universität wien

MASTERARBEIT | MASTER'S THESIS

Titel | Title

Information Theory for Personalised Explainable AI

verfasst von | submitted by

Lukas Anton Karner BSc

angestrebter akademischer Grad | in partial fulfilment of the requirements for the degree of

Master of Science (MSc)

Wien | Vienna, 2026

Studienkennzahl lt. Studienblatt | Degree programme code as it appears on the student record sheet:	UA 066 645
Studienrichtung lt. Studienblatt | Degree programme as it appears on the student record sheet:	Masterstudium Data Science
Betreut von | Supervisor:	Assoz. Prof. Dipl.-Ing. Dr.techn. Sebastian Tschiatschek BSc

Acknowledgements

> A man on a thousand-mile walk has to forget his ultimate goal and say to himself every morning, 'Today I'm going to cover twenty-five miles and then rest up and sleep.'
>
> *Leo Tolstoy, War and Peace*

I would like to thank my parents, Ingeborg and Gerhard, for all their love and continuous support that they have given me throughout the years of my studies and beyond. It is impossible to put into words how grateful I am to have such a wonderful family.

Also, I would like to thank my supervisor Sebastian for the freedom he gave me in exploring this topic and his dedication to the advancement of our field.

Abstract

An important aspect of the widespread usage of AI models is the increasing demand for their interpretability and, thus, XAI has seen a surge of interest over the past years. Consequentially, the field of XAI is constantly bringing forth a large number of methods to provide explanations for the ever-growing number of AI models deployed. Many of these methods claim to enhance the understanding of these models by their users, but fall short of providing quantitative justification, in the form of theoretical motivation or rigorous evaluation, for this claim. To overcome this lack of sound explanation methods, based on tools from information theory, this thesis introduces, to the best of our knowledge, the first model-agnostic method for creating local user-adaptive explanations. Our method enables users to improve their model understanding through personalised explanations adapted to their existing knowledge of the model, and it is backed up by a strong theoretical motivation and rigorous evaluation methods introduced along with it. To demonstrate the user-adaptivity of our explanation framework, we conduct experiments with a range of simulated users representing various levels of model understanding and use information-theoretic metrics to measure the increase in model understanding provided by the explanations. Furthermore, as it is necessary for assuring the truthfulness of explanations produced by our method, we identify the problem of *output encoding* in explanations created through the L2X framework (Chen et al., 2018), develop information-theoretic tools to fix it, and provide an empirical characterisation of its occurrence. Altogether, this thesis provides a contribution towards helping users to better understand the AI models they are dealing with, and towards improving our information-theoretical understanding of explanations for AI models as a whole.

Kurzfassung

Ein wichtiger Aspekt der weit verbreiteten Verwendung von AI-Modellen ist die steigende Nachfrage nach ihrer Interpretierbarkeit, weshalb *erklärbare künstliche Intelligenz* (XAI) in den letzten Jahren einen sprunghaften Anstieg des Interesses verzeichnet hat. Infolgedessen bringt der Bereich der XAI ständig eine Vielzahl von Methoden hervor, um Erklärungen für die ständig wachsende Zahl der eingesetzten AI-Modelle zu liefern. Viele dieser Methoden behaupten, das Verständnis dieser Modelle durch ihre Nutzer zu verbessern, liefern jedoch keine quantitative Begründung in Form einer theoretischen Motivation oder einer rigorosen Evaluation für diese Behauptung. Um diesen Mangel an fundierten Erklärungsmethoden zu beheben, stellt diese Masterarbeit auf der Grundlage von Begriffen aus der Informationstheorie die unseres Wissens nach erste modellunabhängige Methode zur Erstellung lokaler, nutzeradaptiver Erklärungen vor. Unsere Methode ermöglicht es den Nutzern, ihr Modellverständnis durch personalisierte Erklärungen zu verbessern, die an ihr vorhandenes Wissen über das Modell angepasst sind, und wird durch eine starke theoretische Begründung und strenge Evaluierungsmethoden untermauert, die zusammen mit ihr eingeführt werden. Um die Nutzeradaptivität unