 24$ for MNIST and QuickDraw, respectively. For MNIST with LeNet, we set $\zeta = 1, \mu = 1, \eta = 0.5, \delta = 0.2$, and for QuickDraw with ResNet, to emphasize conciseness less and diversity more we set $\zeta = 1, \mu = 2, \eta = 3, \delta = 0.1$. We employ $\beta = 0.1$ for QuickDraw and $\beta = 0.5$ for MNIST. It's slightly more tedious to tune γ. γ is varied between 0.8 to 20. We tune it so that the average value of $\mathcal{L}_{if}$ on $\mathcal{S}$ at least halves by the end of training. γ is set to 0.8 for MNIST and 5.0 for QuickDraw.

Choices for interpretation phase

For a random subset $\mathcal{S}_{rnd}$ consisting of 1000 samples from $\mathcal{S}$, we select class-attribute pairs which have $r_{k,c} > 0.1$ and use gradient as attribution method for LeNet based network and Guided Backpropagation Springenberg et al. (2014) for ResNet based

	$\eta = 1$	$\eta = 2$	$\eta = 3$	$\eta = 5$
$\zeta = 0$	92.7	90.4	91.2	84.2
$\zeta = 1$	91.2	90.7	90.8	82.9

Table G.1: Fidelity variation for η and entropy losses. $\delta = 0.1$ is fixed

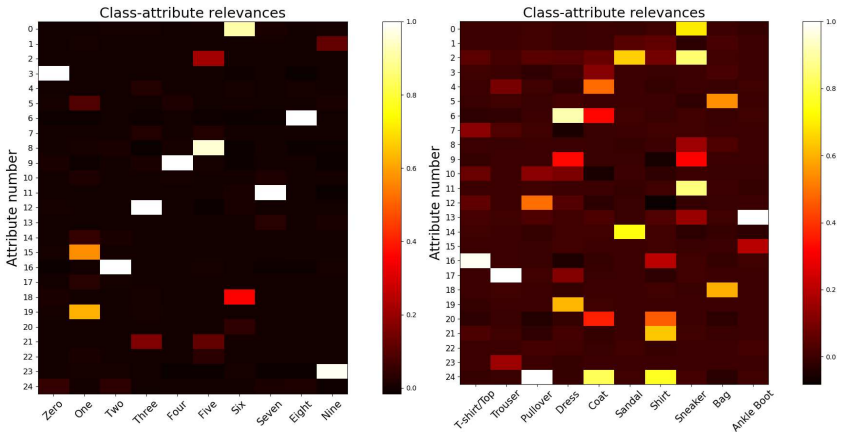


Figure G.3: Global class-attribute relevances $r_{k,c}$ for MNIST (Left) and FashionMNIST (Right). 14 class-attribute pairs for MNIST and 26 pairs for FashionMNIST have relevance $r_{k,c} > 0.2$.

network. We fix parameters for AM+PI for all our experiments as $\lambda_\phi = 2, \lambda_{tv} = 6, \lambda_{bo} = 10$ and for each sample x to be analyzed, we analyze input for this optimization as $0.1x$. For optimization, we use Adam with learning rate 0.05 for 300 iterations, halving learning rate every 50 iterations.

G.1.2 Additional visualizations

For completeness, we show some additional visualizations of global interpretations (relevances, class-attribute pairs) and local interpretations.

Fig. G.3 contains global relevances generated for MNIST and FashionMNIST. Global relevances for QuickDraw and CIFAR10 are in main paper.

Figs. G.4, G.5, G.6, G.7 show some additional class-attribute pairs and their visualizations for all 4 datasets. Local interpretations on some test samples from these datasets are depicted in Figs. G.8, G.9, G.10, G.11.

G.1.3 Other tools for analysis

Although we consider AM+PI as the primary tool for analyzing concepts encoded by attributes (for MAS of each class-attribute), other tools can also be helpful in deeper understanding of the attributes. We introduce two such tools:

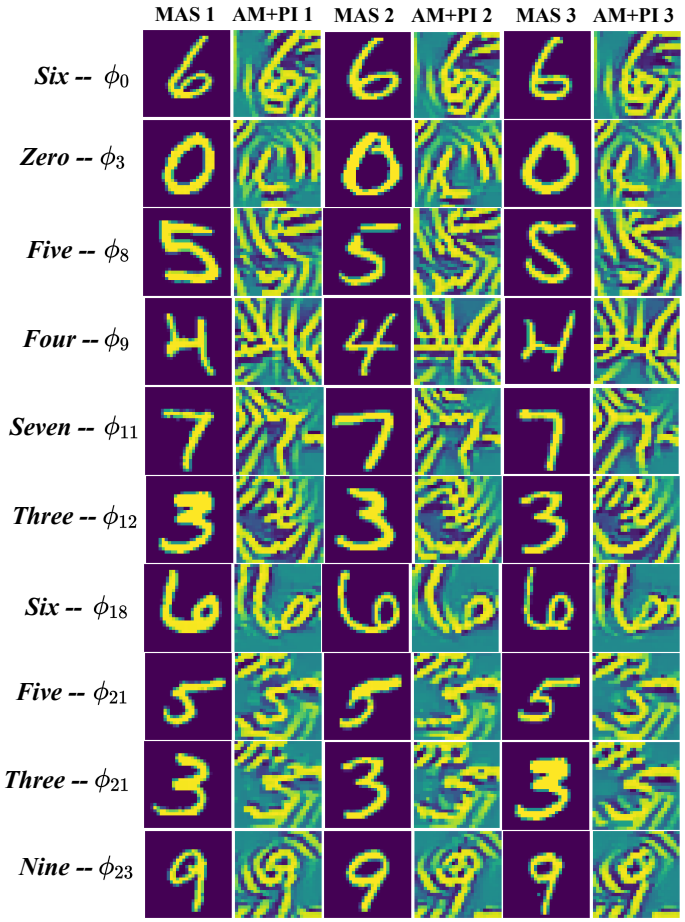


Figure G.4: Additional class-attribute visualizations for MNIST. Three MAS and their corresponding AM+PI outputs are shown.

- *Input attribution*: This is a natural choice to understand an attribute's action for a sample. Any algorithms ranging from black-box local explainers to saliency maps can be employed. These maps are less noisy (compared to AM+PI) and very general choice, applicable to almost all domains.

- *Decoder*: Since we also train a decoder d that uses the attributes as input. Thus, for an attribute j and x, we can compare the reconstructed samples $d(\Phi(x))$ and $d(\Phi(x)\backslash k)$ where $\Phi(x)\backslash k$ denotes attribute vector with $\phi_k(x) = 0$, i.e., removing the effect of attribute k. While, the above comparison can be helpful in revealing information encoded in attribute k, it is not guaranteed to do so as the attributes can be entangled.

We illustrate the use of these tools for certain example class-attribute pairs on Quick-Draw in Fig. G.12 and G.13. Note that as discussed in the main paper, these tools are not guaranteed to be always insightful, but their use can help in some cases.

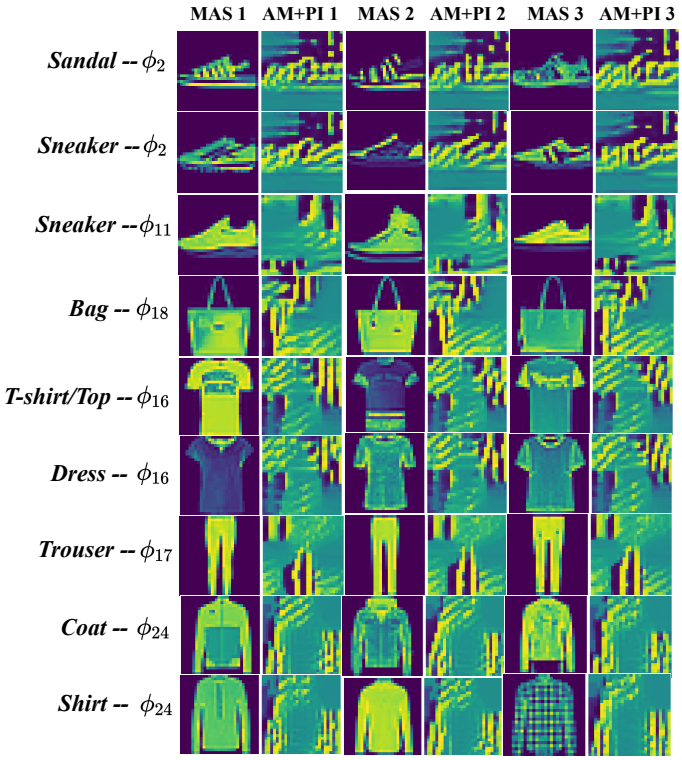


Figure G.5: Additional class-attribute visualizations for Fashion-MNIST. Three MAS and their corresponding AM+PI outputs are shown.

Fig. G.12 depicts example class-attribute pairs where decoder d contributes in understanding of attributes. The with ϕ_k column denotes the reconstructed sample $d(\Phi(x))$ for the maximum activating sample x under consideration. The without ϕ_k column is the reconstructed sample $d(\Phi(x))\backslash k)$ with the effect of attribute ϕ_k removed for the sample under consideration ($\phi_k(x) = 0$). For eg. ϕ_1, ϕ_{23}, strongly relevant for Cat class, detect similar patterns, primarily related to the face and ears of a cat. The decoder images suggest that ϕ_1 very likely is more responsible for detecting the left ear of cat and ϕ_{23}, the right ear. Similarly analyzing decoder images for ϕ_{22} in the third row reveals that it is likely has a preference for detecting heads present towards the right side of the image. This is certainly not the primary concept ϕ_{22} detects as it mainly detects blotted textures, but it certainly carries information about head location to the decoder.

Fig. G.13 depicts example class-attribute pairs where input attribution contributes in understanding of attributes. We use Guided Backpropagation Springenberg et al. (2014) (GBP) as input attribution method for ResNet on QuickDraw. It mainly assists in adding more support to our previously developed understanding of attributes. For eg., analyzing ϕ_5 (relevant for Dog, Lion) based on AM+PI outputs suggested that it mainly detects curves similar to dog ears. The GBP output support this understanding as the most salient regions of the map correspond to curves similar to dog ears.

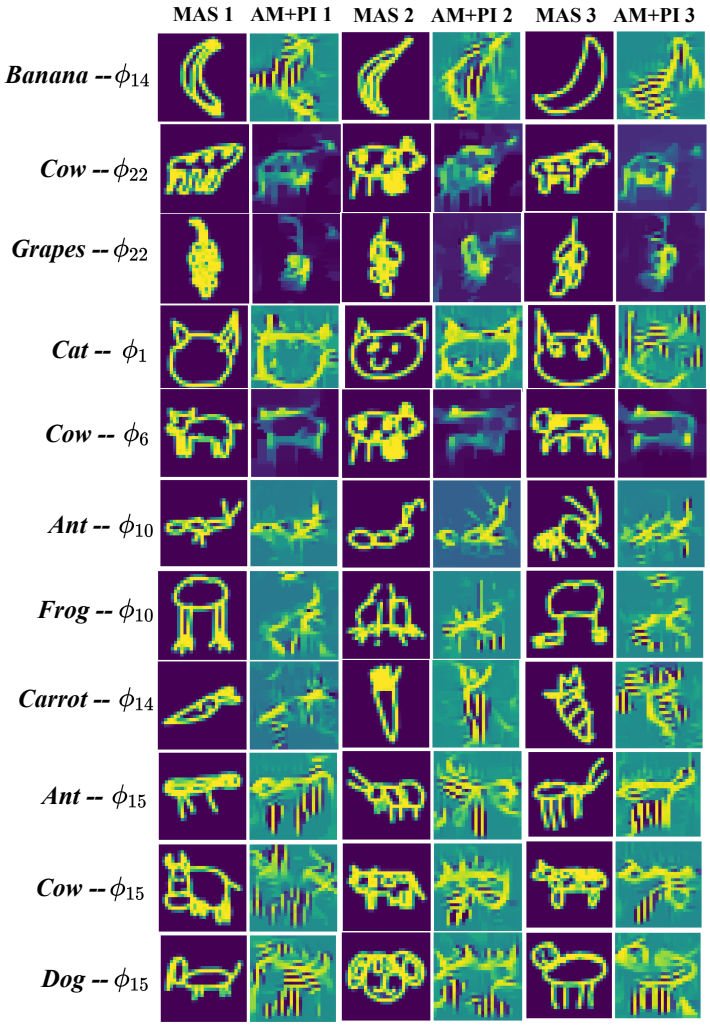


Figure G.6: Additional class-attribute visualizations for QuickDraw. Three MAS and their corresponding AM+PI outputs are shown.

G.1.4 Effect of autoencoder loss

Although the effect of $\mathcal{L}_{of}, \mathcal{L}_{cd}$ can be objectively assessed to some extent, the effect of $\mathcal{L}_{if}$ can only be seen subjectively. If the model is trained with $\gamma = 0$, the attributes still demonstrate high overlap, nice conciseness. However, it becomes much harder to understand concepts encoded by them. For majority of attributes, MAS and the outputs of the analysis tools do not show any consistency of detected pattern. Two such attributes are depicted in Fig. G.14. Attributes like these can be learnt even for the model trained with autoencoder, but are quite rare. We thus believe that autoencoder loss enforces a consistency in detected patterns for attributes. It does not necessarily guarantee semantic meaningfulness in attributes, however it's still important for encoding meaningful patterns about the input.

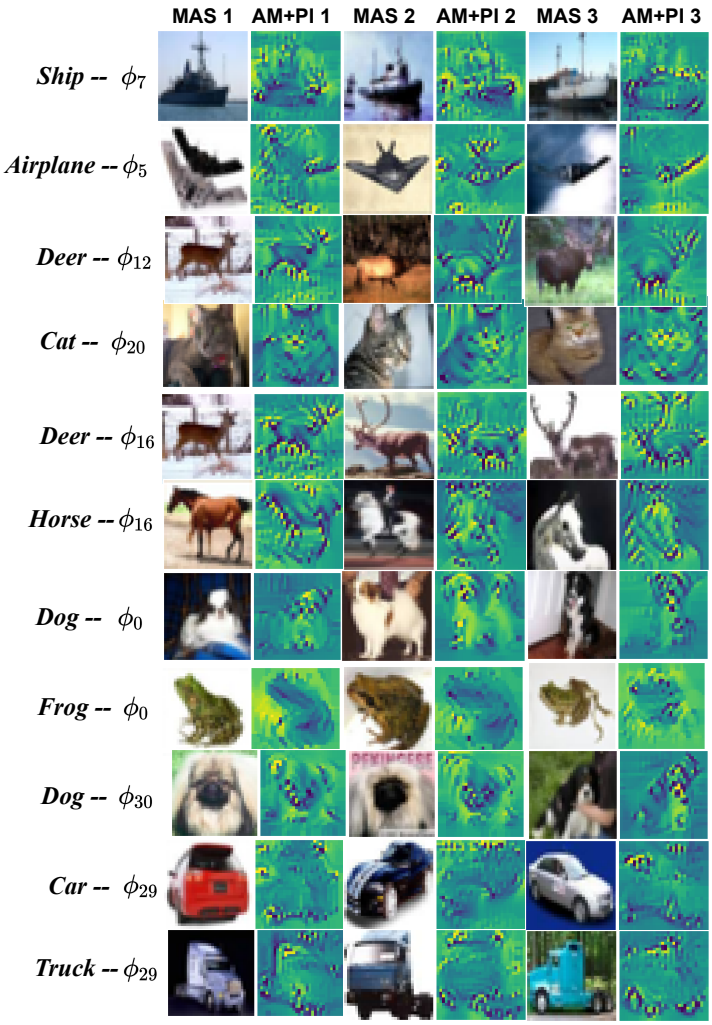


Figure G.7: Additional class-attribute visualizations for CIFAR-10. Three MAS and their corresponding AM+PI outputs are shown.

G.1.5 Baseline implementations

We cover the implementation details of various baselines used in this work (Tab 2, 3, 4 from main paper). As stated in the main paper, implementation of our method is available on Github [1].The accuracy of FLINT-f is compared against BASE-f, PrototypeDNN, SENN. Fidelity of FLINT-g is compared against VIBI and LIME.

BASE-f We compare accuracy of FLINT-f with BASE-f. The BASE-f model has the same architecture as FLINT-f but is trained with $\beta, \gamma, \delta = 0$, that is, only with the loss $\mathcal{L}_{pred}$ and not interpretability loss term. All the experimental settings while training this model are same as FLINT.

[1] https://github.com/jayneelparekh/FLINT

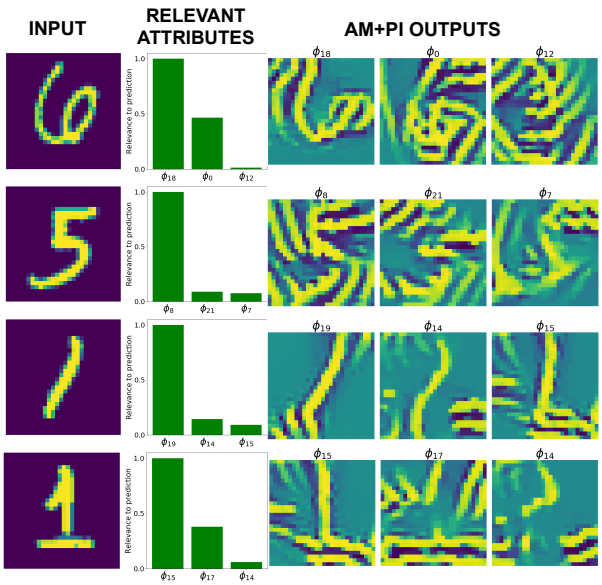


Figure G.8: Local interpretations on test samples for MNIST. True labels are: 'Six', 'Five', 'One' and 'One'. Top 3 most relevant attributes and their corresponding AM+PI outputs are shown.

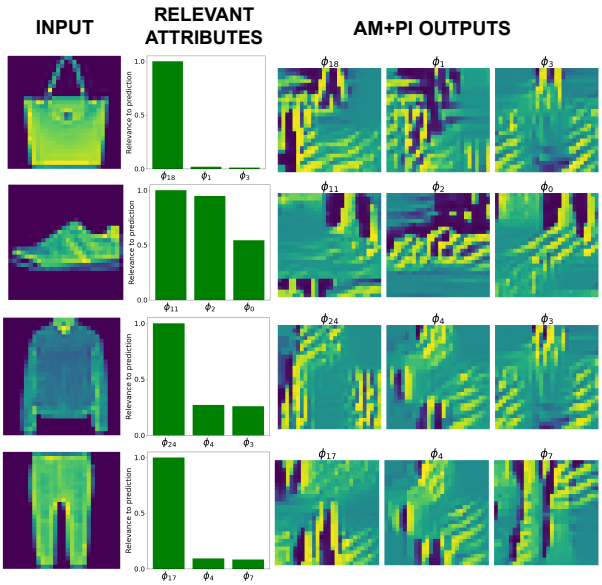


Figure G.9: Local interpretations on test samples for Fashion-MNIST. True labels are: 'Bag', 'Sneaker, 'Coat', 'Trousers'. Top 3 most relevant attributes and their corresponding AM+PI outputs are shown.

PrototypeDNN We directly report the accuracy of PrototypeDNN on MNIST, FashionMNIST (Tab 2 main paper) from the results mentioned in their paper Li et al. (2018). Note that we do not report any results of PrototypeDNN on CIFAR10 and QuickDraw. This is because for processing more complex images and achieving higher accuracy, one would need to non-trivially modify architecture of their proposed model.

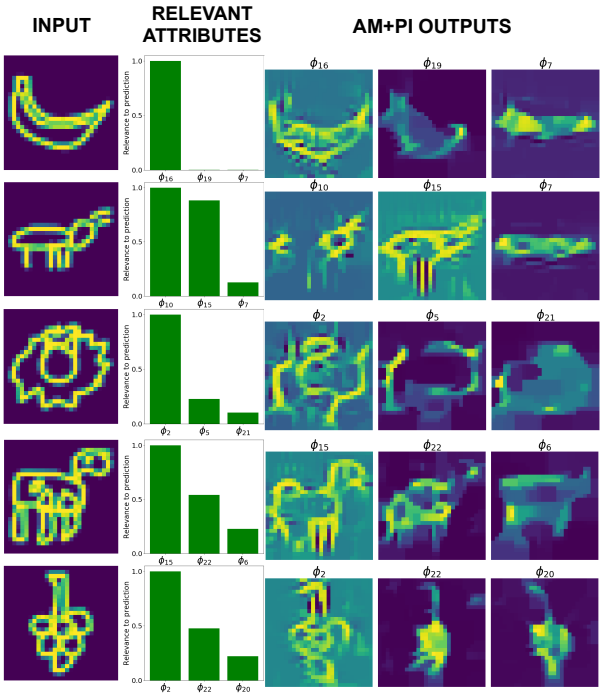


Figure G.10: Local interpretations on test samples for QuickDraw. True labels are: 'Banana', 'Ant', 'Lion', 'Cow' and 'Grapes'. Top 3 most relevant attributes and their corresponding AM+PI outputs are shown.

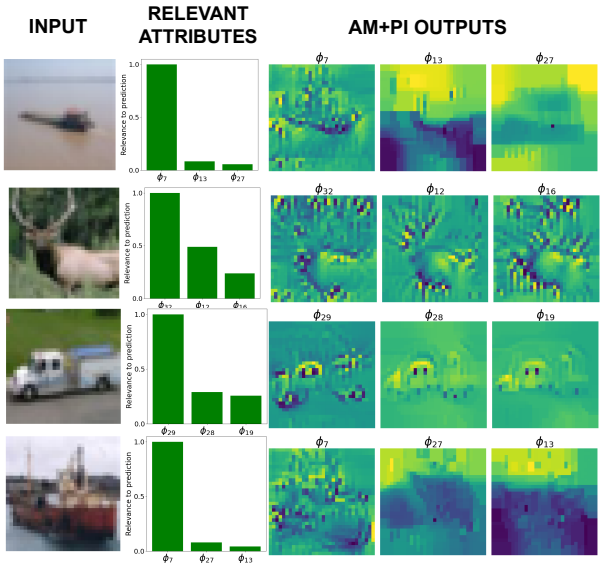


Figure G.11: Local interpretations on test samples for CIFAR-10. True labels are: 'Ship', 'Deer', 'Truck' and 'Ship'. Top 3 most relevant attributes and their corresponding AM+PI outputs are shown.

Thus to avoid any unfair comparison, we did not report this result. The results of BASE-f and SENN on CIFAR, QuickDraw help validate performance of FLINT-f on

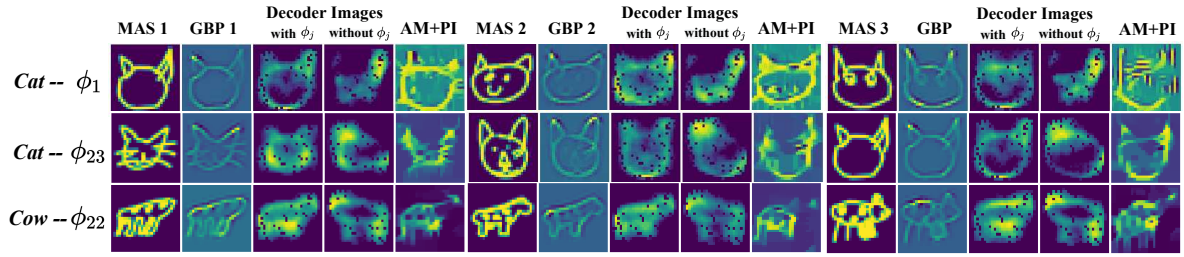


Figure G.12: Examples of class-attribute pairs on QuickDraw, where decoder assists in understanding of encoded concept for the attribute.

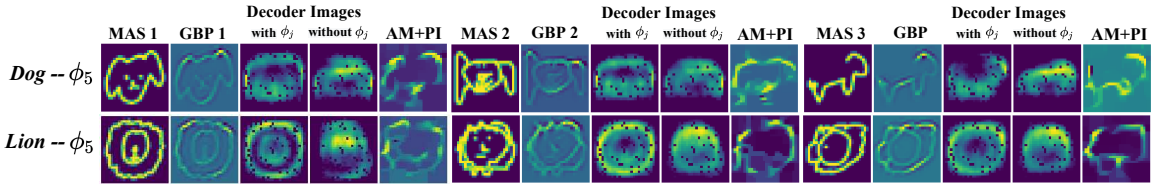


Figure G.13: Examples of class-attribute pairs on QuickDraw, where input attribution (GBP) assists in understanding of encoded concept for the attribute. GBP stands for Guided Backpropagation.

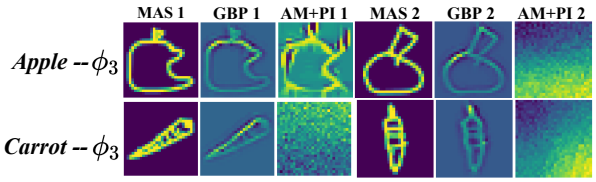


Figure G.14: Sample attribute ϕ_3 learnt without $\mathcal{L}_{if}$. GBP stands for Guided Backpropagation.

QuickDraw.

SENN We compare the accuracy as well as conciseness curve for FLINT with Self-Explaining Neural Networks (SENN) Alvarez-Melis and Jaakkola (2018a). We implemented it with the help of their official implementation available on GitHub [2]. SENN employs a LeNet styled network for MNIST in their paper. We use the same architecture for MNIST and FashionMNIST. For QuickDraw and CIFAR10 we use the VGG based architecture proposed for SENN in their paper to process more complex images. However, to maintain fairness, the number of attributes used in all the experiments for SENN are same as those for FLINT, that is, 25 for MNIST & FashionMNIST, 24 for QuickDraw and 36 for CIFAR10, and also train for the same number of epochs. We use the default choices in their implementation for all hyperparameters and other settings. Another notable point is that although interpretations of SENN are worse than FLINT in conciseness (even when compared non-entropy version of FLINT), the

[2] https://github.com/dmelis/SENN

strength of ℓ_1 regularization in SENN is 2.56 times our strength (for identical $\mathcal{L}_{pred}$, i.e, cross-entropy loss with weight 1.0).

VIBI & LIME We benchmark the fidelity of interpretations of FLINT-g for both by-design and post-hoc interpretation applications against a state-of-the-art black box explainer variational information bottleneck for interpretation (VIBI) Bang et al. (2021) and traditional explainer LIME Ribeiro et al. (2016). Note that VIBI also possesses a model approximating the predictor for all samples. Both methods are implemented using the official repository for VIBI [3]. We compute the "*Approximator Fidelity*" metric as described in their paper, for both systems. In the case of VIBI, this metric exactly coincides with our definition of fidelity. We set the hyperparameters to the setting that yielded best fidelity for datasets reported in their paper. For VIBI, chunk size 4×4, number of chunks $k = 20$, for LIME, chunk size 2×2, number of chunks $k = 40$. The other hyperparameters were the default parameters in their code.

G.1.6 Subjective evaluation details

The form taken by the participants can be accessed here [4]. 17 of the 20 respondents were in the age range 24-31 and at least 16 had completed a minimum of masters level of education in fields strongly related to computer science, electrical engineering or statistics. The form consists of a description where the participants are briefly explained through an example the various information (class-attribute pair visualizations and textual description) they are shown and the response they are supposed to report for each attribute, which is the level of agreement/disagreement with the statement: “The patterns depicted in AM + PI outputs can be meaningfully associated to the textual description". As mentioned in the chapter, four descriptions (questions #2, #5, #8, #9 in the form) were manually corrupted to better ensure that participants are informed about their responses. The corruption mainly consisted of referring to other parts or concepts regarding the relevant class which are *not* emphasized in the AM+PI outputs.

G.2 Post-hoc interpretation: Further analysis

G.2.1 Additional visualizations

Figs. G.15 and G.16 contain global relevances for post-hoc interpretations on all four datasets. Figs. G.17 and G.18 illustrate some additional visualizations of class-attribute pairs.

[3] https://github.com/SeojinBang/VIBI
[4] https://forms.gle/PW6DEPZSmXb46Lnv9

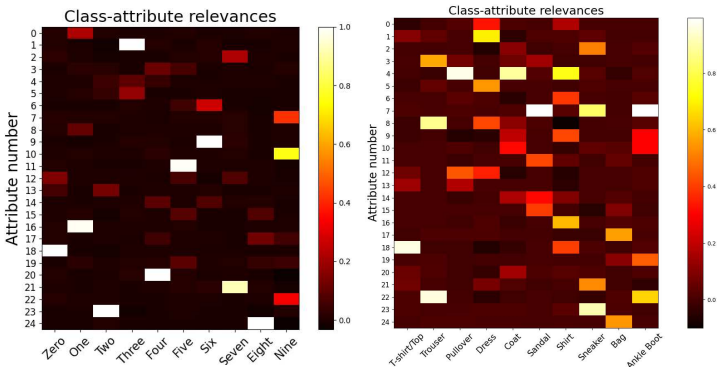


Figure G.15: Global class-attribute relevances $r_{k,c}$ for post-hoc interpretations on MNIST (Left) and FashionMNIST (Right). 15 class-attribute pairs for MNIST and 28 pairs for FashionMNIST have relevance $r_{k,c} > 0.2$.

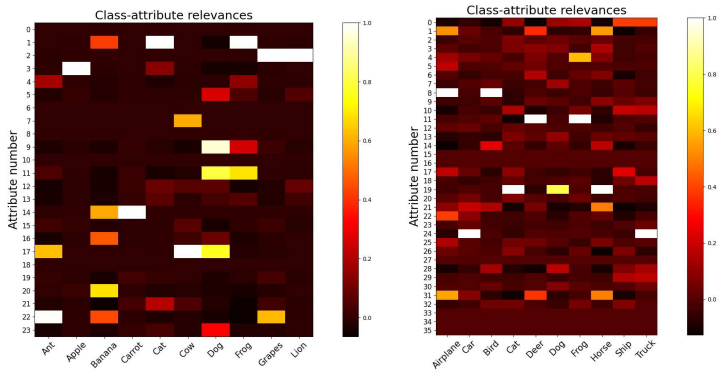


Figure G.16: Global class-attribute relevances $r_{k,c}$ for post-hoc interpretations on QuickDraw (Left) and CIFAR10 (Right). 24 class-attribute pairs for QuickDraw and 26 pairs for CIFAR10 have relevance $r_{k,c} > 0.2$.

G.2.2 Experiments using ACE

We conducted additional experiments using ACE to interpret trained models from our experiments. The key bottleneck for ACE's application on our datasets and networks is the use of CNN as a similarity metric (to automate human annotation) for image segments irrespective of their scale, aspect ratio. This is a specialized property only been empirically shown for specific CNN's trained on ImageNet (as discussed in their paper). The networks trained on our datasets thus very often cluster unrelated segments, resulting in little to no consistency in any extracted concept. To illustrate the above we describe the experimental settings and show extracted concepts for a few classes from QuickDraw and CIFAR-10 on the BASE-f models. The quality of results is the same when interpreting FLINT-f models although we only illustrate interpretations from BASE-f models.

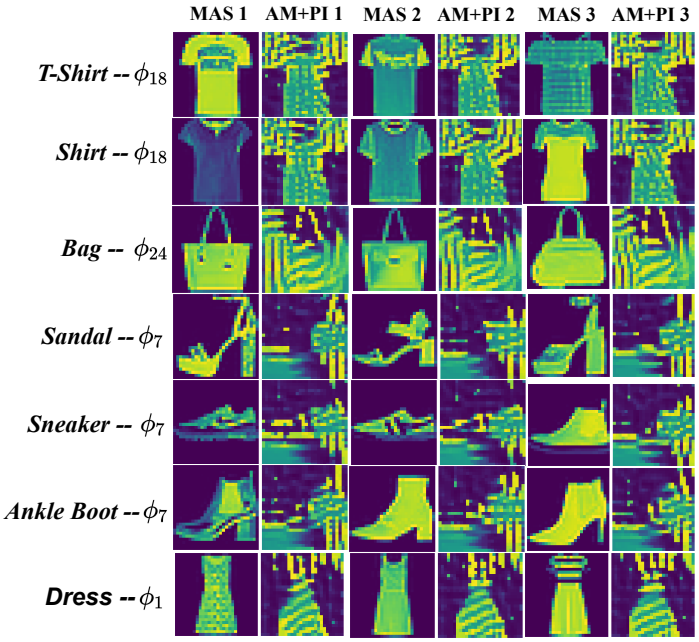


Figure G.17: Sample class-attribute visualizations for post-hoc interpretations for Fashion-MNIST

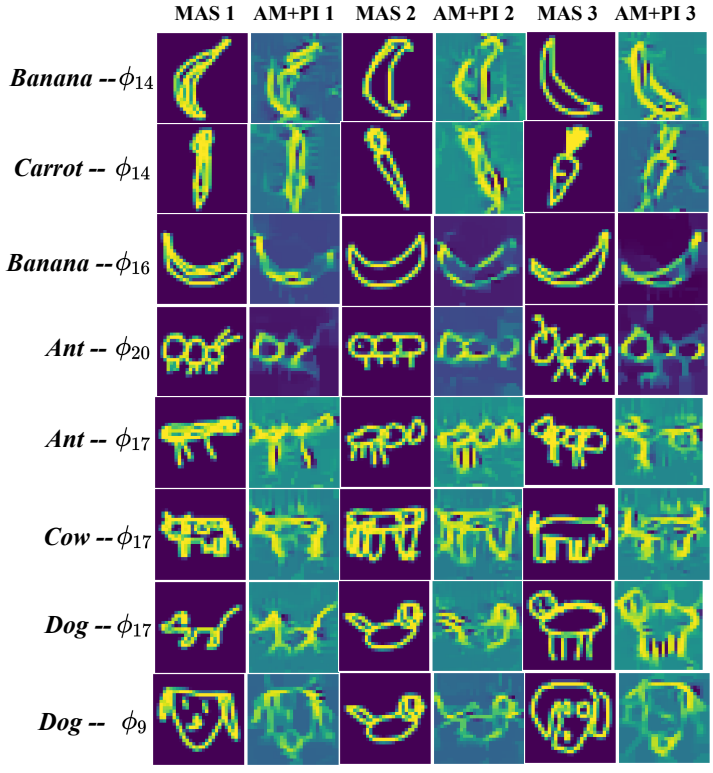


Figure G.18: Sample class-attribute visualizations for post-hoc interpretations on QuickDraw

Experimental setting. We utilize the official open-sourced implementation of their method [5]. Due to the smaller sized images we perform segmentation at a single scale. We experimented with different configurations for “number of segments" and “number of clusters/concepts". The number of segments were varied from 3 to 15. For higher values the segments were often too small for concepts to be meaningful. We thus kept the number of segments 5 for each sample. For each class we chose 100 samples. The number of clusters were varied from 5 to 25. Due to the smaller number of segments (compared to original experiments from ACE which used 25), we kept number of clusters at 12. We access the deepest intermediate layer used in experiments with FLINT (shown in Fig. G.2).

Results. The top 3 discovered concepts (according to the TCAV scores) are shown in Fig. G.19. The segments for any concept on CIFAR show almost no consistency. This is mainly because the second step pf ACE, requiring a CNN's intermediate representations to replace a human subject for measuring the similarity of superpixels/segments, is hard to expect for these networks not trained on ImageNet. Thus, segments capturing background or any random part of the object, completely unrelated, end up clustered together. For QuickDraw, the segmentation algorithm also suffers problems in extracting meaningful segments due to sparse grayscale images. It generally extracts empty spaces or a big chunk of the object itself. This, compounded with the earlier issue about segment similarity results in mostly meaningless concepts. The only slight exception to this is concept 3 for 'Ant' for which two segments capture a single flat blob with small tentacles.

[5] https://github.com/amiratag/ACE

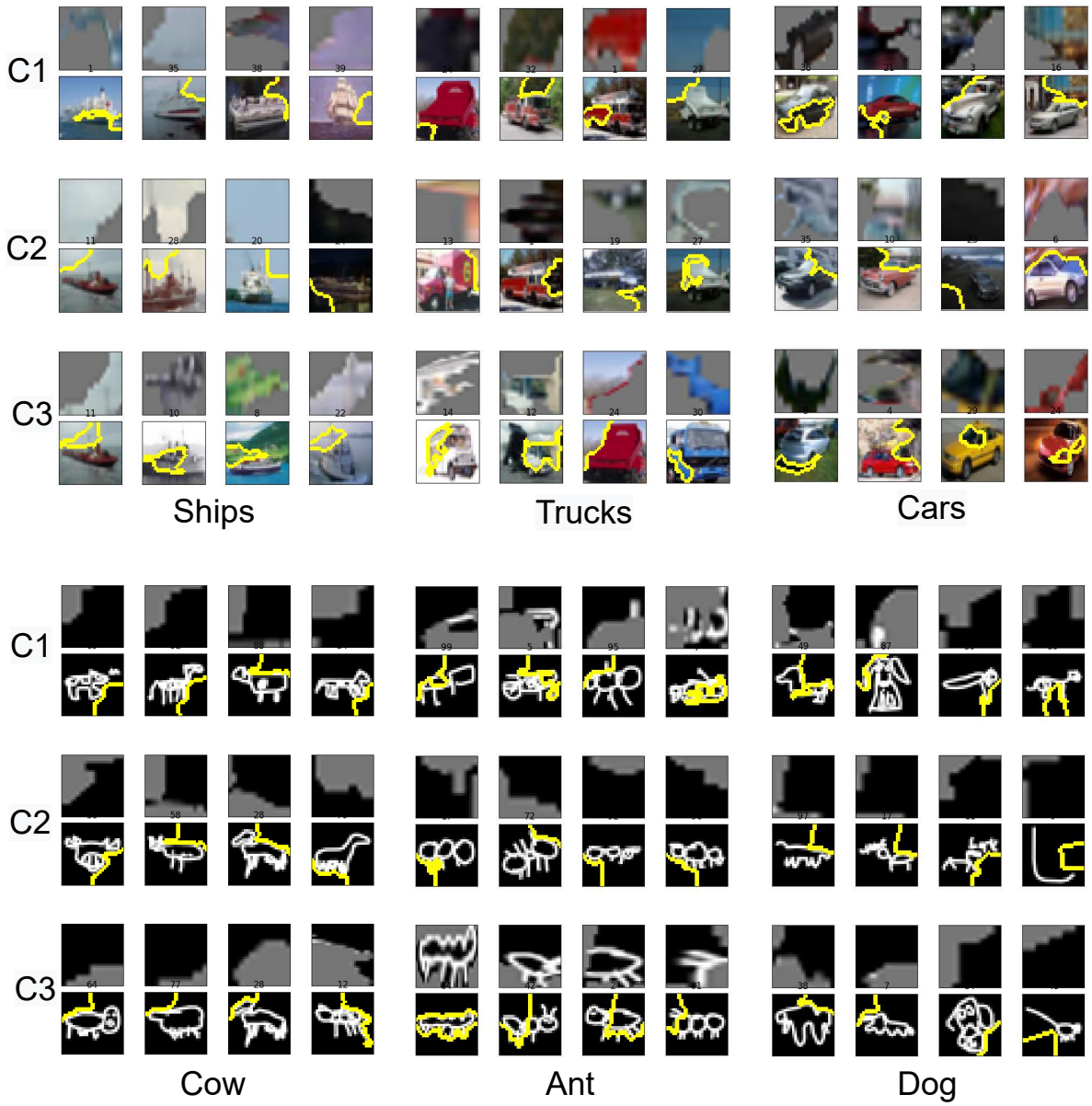


Figure G.19: Discovered concepts using ACE for 3 classes on CIFAR-10 (Top) and QuickDraw (Bottom). We show the top 3 concepts according to their TCAV scores. Each concept consists of 4 segments extracted from images of the class. They are shown in 2 rows, the first contains the segments and the second shows where the segment was extracted from.

H

Appendix for Chapter 5

Contents

H.1 Further discussion on L2I interpretations

H.1.1 Corruption samples ESC-50

The goal of this experiment is to qualitatively illustrate that our method can generate interpretations on ESC-50 in various noisy situations. For this, we corrupt a given sample from a target class in two ways: (i) With sample from a different class (Overlap experiment), and (ii) Adding high amount of white noise, at 0dB SNR (Noise experiment). The key question that we want the interpretations to offer insight on is: *did the classifier truly make its decision because it "heard" the target class or is it making the decision based on the corruption part of the audio?* The cases where classifier misclassifies are analyzed in Sec. H.1.2. As already highlighted in Sec. 1, listenable interpretations are not expected to perform source separation for the class of interest, but to confirm if decision corresponds entirely/mostly to target class or not. All examples can be listened to on our companion website [1]. Since the target and corrupting signals and their classes are already known, we can reinforce the observations drawn by listening to the interpretations through spectrograms (Figs. H.1, H.2).

[1] https://jayneelparekh.github.io/listen2interpretV2/

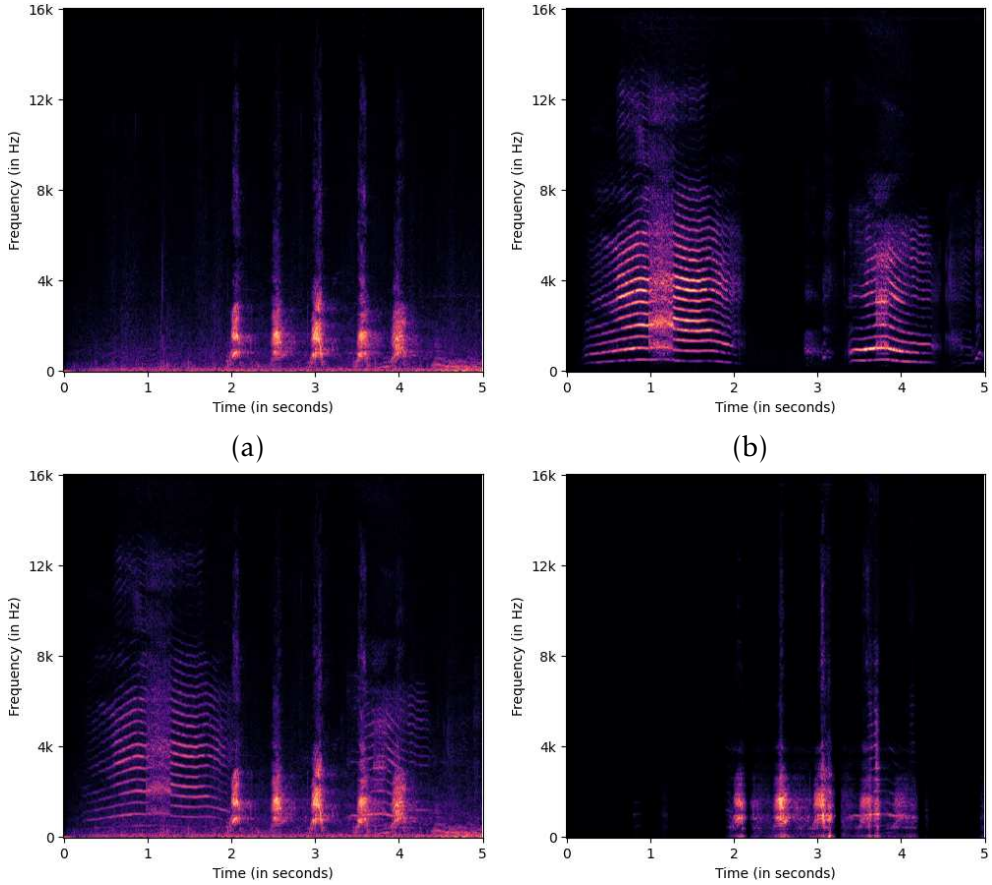
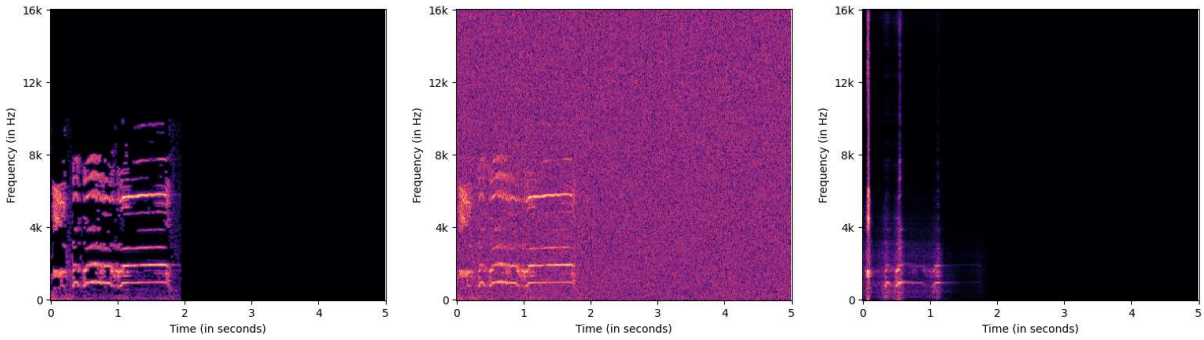


Figure H.1: Log-magnitude spectrograms of an example from Overlap experiment: (a) Target class ('Dog') original uncorrupted signal (b) Corrupting/Mixing class ('Crying-Baby') signal (c) Corrupted/mixed signal, also the input audio to the classifier (d) Interpretation audio for the predicted class ('Dog'). The interesting observation is that spectrogram of interpretation audio almost entirely consists of parts from target class ('Dog') signal with only a very weak presence of corrupting class ('Crying-Baby') close to the end.



(a) (b) (c)

Figure H.2: Log-magnitude spectrograms of an example from ESC-50 Noise experiment: (a) Target class ('Rooster') original uncorrupted signal, (b) White noise corrupted signal, also the input audio to the classifier (c) Interpretation audio for the predicted class ('Rooster'). Again, the interpretation audio is almost entirely free of corrupting signal (white noise in this case) and mostly consists of parts of the original target signal. This strongly indicates that the classifier relied on parts of audio corresponding to the target class to make its decision, and not the white noise.

H.1.2 Misclassification samples ESC-50

When the classifier prediction is incorrect, the interpretations may still provide insight into the classifier's decision by indicating what the classifier "heard" in the input signal. We give examples for this on the webpage[1]. For instance, one of the example is of a sample with ground-truth class 'Crying-Baby' misclassified as a 'Car-horn'. Interestingly, the interpretation is acoustically similar to car horns. Please note the importance of *listenable* interpretations that aid such understanding into the audio network's decisions.

H.2 Discussion on interpretations from other methods

H.2.1 Attribution maps for listenable output

Input attribution/saliency maps in their current form are more suitable for images. These maps are generally spatially smooth, which aids visual understandability, but are not effective masks to clearly emphasize time-frequency bins. Thus, for audio spectrogram like inputs, while they can be useful in visually indicating the important regions, they are poor masks to filter such information for listenable output. We experimented with two approaches to generate attribution maps for few samples on ESC50-Noise Experiment. One is based on information bottleneck attribution (IBA) (Schulz et al., 2019), and other is a gradient backpropagation variant, GuidedBackProp (Springenberg et al., 2014).

Experimental details: For IBA, we used the python PyTorch version of their package and follow the standard example version given in their repository [2]. The example inserts a bottleneck in conv layer from 4th block of VGG16. Our network architecture is also similar to VGG architectures. So we applied a bottleneck at the output of 4th conv block (B4), which we also access via our interpreter. We also follow the same optimization procedure as in the example, i.e. Adam for 10 iterations. The saliency map is applied as a filter on the mel-spectrogram. We then approximate STFT from mel-spectrogram and invert it using input phase for a time-domain audio output. For GuidedBackProp, we simply used a standard implementation [3]. It doesn't require setting any hyperparameters. The saliency map is normalized to a range of $[-1, 1]$ and then applied as a filter on mel-spectrogram and converted to audio domain as for IBA.

Outputs can be heard on our companion website [1]. We provide visualizations for a sample in Fig. H.3 for IBA and Fig. H.4 for GuidedBackProp. For IBA, while the saliency map indeed visually indicates relevant regions, the time-domain signal still contains considerable noise and is not very useful. The smoothness of saliency maps in this case can be partly attributed to upsampling of information extracted from lower resolution feature maps, similar to GradCAM (Selvaraju et al., 2017). On the other hand the saliency map for GuidedBackProp is extremely sharp and jagged as a filter. It still tends to indicate visually the correct regions but results in high number of artifacts and high relative noise still in interpretation audio.

Another limitation of applying these methods to 2D CNN's is the frequent use of log-mel spectrogram as input (current model uses 128 mel bands) for the networks. The saliency map is then over the mel-spectrogram space. There is no trivial way of applying saliency maps as a filter. Moreover, relying on mel-spectrograms for filtering adds to the loss of information when converting to time-domain. Despite their visual usefulness, we believe these methods require non-trivial updates to be suitable for generating listenable interpretations.

H.2.2 Interpretations of FLINT

For completeness, we also provide examples of interpretations by FLINT on ESC-50 Noise samples. As discussed in Sec. 2, FLINT uses a visualization pipeline to understand high-level attributes, which primarily consists of using activation maximization (Mahendran and Vedaldi, 2016) based procedure to emphasize patterns relevant for the activation of an attribute.

In our current setting, this optimization procedure takes place in the log-mel spectrogram space. For initialization with a "weak version" version of the input we subtract 10 from the input log-mel spectrogram. We use Adam optimizer for 1500 iterations We add below examples of this visualization strategy after estimating log-magnitude spectrogram from the output of optimization procedure. Additionally we also estim-

[2] https://github.com/BioroboticsLab/IBA
[3] https://github.com/utkuozbulak/pytorch-cnn-visualizations

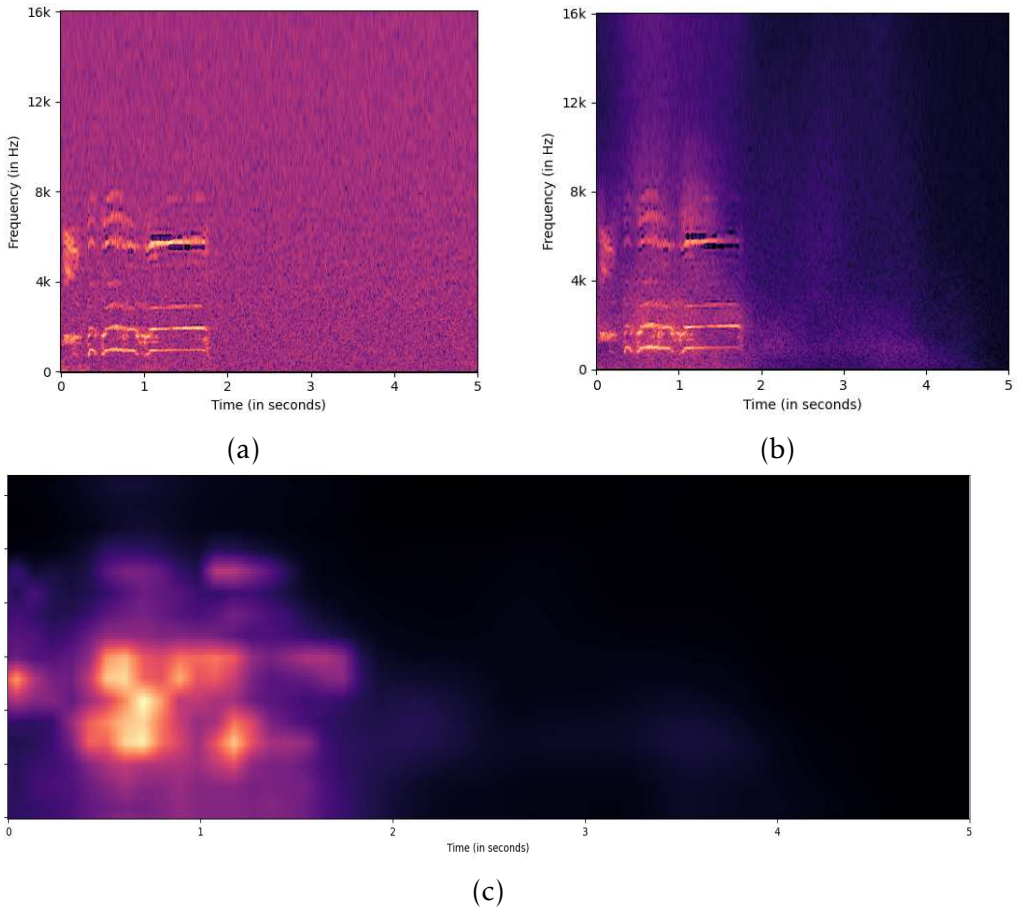
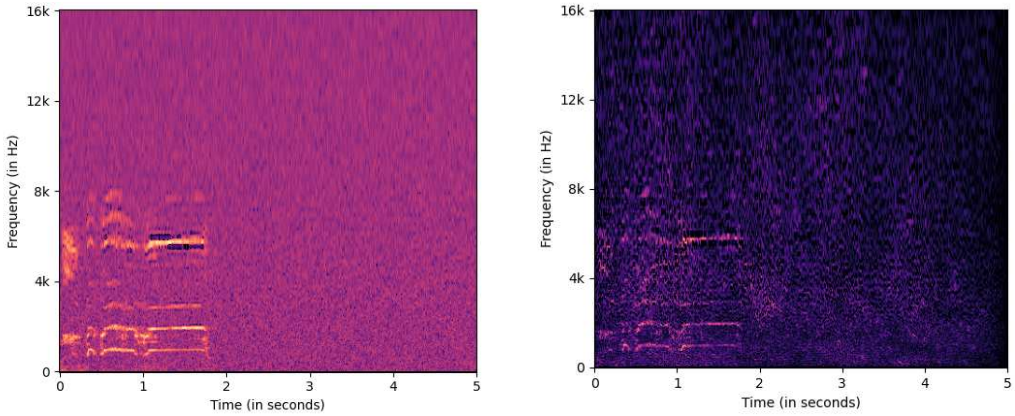


Figure H.3: Log-magnitude spectrograms and IBA saliency map to visualize an attribution map on ESC50-Noise sample: (a) White noise corrupted signal (from class 'Rooster'), also the input audio to the classifier, (b) Interpretation audio for the predicted class ('Rooster'), (c) Saliency map on the log-mel spectra space. The regions corresponding to the signal frequencies are brightest in the saliency map. However, owing to it's smoothness and loss of information in mel-spectrogram space, high amount of noise is still a part of interpretation signal.

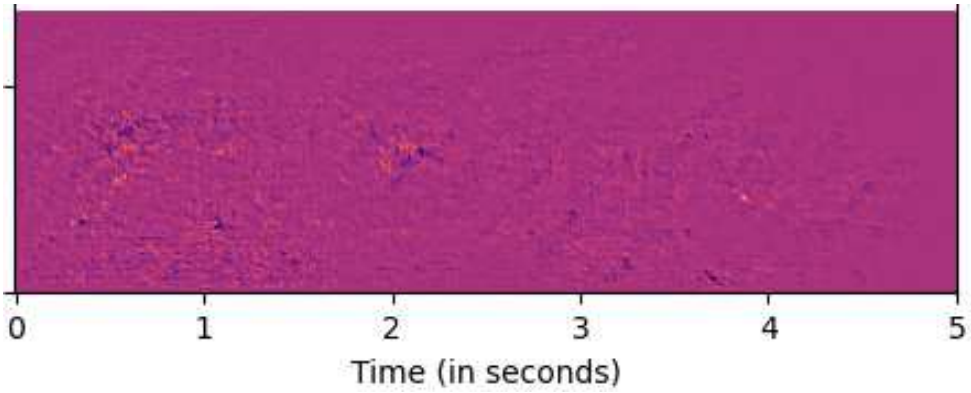
ate the time-domain signal as before to verify any potential as listenable output on the webpage [1].

The optimization in general results in specific patterns added in a log mel-spectrogram and thus the magnitude spectrogram. However, visually understanding the significance of the patterns is a very hard task. Listening to the resulting spectrograms is not informative either as they typically do not remove the noise, nor do they correspond to recognizable phenomenon. Compared to dictionary of pre-learnt spectral patterns, the dictionary of attributes is less constrained in the information an individual attribute encodes. Moreover, FLINT's visualization pipeline provides finer-grained interpretation at an attribute level. Both these considerations require the pipeine to be lot more effective to convey the interpretation understandably for audio modality.



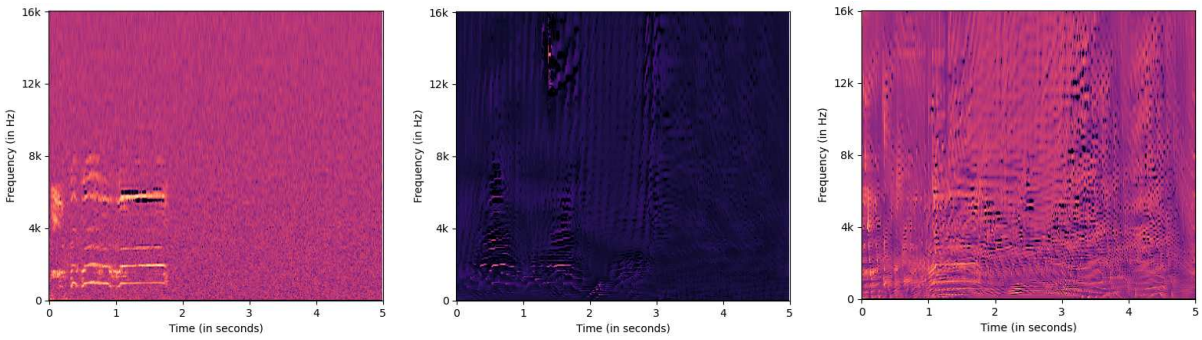
(a)

(b)



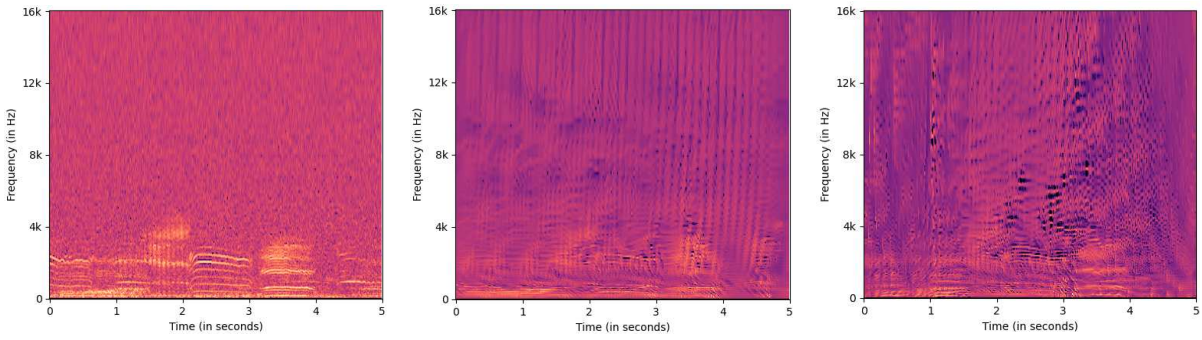
(c)

Figure H.4: Log-magnitude spectrograms and GuidedBackProp saliency map to visualize an attribution map on ESC50-Noise sample: (a) White noise corrupted signal (from class 'Rooster'), also the input audio to the classifier, (b) Interpretation audio for the predicted class ('Rooster'), (c) Saliency map on the log-mel spectra space. It is hard to understand the saliency map directly but the absolute relevances tend to be higher in region corresponding to signal. However, it is still not a suitable filter for listenable audio and results in many artifacts, high presence of noise still (relative to weakened signal strength).



(a) (b) (c)

Figure H.5: Log-magnitude spectrogram visualizations for two relevant attributes of FLINT on a sample from ESC50-Noise experiment: (a) White noise corrupted input audio (class: 'Rooster'), (b) Activation maximization output for attribute 62, (c) Activation maximization output for attribute 77.



(a) (b) (c)

Figure H.6: Log-magnitude spectrogram visualizations for two relevant attributes of FLINT on a sample from ESC50-Noise experiment: (a) White noise corrupted input audio (class: 'Sheep'), (b) Activation maximization output for attribute 7, (c) Activation maximization output for attribute 77.

H.2.3 Baseline implementations details

FLINT: We implemented it with the help of their official implementation available on GitHub.[4] For each experiment, we fix their number of attributes J equal to the number of our NMF components K. We also choose the same hidden layers for their system as we choose for ours. This baseline is trained for the same number of epochs as us. We use same values for our $\mathcal{L}_{\text{NMF}}$ loss weight, α, and their $\mathcal{L}_{if}$ loss weight γ. For the other loss hyperparameters, we use their default values and training strategy.

[4] https://github.com/jayneelparekh/FLINT

VIBI: We implemented this using their official repository.[5] The key hyperparameters that we set are the input chunk size and their parameter K, the number of chunks to use for interpretation. We use a larger chunk size than in their experiments to limit the number of chunks. On ESC-50, we use a chunk size of 32×43, and on SONYC-UST, a chunk size of 32×86. This yields 40 chunks for each input on both the datasets. We varied the K from 5 to 20, and report the results with best fidelity. The system was trained for 100 epochs on ESC-50 and 30 epochs on SONYC-UST.

SLIME: We primarily relied on implementation from their robustness analysis repository [6]. The key hyperparameters to balance are the number of chunks vs chunk size. SONYC-UST contains 10 second audio files. This is much longer than 1.6 second audio files for which SLIME was originally demonstrated (Mishra et al., 2017). Therefore, we divide only on the time-axis to limit the number of chunks. SLIME recommends a chunk size of at least 100ms. They operate on upto 290ms chunk size. We balance these two hyperparameters by dividing our audio files in 20 chunks of 500ms chunk size. We select a maximum of 5 chunks for interpretations and a neighbourhood size of 1000.

APNet: We utilized their source code [7] for implementing their method on our datasets. We did not modify their network design or loss weights and set the number of prototypes same as our number of components. The number of mel filters was chosen between 64 and 128. We trained their system for 100 epochs on ESC-50 (each fold) and 21 epochs for SONYC-UST and OpenMIC-2018 and report the highest recorded metrics.

NMF variants: For implementing both TDL-NMF and Unsupervised-NMF, we utilized the source repository [8] of (Bisot et al., 2017). The unsupervised-NMF variant simply trains a linear model on top of generated time activations for predictions while the dictionary is also updated with classification loss for TDL-NMF.. We trained dictionaries of multiple sizes, ranging from 32 to 256 for each dataset and two different audio representations, log-magnitude spectrogram and mel-spectrogram. The best performance among all these configurations is reported.

H.2.4 Subjective evaluation implementation

The subjective evaluation interface was implemented using webMUSHRA (Schoeffler et al., 2018). Prior to voting on the test samples, participants were provided with an instruction page and then a training page with an example to get used to interface, instructions, tune their volume etc. Screenshots of the instruction and training page are given in Fig. H.7, Fig. H.8 respectively.

[5] https://github.com/SeojinBang/VIBI
[6] https://github.com/saum25/local_exp_robustness
[7] https://github.com/pzinemanas/APNet
[8] https://github.com/rserizel/TGNMF

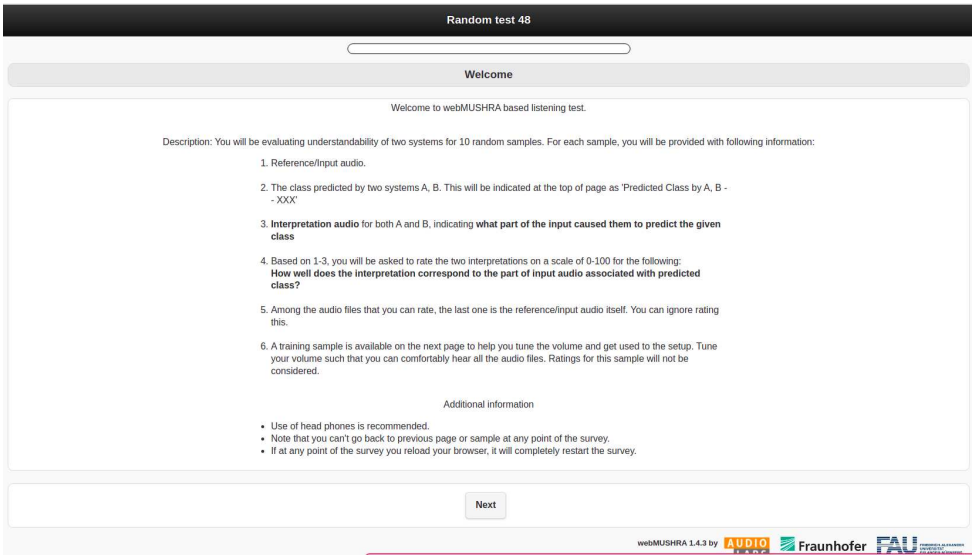


Figure H.7: Instructions for the participants at the start of the subjective evaluation

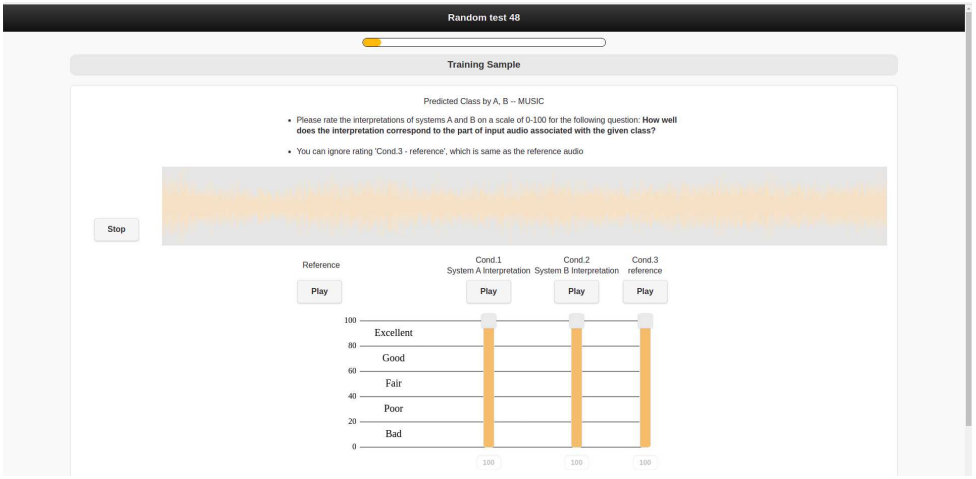


Figure H.8: Training page for subjective evaluation that illustrates the interface for scoring for the participants.

Bibliography

A. Abujabal, R. S. Roy, M. Yahya, and G. Weikum. Quint: Interpretable question answering over knowledge bases. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 61–66, 2017. page 30

R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE transactions on pattern analysis and machine intelligence*, 34(11):2274–2282, 2012. page 25

C. Agarwal, N. Johnson, M. Pawelczyk, S. Krishna, E. Saxena, M. Zitnik, and H. Lakkaraju. Rethinking stability for attribution-based explanations. *arXiv preprint arXiv:2203.06877*, 2022. page 36

R. Agarwal, N. Frosst, X. Zhang, R. Caruana, and G. Hinton. Neural additive models: Interpretable machine learning with neural nets. *arXiv preprint arXiv:2004.13912*, 2020. pages 20, 45, 109

M. Al-Shedivat, A. Dubey, and E. Xing. Contextual explanation networks. *arXiv preprint arXiv:1705.10301*, 2017. pages 17, 20, 33

D. Alvarez-Melis and T. Jaakkola. Towards robust interpretability with self-explaining neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 7775–7784, 2018a. pages 17, 19, 20, 22, 29, 33, 36, 44, 47, 58, 62, 63, 93, 133

D. Alvarez-Melis and T. S. Jaakkola. A causal framework for explaining the predictions of black-box sequence-to-sequence models. *arXiv preprint arXiv:1707.01943*, 2017. page 20

D. Alvarez-Melis and T. S. Jaakkola. On the robustness of interpretability methods. *arXiv preprint arXiv:1806.08049*, 2018b. pages 17, 36

P. Angelov and E. Soares. Towards explainable deep neural networks (xdnn). *Neural Networks*, 130:185–194, 2020. pages 26, 31

M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein generative adversarial networks. In *International conference on machine learning*, pages 214–223. PMLR, 2017. page 113

A. Arnault and N. Riche. CRNNs for urban sound tagging with spatiotemporal context. Technical report, DCASE2020 Challenge, October 2020. pages 90, 91

I. Arous, L. Dolamic, J. Yang, A. Bhardwaj, G. Cuccu, and P. Cudré-Mauroux. Marta: Leveraging human rationales for explainable text classification. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 5868–5876, 2021. page 21

V. Arya, R. K. Bellamy, P.-Y. Chen, A. Dhurandhar, M. Hind, S. C. Hoffman, S. Houde, Q. V. Liao, R. Luss, A. Mojsilović, et al. One explanation does not fit all: A toolkit and taxonomy of ai explainability techniques. *arXiv preprint arXiv:1909.03012*, 2019. page 36

M. Asteris, D. Papailiopoulos, and A. G. Dimakis. Orthogonal nmf through subspace exploration. *Advances in neural information processing systems*, 28, 2015. page 108

S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7):e0130140, 2015. pages 16, 20, 23, 36

R. Badeau and T. Virtanen. Nonnegative matrix factorization. *Audio Source Separation and Speech Enhancement*, pages 131–160, 2018. page 82

D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014. page 20

S. Bang, P. Xie, H. Lee, W. Wu, and E. Xing. Explaining a black-box by using a deep variational information bottleneck approach. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 11396–11404, 2021. pages 20, 25, 33, 53, 64, 65, 66, 93, 96, 134

D. Bank, N. Koenigstein, and R. Giryes. Autoencoders. *arXiv preprint arXiv:2003.05991*, 2020. page 48

W. Bank. *World development report 2019: The changing nature of work*. The World Bank, 2018. page 7

S. Becker, M. Ackermann, S. Lapuschkin, K.-R. Müller, and W. Samek. Interpreting and explaining deep neural networks for classification of audio signals. *arXiv preprint arXiv:1807.03418*, 2018. page 34

C. Bénard, G. Biau, S. Da Veiga, and E. Scornet. Sirus: Stable and interpretable rule set for classification. *Electronic Journal of Statistics*, 15(1):427–505, 2021. page 21

Y. Bengio et al. Learning deep architectures for ai. *Foundations and trends® in Machine Learning*, 2(1):1–127, 2009. page 44

N. Bertin, R. Badeau, and G. Richard. Blind signal decompositions for automatic transcription of polyphonic music: Nmf and k-svd on the benchmark. In *2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07*, volume 1, pages I–65. IEEE, 2007. page 83

A. Bertrand, R. Belloum, J. R. Eagan, and W. Maxwell. How cognitive biases affect xai-assisted decision-making: A systematic review. In *Proceedings of the 2022 AAAI/ACM conference on AI, ethics, and society*, pages 78–91, 2022. page 111

U. Bhatt, A. Weller, and J. M. Moura. Evaluating and aggregating feature-based model explanations. *arXiv preprint arXiv:2005.00631*, 2020a. page 36

U. Bhatt, A. Xiang, S. Sharma, A. Weller, A. Taly, Y. Jia, J. Ghosh, R. Puri, J. M. Moura, and P. Eckersley. Explainable machine learning in deployment. In *Proceedings of the 2020 conference on fairness, accountability, and transparency*, pages 648–657, 2020b. page 8

J. Bien and R. Tibshirani. Prototype selection for interpretable classification. *Annals of Applied Statistics*, 2011. page 26

C. M. Bishop and N. M. Nasrabadi. *Pattern recognition and machine learning*. Springer, 2006. page 7

C. M. Bishop et al. *Neural networks for pattern recognition*. Oxford university press, 1995. page 7

V. Bisot, R. Serizel, S. Essid, and G. Richard. Feature learning with matrix factorization applied to acoustic scene classification. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 25(6):1216–1229, 2017. pages 83, 94, 102, 103, 110, 146

V. Bourgeais, F. Zehraoui, M. Ben Hamdoune, and B. Hanczar. Deep gonet: self-explainable deep neural network based on gene ontology for phenotype prediction from gene expression data. *BMC bioinformatics*, 22(10):1–25, 2021. page 21

O. Boz. Extracting decision trees from trained neural networks. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 456–461, 2002. page 21

J. L. Brand. The misdirected approach of open source algorithms. *AI & society*, pages 1–2, 2022. page 111

A. S. Bregman. *Auditory scene analysis: The perceptual organization of sound*. MIT press, 1994. page 81

M. Cartwright, A. E. M. Mendez, J. Cramer, V. Lostanlen, G. Dove, H.-H. Wu, J. Salamon, O. Nov, and J. Bello. SONYC urban sound tagging (SONYC-UST): A multilabel dataset from an urban acoustic sensor network. In *Proceedings of the Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE)*, pages 35–39, October 2019. page 89

M. Cartwright, J. Cramer, A. E. M. Mendez, Y. Wang, H.-H. Wu, V. Lostanlen, M. Fuentes, G. Dove, C. Mydlarz, J. Salamon, et al. Sonyc-ust-v2: An urban sound tagging dataset with spatiotemporal context. *arXiv preprint arXiv:2009.05188*, 2020. page 91

P. Chalasani, J. Chen, A. R. Chowdhury, X. Wu, and S. Jha. Concise explanations of neural networks using adversarial training. In *International Conference on Machine Learning*, pages 1383–1391. PMLR, 2020. page 36

R. Chatila, V. Dignum, M. Fisher, F. Giannotti, K. Morik, S. Russell, and K. Yeung. Trustworthy ai. *Reflections on Artificial Intelligence for Humanity*, pages 13–39, 2021. page 111

C. Chen, O. Li, D. Tao, A. Barnett, C. Rudin, and J. K. Su. This looks like that: deep learning for interpretable image recognition. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 8928–8939, 2019. pages 26, 33

J. Chen, L. Song, M. Wainwright, and M. Jordan. Learning to explain: An information-theoretic perspective on model interpretation. In *International Conference on Machine Learning*, pages 883–892. PMLR, 2018. pages 21, 25, 33

Z. Chen, Y. Bei, and C. Rudin. Concept whitening for interpretable image recognition. *Nature Machine Intelligence*, 2(12):772–782, 2020. page 30

H. Cho, E. K. Lee, and I. S. Choi. Layer-wise relevance propagation of interaction-net explains protein–ligand interactions at the atom level. *Scientific reports*, 10(1): 21155, 2020. page 20

S. Chowdhury, V. Praher, and G. Widmer. Tracing back music emotion predictions to sound sources and intuitive perceptual qualities. *arXiv preprint arXiv:2106.07787*, 2021. page 34

W. J. Clancey. The epistemology of a rule-based expert system—a framework for explanation. *Artificial intelligence*, 20(3):215–251, 1983. page 15

D. Croce, D. Rossini, and R. Basili. Auditing deep learning processes through kernel-based explanatory models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4037–4046, 2019. page 20

L. De Lara, A. González-Sanz, N. Asher, and J.-M. Loubes. Transport-based counterfactual models. *arXiv preprint arXiv:2108.13025*, 2021. page 32

C. Dittmar and D. Gärtner. Real-time transcription and separation of drum recordings based on nmf decomposition. In *DAFx*, pages 187–194, 2014. page 83

S. Dong, P. Wang, and K. Abbas. A survey on deep learning and its applications. *Computer Science Review*, 40:100379, 2021. page 8

F. Doshi-Velez and B. Kim. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*, 2017. pages 9, 34, 35, 53, 64

B. J. Dyson and C. Alain. Representation of concurrent acoustic objects in primary auditory cortex. *The Journal of the Acoustical Society of America*, 115(1):280–288, 2004. page 81

R. C. Fong and A. Vedaldi. Interpretable explanations of black boxes by meaningful perturbation. In *Proceedings of the IEEE international conference on computer vision*, pages 3429–3437, 2017. page 33

F. Foscarin, K. Hoedt, V. Praher, A. Flexer, and G. Widmer. Concept-based techniques for" musicologist-friendly" explanations in a deep music classifier. *ISMIR*, 2022. pages 34, 110

R. Gao, R. Feris, and K. Grauman. Learning to separate object sounds by watching unlabeled video. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 35–53, 2018. page 83

A. Garcia, C. Clavel, S. Essid, and F. d'Alché Buc. Structured output learning with abstention: Application to accurate opinion prediction. In *International Conference on Machine Learning*, pages 1695–1703. PMLR, 2018. page 43

S. Gautam, A. Boubekki, S. Hansen, S. A. Salahuddin, R. Jenssen, M. Höhne, and M. Kampffmeyer. Protovae: A trustworthy self-explainable prototypical variational model. *arXiv preprint arXiv:2210.08151*, 2022. pages 27, 33

J. F. Gemmeke, D. P. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter. Audio set: An ontology and human-labeled dataset for audio events. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 776–780. IEEE, 2017. page 90

A. Ghorbani, J. Wexler, J. Y. Zou, and B. Kim. Towards automatic concept-based explanations. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 9277–9286, 2019. pages 19, 22, 28, 32, 33, 58, 107

J. Goldstein. Auditory nonlinearity. *The Journal of the Acoustical Society of America*, 41 (3):676–699, 1967. page 85

I. Goodfellow, Y. Bengio, and A. Courville. *Deep learning*. MIT press, 2016. page 7

I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020. pages 109, 112

J. Gou, B. Yu, S. J. Maybank, and D. Tao. Knowledge distillation: A survey. *International Journal of Computer Vision*, 129:1789–1819, 2021. page 43

T. D. Griffiths and J. D. Warren. What is an auditory object? *Nature Reviews Neuroscience*, 5(11):887–892, 2004. page 81

T. Grigoryev, A. Voynov, and A. Babenko. When, why, and which pretrained gans are useful? In *International Conference on Learning Representations*, 2022. page 114

R. Guidotti, A. Monreale, F. Giannotti, D. Pedreschi, S. Ruggieri, and F. Turini. Factual and counterfactual explanations for black box decision making. *IEEE Intelligent Systems*, 2019. page 32

D. Ha and D. Eck. A neural representation of sketch drawings. In *International Conference on Learning Representations*, 2018. page 64

B. Hanczar, F. Zehraoui, T. Issa, and M. Arles. Biological interpretation of deep neural network for phenotype prediction based on gene expression. *BMC bioinformatics*, 21:1–18, 2020. page 20

V. Haunschmid, E. Manilow, and G. Widmer. audiolime: Listenable explanations using source separation. *arXiv preprint arXiv:2008.00582*, 2020. pages 34, 81, 93

K. He, X. Zhang, S. Ren, and J. Sun. Identity mappings in deep residual networks. In *European conference on computer vision*, pages 630–645. Springer, 2016. pages 64, 73, 123, 125

A. Hedström, L. Weber, D. Krakowczyk, D. Bareeva, F. Motzkus, W. Samek, S. Lapuschkin, and M. M.-C. Höhne. Quantus: An explainable ai toolkit for responsible evaluation of neural network explanations and beyond. *Journal of Machine Learning Research*, 24(34):1–11, 2023. page 36

L. A. Hendricks, Z. Akata, M. Rohrbach, J. Donahue, B. Schiele, and T. Darrell. Generating visual explanations. In *European Conference on Computer Vision*, pages 3–19. Springer, 2016. pages 31, 33

J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020. page 112

A. Hoffmann, C. Fanconi, R. Rade, and J. Kohler. This looks like that... does it? shortcomings of latent space prototype interpretability in deep networks. *arXiv preprint arXiv:2105.02968*, 2021. page 27

D. Howes. The senses: polysensoriality. *A Companion to the Anthropology of the Body and Embodiment*, pages 435–450, 2011. page 8

Q. Huang, M. Yamada, Y. Tian, D. Singh, and Y. Chang. Graphlime: Local interpretable model explanations for graph neural networks. *IEEE Transactions on Knowledge and Data Engineering*, 2022. page 20

S. Huang and T. D. Tran. Sparse signal recovery via generalized entropy functions minimization. *IEEE Transactions on Signal Processing*, 67(5):1322–1337, 2018. page 61

E. Humphrey, S. Durand, and B. McFee. Openmic-2018: An open data-set for multiple instrument recognition. In *ISMIR*, pages 438–444, 2018. pages 90, 91

M. Ilse, J. Tomczak, and M. Welling. Attention-based deep multiple instance learning. In *International conference on machine learning*, pages 2127–2136. PMLR, 2018. page 87

P. Jacob, É. Zablocki, H. Ben-Younes, M. Chen, P. Pérez, and M. Cord. Steex: steering counterfactual explanations with semantics. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XII*, pages 387–403. Springer, 2022. page 32

H. Jain, J. Zepeda, P. Pérez, and R. Gribonval. Subic: A supervised, structured binary code for image search. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 833–842, 2017. page 61

A. Kanehira, K. Takemoto, S. Inayoshi, and T. Harada. Multimodal explanations by predicting counterfactuality in videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8594–8602, 2019. pages 20, 31

T. Karras, M. Aittala, J. Hellsten, S. Laine, J. Lehtinen, and T. Aila. Training generative adversarial networks with limited data. *Advances in neural information processing systems*, 33:12104–12114, 2020a. page 117

T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8110–8119, 2020b. pages 113, 117

T. Karras, M. Aittala, S. Laine, E. Härkönen, J. Hellsten, J. Lehtinen, and T. Aila. Alias-free generative adversarial networks. *Advances in Neural Information Processing Systems*, 34:852–863, 2021. page 113

D. Kazhdan, B. Dimanov, M. Jamnik, P. Liò, and A. Weller. Now you see me (cme): Concept-based model extraction. *arXiv preprint arXiv:2010.13233*, 2020. page 30

J. D. M.-W. C. Kenton and L. K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186, 2019. page 20

B. Kim, M. Wattenberg, J. Gilmer, C. Cai, J. Wexler, F. Viegas, and R. Sayres. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). *arXiv preprint arXiv:1711.11279*, 2017. pages 19, 21, 28, 32, 33, 44

P.-J. Kindermans, S. Hooker, J. Adebayo, M. Alber, K. T. Schütt, S. Dähne, D. Erhan, and B. Kim. The (un) reliability of saliency methods. In *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, pages 267–280. Springer, 2019. pages 17, 37

D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. pages 92, 124

D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. pages 109, 112

D. Kitamura, H. Saruwatari, K. Yagi, K. Shikano, Y. Takahashi, and K. Kondo. Music signal separation based on supervised nonnegative matrix factorization with orthogonality and maximum-divergence penalties. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, 97(5):1113–1118, 2014. page 108

P. W. Koh, T. Nguyen, Y. S. Tang, S. Mussmann, E. Pierson, B. Kim, and P. Liang. Concept bottleneck models. In *International Conference on Machine Learning*, pages 5338–5348. PMLR, 2020. pages 8, 21, 30, 33, 35

M. Kohlbrenner, A. Bauer, S. Nakajima, A. Binder, W. Samek, and S. Lapuschkin. Towards best practice in explaining neural network decisions with lrp. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7. IEEE, 2020. page 37

K. Koutini, H. Eghbal-Zadeh, V. Haunschmid, P. Primus, S. Chowdhury, and G. Widmer. Receptive-field regularized cnns for music classification and tagging. *arXiv preprint arXiv:2007.13503*, 2020. page 91

A. Krizhevsky, G. Hinton, et al. Learning multiple layers of features from tiny images. 2009. pages 64, 73

A. Kumar, M. Khadkevich, and C. Fügen. Knowledge transfer from weakly labeled audio using convolutional neural network for sound events and scenes. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 326–330. IEEE, 2018. page 90

R. Kusters, Y. Kim, M. Collery, C. d. S. Marie, and S. Gupta. Differentiable rule induction with learned relational features. *arXiv preprint arXiv:2201.06515*, 2022. pages 31, 109

I. Lage, A. Ross, S. J. Gershman, B. Kim, and F. Doshi-Velez. Human-in-the-loop interpretability prior. *Advances in neural information processing systems*, 31, 2018. page 48

H. Lakkaraju, E. Kamar, R. Caruana, and J. Leskovec. Faithful and customizable explanations of black box models. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, pages 131–138, 2019. pages 16, 18, 20, 31

H. Lakkaraju, N. Arsov, and O. Bastani. Robust and stable black box explanations. In *International Conference on Machine Learning*, pages 5628–5638. PMLR, 2020. pages 18, 20, 53, 65

H. Lakkaraju, D. Slack, Y. Chen, C. Tan, and S. Singh. Rethinking explainability as a dialogue: A practitioner's perspective. *arXiv preprint arXiv:2202.01875*, 2022. page 30

O. Lang, Y. Gandelsman, M. Yarom, Y. Wald, G. Elidan, A. Hassidim, W. T. Freeman, P. Isola, A. Globerson, M. 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. pages 31, 33, 112, 113

J. Le Roux, H. Kameoka, N. Ono, A. De Cheveigne, and S. Sagayama. Computational auditory induction as a missing-data model-fitting problem with bregman divergence. *Speech Communication*, 53(5):658–676, 2011. page 83

J. Le Roux, J. R. Hershey, and F. Weninger. Deep nmf for speech separation. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 66–70. IEEE, 2015a. page 83

J. Le Roux, F. J. Weninger, and J. R. Hershey. Sparse nmf–half-baked or well done? *Mitsubishi Electric Research Labs (MERL), Cambridge, MA, USA, Tech. Rep., no. TR2015-023*, 11:13–15, 2015b. pages 83, 86, 88

Y. LeCun. Lenet-5, convolutional neural networks. *URL: http://yann. lecun. com/exdb/lenet*, 2015. pages 64, 123, 124

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. page 64

D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. In T. Leen, T. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems*, volume 13. MIT Press, 2001. URL https://proceedings.neurips.cc/paper/2000/file/f9d1152547c0bde01830b7e8bd60024c-Paper.pdf. pages 82, 83

G.-H. Lee, W. Jin, D. Alvarez-Melis, and T. S. Jaakkola. Functional transparency for structured data: a game-theoretic approach. *arXiv preprint arXiv:1902.09737*, 2019. pages 17, 21

O. Li, H. Liu, C. Chen, and C. 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. pages 17, 19, 22, 26, 33, 34, 47, 64, 65, 131

Y. Li, J. Zhou, S. Verma, and F. Chen. A survey of explainable graph neural networks: Taxonomy and evaluation metrics. *arXiv preprint arXiv:2207.12599*, 2022. page 20

C. Lin, G. J. Sun, K. C. Bulusu, J. R. Dry, and M. Hernandez. Graph neural networks including sparse interpretability. *arXiv preprint arXiv:2007.00119*, 2020. page 20

Z. C. Lipton. The mythos of model interpretability. *Commun. ACM*, 61(10):36–43, 2018. pages 9, 10

B. Liu, Y. Zhu, Z. Fu, G. De Melo, and A. Elgammal. Oogan: Disentangling gan with one-hot sampling and orthogonal regularization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 4836–4843, 2020. page 108

M. LJPvd and G. Hinton. Visualizing high-dimensional data using t-sne. *J Mach Learn Res*, 9(2579-2605):9, 2008. page 99

R. Loiseau, B. Bouvier, Y. Teytaut, E. Vincent, M. Aubry, and L. Landrieu. A model you can hear: Audio identification with playable prototypes. *arXiv preprint arXiv:2208.03311*, 2022. pages 26, 27

S. M. Lundberg and S.-I. Lee. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems*, pages 4765–4774, 2017. pages 16, 18, 20, 24, 33, 46

D. Luo, W. Cheng, D. Xu, W. Yu, B. Zong, H. Chen, and X. Zhang. Parameterized explainer for graph neural network. *Advances in neural information processing systems*, 33:19620–19631, 2020. page 20

T. Ma and A. Zhang. Incorporating biological knowledge with factor graph neural network for interpretable deep learning. *arXiv preprint arXiv:1906.00537*, 2019. page 21

A. Mahendran and A. Vedaldi. Visualizing deep convolutional neural networks using natural pre-images. *International Journal of Computer Vision*, 120(3):233–255, 2016. pages 63, 142

A. Margeloiu, M. Ashman, U. Bhatt, Y. Chen, M. Jamnik, and A. Weller. Do concept bottleneck models learn as intended? *arXiv preprint arXiv:2105.04289*, 2021. page 30

S. Mishra, B. L. Sturm, and S. Dixon. Local interpretable model-agnostic explanations for music content analysis. In *ISMIR*, pages 537–543, 2017. pages 16, 25, 34, 146

S. Mishra, E. Benetos, B. L. Sturm, and S. Dixon. Reliable local explanations for machine listening. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2020. pages 34, 81, 94

C. Molnar. *Interpretable machine learning*. Lulu. com, 2020. page 9

G. Montavon, M. L. Braun, T. Krueger, and K.-R. Muller. Analyzing local structure in kernel-based learning: Explanation, complexity, and reliability assessment. *IEEE Signal Processing Magazine*, 30(4):62–74, 2013. page 15

G. Montavon, W. Samek, and K.-R. Müller. Methods for interpreting and understanding deep neural networks. *Digital Signal Processing*, 73:1–15, 2018. pages 18, 36

M. Moradi and M. Samwald. Post-hoc explanation of black-box classifiers using confident itemsets. *Expert Systems with Applications*, 165:113941, 2021. page 31

K. Mosler. Depth statistics. In C. Becker, R. Fried, and S. Kuhnt, editors, *Robustness and Complex Data Structures: Festschrift in Honour of Ursula Gather*, pages 17–34. Springer. Berlin, 2013. page 70

H. Muckenhirn, V. Abrol, M. Magimai-Doss, and S. Marcel. Understanding and visualizing raw waveform-based CNNs. In *Interspeech*, pages 2345–2349, 2019. page 34

J. Mullenbach, S. Wiegreffe, J. Duke, J. Sun, and J. Eisenstein. Explainable prediction of medical codes from clinical text. *arXiv preprint arXiv:1802.05695*, 2018. page 20

A.-p. Nguyen and M. R. Martínez. On quantitative aspects of model interpretability. *arXiv preprint arXiv:2007.07584*, 2020. pages 36, 37, 54

J. Parekh, P. Mozharovskyi, and F. d'Alché Buc. A framework to learn with interpretation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. page 93

S. Parekh, A. Ozerov, S. Essid, N. Q. Duong, P. Pérez, and G. Richard. Identify, locate and separate: Audio-visual object extraction in large video collections using weak supervision. In *2019 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pages 268–272. IEEE, 2019. page 83

D. H. Park, L. A. Hendricks, Z. Akata, A. Rohrbach, B. Schiele, T. Darrell, and M. Rohrbach. Multimodal explanations: Justifying decisions and pointing to the evidence. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8779–8788, 2018. page 20

A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32:8026–8037, 2019. page 124

K. J. Piczak. Esc: Dataset for environmental sound classification. In *Proceedings of the 23rd ACM international conference on Multimedia*, pages 1015–1018, 2015. pages 89, 91

G. Plumb, M. Al-Shedivat, Á. A. Cabrera, A. Perer, E. Xing, and A. Talwalkar. Regularizing black-box models for improved interpretability. *Advances in Neural Information Processing Systems*, 33:10526–10536, 2020. page 17

P. E. Pope, S. Kolouri, M. Rostami, C. E. Martin, and H. Hoffmann. Explainability methods for graph convolutional neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10772–10781, 2019. page 20

H. Purwins, B. Li, T. Virtanen, J. Schlüter, S.-Y. Chang, and T. Sainath. Deep learning for audio signal processing. *IEEE Journal of Selected Topics in Signal Processing*, 13: 206–219, 2019. page 8

L. Qiao, W. Wang, and B. Lin. Learning accurate and interpretable decision rule sets from neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 4303–4311, 2021. page 31

F. Radenovic, A. Dubey, and D. Mahajan. Neural basis models for interpretability. *arXiv preprint arXiv:2205.14120*, 2022. pages 20, 45, 109

N. F. Rajani, B. McCann, C. Xiong, and R. Socher. Explain yourself! leveraging language models for commonsense reasoning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4932–4942, 2019. page 30

S. Reed, Z. Akata, H. Lee, and B. Schiele. Learning deep representations of fine-grained visual descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 49–58, 2016. page 31

M. T. Ribeiro, S. Singh, and C. Guestrin. Why should i trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144. ACM, 2016. pages 16, 18, 20, 21, 22, 24, 32, 33, 64, 66, 94, 134

L. Rieger, C. Singh, W. Murdoch, and B. Yu. Interpretations are useful: penalizing explanations to align neural networks with prior knowledge. In *International conference on machine learning*, pages 8116–8126. PMLR, 2020. page 108

N. J. Roese. Counterfactual thinking. *Psychological bulletin*, 121(1):133, 1997. page 31

R. Rombach, P. Esser, and B. Ommer. Making sense of cnns: Interpreting deep representations and their invariances with inns. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVII 16*, pages 647–664. Springer, 2020. page 112

A. S. Ross, M. C. Hughes, and F. Doshi-Velez. Right for the right reasons: Training differentiable models by constraining their explanations. *arXiv preprint arXiv:1703.03717*, 2017. page 20

C. Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5): 206–215, 2019. page 17

C. Rudin, C. Chen, Z. Chen, H. Huang, L. Semenova, and C. Zhong. Interpretable machine learning: Fundamental principles and 10 grand challenges. *Statistic Surveys*, 16:1–85, 2022. pages 9, 10

M. H. Sarhan, N. Navab, A. Eslami, and S. Albarqouni. Fairness by learning orthogonal disentangled representations. In *European Conference on Computer Vision*, pages 746–761. Springer, 2020. page 108

A. Sarkar, D. Vijaykeerthy, A. Sarkar, and V. 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. pages 17, 30, 52, 109

Y. Sawada and K. Nakamura. Concept bottleneck model with additional unsupervised concepts. *IEEE Access*, 10:41758–41765, 2022. page 30

T. Schnake, O. Eberle, J. Lederer, S. Nakajima, K. T. Schütt, K.-R. Müller, and G. Montavon. Higher-order explanations of graph neural networks via relevant walks. *IEEE transactions on pattern analysis and machine intelligence*, 44(11):7581–7596, 2021. page 20

M. Schoeffler, S. Bartoschek, F.-R. Stöter, M. Roess, S. Westphal, B. Edler, and J. Herre. webmushra—a comprehensive framework for web-based listening tests. *Journal of Open Research Software*, 6(1), 2018. page 146

K. Schulz, L. Sixt, F. Tombari, and T. Landgraf. Restricting the flow: Information bottlenecks for attribution. In *International Conference on Learning Representations*, 2019. pages 21, 24, 33, 141

K. T. Schütt, F. Arbabzadah, S. Chmiela, K. R. Müller, and A. Tkatchenko. Quantum-chemical insights from deep tensor neural networks. *Nature communications*, 8(1): 13890, 2017. page 8

R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. 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. pages 16, 21, 23, 33, 35, 142

R. R. Selvaraju, P. Tendulkar, D. Parikh, E. Horvitz, M. T. Ribeiro, B. Nushi, and E. Kamar. Squinting at vqa models: Introspecting vqa models with sub-questions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10003–10011, 2020. page 20

E. H. Shortliffe and B. G. Buchanan. A model of inexact reasoning in medicine. *Mathematical Biosciences*, 23(3):351 – 379, 1975. ISSN 0025-5564. doi: https://doi.org/10.1016/0025-5564(75)90047-4. URL http://www.sciencedirect.com/science/article/pii/0025556475900474. page 15

A. Shrikumar, P. Greenside, A. Shcherbina, and A. Kundaje. Not just a black box: Learning important features through propagating activation differences. *arXiv preprint arXiv:1605.01713*, 2016. page 23

K. Simonyan, A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013. pages 15, 23, 33

P. Smaragdis. Non-negative matrix factor deconvolution; extraction of multiple sound sources from monophonic inputs. In *International Conference on Independent Component Analysis and Signal Separation*, pages 494–499. Springer, 2004. page 83

D. Smilkov, N. Thorat, B. Kim, F. Viégas, and M. Wattenberg. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*, 2017. pages 16, 20

I. Sobieraj, L. Rencker, and M. D. Plumbley. Orthogonality-regularized masked nmf for learning on weakly labeled audio data. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2436–2440. IEEE, 2018. page 108

J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*, 2014. pages 16, 23, 125, 128, 141

J. Strout, Y. Zhang, and R. J. Mooney. Do human rationales improve machine explanations? *arXiv preprint arXiv:1905.13714*, 2019. page 20

I. Sturm, S. Lapuschkin, W. Samek, and K.-R. Müller. Interpretable deep neural networks for single-trial eeg classification. *Journal of neuroscience methods*, 274:141–145, 2016. page 8

M. Sundararajan, A. Taly, and Q. Yan. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3319–3328. JMLR. org, 2017. pages 16, 20, 37, 46

A. Sydorova, N. Poerner, and B. Roth. Interpretable question answering on knowledge bases and text. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4943–4951, 2019. page 30

A. Taeb, N. Ruggeri, C. Schnuck, and F. Yang. Provable concept learning for interpretable predictions using variational inference. *arXiv preprint arXiv:2204.00492*, 2022. page 112

V. Y. Tan and C. Févotte. Automatic relevance determination in nonnegative matrix factorization with the/spl beta/-divergence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(7):1592–1605, 2012. page 101

F. Thomas, I. F. R. Rodriguez, D. Linsley, and T. Serre. Harmonizing the object recognition strategies of deep neural networks with humans. In *Advances in Neural Information Processing Systems*, 2022. page 24

Y. Tian, C. Guan, J. Goodman, M. Moore, and C. Xu. Audio-visual interpretable and controllable video captioning. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition workshops*, 2019. page 20

H. Van Luong, B. Joukovsky, and N. Deligiannis. Designing interpretable recurrent neural networks for video reconstruction via deep unfolding. *IEEE Transactions on Image Processing*, 30:4099–4113, 2021. page 21

P. Voigt and A. Von dem Bussche. The eu general data protection regulation (gdpr). *A Practical Guide, 1st Ed., Cham: Springer International Publishing*, 10(3152676):10–5555, 2017. pages 8, 17

A. Voulodimos, N. Doulamis, A. Doulamis, E. Protopapadakis, et al. Deep learning for computer vision: A brief review. *Computational intelligence and neuroscience*, 2018, 2018. page 8

M. Vu and M. T. Thai. Pgm-explainer: Probabilistic graphical model explanations for graph neural networks. *Advances in neural information processing systems*, 33: 12225–12235, 2020. page 20

S. Wachter, B. Mittelstadt, and C. Russell. Counterfactual explanations without opening the black box: Automated decisions and the gpdr. *Harv. JL & Tech.*, 31:841, 2017. page 31

C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The caltech-ucsd birds-200-2011 dataset. 2011. pages 64, 73, 114

A. R. Webb. *Statistical pattern recognition*. John Wiley & Sons, 2003. page 7

K. W. Wilson, B. Raj, P. Smaragdis, and A. Divakaran. Speech denoising using non-negative matrix factorization with priors. In *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 4029–4032. IEEE, 2008. page 83

S. Wisdom, T. Powers, J. Pitton, and L. Atlas. Deep recurrent nmf for speech separation by unfolding iterative thresholding. In *2017 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pages 254–258. IEEE, 2017. page 83

M. Won, S. Chun, and X. Serra. Toward interpretable music tagging with self-attention. *arXiv preprint arXiv:1906.04972*, 2019. page 34

X. Wu, L. Xiao, Y. Sun, J. Zhang, T. Ma, and L. He. A survey of human-in-the-loop for machine learning. *Future Generation Computer Systems*, 2022. page 21

H. Xiao, K. Rasul, and R. Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017. page 64

X. Xing, F. Yang, H. Li, J. Zhang, Y. Zhao, M. Gao, J. Huang, and J. Yao. An interpretable multi-level enhanced graph attention network for disease diagnosis with gene expression data. In *2021 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 556–561. IEEE, 2021. page 21

X. Yao, A. Newson, Y. Gousseau, and P. Hellier. A latent transformer for disentangled face editing in images and videos. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 13789–13798, 2021. page 116

X. Yao, A. Newson, Y. Gousseau, and P. Hellier. A style-based gan encoder for high fidelity reconstruction of images and videos. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XV*, pages 581–597. Springer, 2022. pages 116, 117

C.-K. Yeh, C.-Y. Hsieh, A. Suggala, D. I. Inouye, and P. K. Ravikumar. On the (in) fidelity and sensitivity of explanations. *Advances in Neural Information Processing Systems*, 32, 2019a. page 36

C.-K. Yeh, B. Kim, S. O. Arik, C.-L. Li, P. Ravikumar, and T. Pfister. On concept-based explanations in deep neural networks. *arXiv preprint arXiv:1910.07969*, 2019b. page 33

F. Yin, Z. Shi, C.-J. Hsieh, and K.-W. Chang. On the faithfulness measurements for model interpretations. *arXiv preprint arXiv:2104.08782*, 2021. page 78

Z. Ying, D. Bourgeois, J. You, M. Zitnik, and J. Leskovec. Gnnexplainer: Generating explanations for graph neural networks. *Advances in neural information processing systems*, 32, 2019. page 20

J. Yoon, J. Jordon, and M. van der Schaar. Invase: Instance-wise variable selection using neural networks. In *International Conference on Learning Representations*, 2018. pages 17, 20

H. Yuan, J. Tang, X. Hu, and S. Ji. Xgnn: Towards model-level explanations of graph neural networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 430–438, 2020. page 20

H. Yuan, H. Yu, J. Wang, K. Li, and S. Ji. On explainability of graph neural networks via subgraph explorations. In *International Conference on Machine Learning*, pages 12241–12252. PMLR, 2021. pages 20, 110

M. Yuksekgonul, M. Wang, and J. Zou. Post-hoc concept bottleneck models. *arXiv preprint arXiv:2205.15480*, 2022. page 30

M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I 13*, pages 818–833. Springer, 2014. page 44

J. Zhang, S. A. Bargal, Z. Lin, J. Brandt, X. Shen, and S. Sclaroff. Top-down neural attention by excitation backprop. *International Journal of Computer Vision*, 126(10): 1084–1102, 2018a. page 37

Q. Zhang, Y. Nian Wu, and S.-C. Zhu. Interpretable convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8827–8836, 2018b. page 17

R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018c. page 113

P. Zinemanas, M. Rocamora, M. Miron, F. Font, and X. Serra. An interpretable deep learning model for automatic sound classification. *Electronics*, 10(7):850, 2021. pages 19, 26, 27, 34, 81, 93, 94, 103

Y. Zuo and R. Serfling. General notions of statistical depth function. *The Annals of Statistics*, 28(2):461–482, 2000. page 70

ECOLE
DOCTORALE

Titre : Un cadre flexible pour l'apprentissage automatique interprétable: application à la classification d'images et d'audio

Mots clés : Interprétabilité, Explicabilité, Apprentissage automatique, L'apprentissage profond

Résumé : Les systèmes d'apprentissage automatique, et en particulier les réseaux de neurones, ont rapidement développé leur capacité à résoudre des problèmes d'apprentissage complexes. Par conséquent, ils sont intégrés dans la société avec une influence de plus en plus grande sur tous les niveaux de l'expérience humaine. Cela a entraîné la nécessité d'acquérir des informations compréhensibles par l'homme dans leur processus de prise de décision pour s'assurer que les décisions soient prises de manière éthique et fiable. L'étude et le développement de méthodes capables de générer de telles informations constituent de manière générale le domaine de l'apprentissage automatique interprétable. Cette thèse vise à développer un nouveau cadre pour aborder deux problématiques majeures dans ce domaine, l'interprétabilité post-hoc et par conception. L'interprétabilité post-hoc conçoit des méthodes pour analyser les décisions d'un modèle prédictif pré-entraîné, tandis que l'interprétabilité par conception vise à apprendre un modèle unique capable à la fois de prédiction et d'interprétation. Pour ce faire, nous étendons la formulation traditionnelle de l'apprentissage supervisé pour inclure l'interprétation en tant que tâche supplémentaire en plus de la prédiction, chacune étant traitée par des modèles distincts, mais liés, un prédicteur et un interpréteur. Fondamentalement, l'interpréteur dépend du prédicteur à travers ses couches cachées et utilise un dictionnaire de concepts comme représentation pour l'interprétation avec la capacité de générer des interprétations locales et globales. Le cadre est instancié séparément pour résoudre les problèmes d'interprétation dans le contexte de la classification d'images et de sons. Les deux systèmes ont fait l'objet d'une évaluation approfondie de leurs interprétations sur de multiples ensembles de données publics. Dans les deux cas, nous démontrons des performances de prédiction élevées, ainsi qu'une haute fidélité des interprétations. Bien qu'ils adhèrent à la même structure sous-jacente, les deux systèmes sont distinctement conçus pour l'interprétation. Le système d'interprétabilité des images fait avancer le protocole de découverte des concepts appris pour une meilleure compréhension, laquelle est évaluée qualitativement. Le système d'interprétabilité audio est, quant à lui, conçu avec une nouvelle représentation basée sur une factorisation matricielle non-négative pour faciliter les interprétations écoutables, tout en modélisant les objets audio composant une scène.

Title : A flexible framework for interpretable machine learning: application to image and audio classification

Keywords : Interpretability, Explainability, Machine learning, Deep learning

Abstract : Machine learning systems and specially neural networks, have rapidly grown in their ability to address complex learning problems. Consequently, they are being integrated into society with an ever-rising influence on all levels of human experience. This has resulted in a need to gain human-understandable insights in their decision making process to ensure the decisions are being made ethically and reliably. The study and development of methods which can generate such insights broadly constitutes the field of interpretable machine learning. This thesis aims to develop a novel framework that can tackle two major problem settings in this field, post-hoc and by-design interpretation. Post-hoc interpretability devises methods to interpret decisions of a pre-trained predictive model, while by-design interpretability targets to learn a single model capable of both prediction and interpretation. To this end, we extend the traditional supervised learning formulation to include interpretation as an additional task besides prediction, each addressed by separate but related models, a predictor and an interpreter. Crucially, the interpreter is dependent on the predictor through its hidden layers and utilizes a dictionary of concepts as its representation for interpretation with the capacity to generate local and global interpretations. The framework is separately instantiated to address interpretability problems in the context of image and audio classification. Both systems are extensively evaluated for their interpretations on multiple publicly available datasets. We demonstrate high predictive performance and fidelity of interpretations in both cases. Despite adhering to the same underlying structure the two systems are designed differently for interpretations. The image interpretability system advances the pipeline for discovering learnt concepts for improved understandability that is qualitatively evaluated. The audio interpretability system instead is designed with a novel representation based on non-negative matrix factorization to facilitate listenable interpretations whilst modeling audio objects composing a scene.

Institut Polytechnique de Paris
91120 Palaiseau, France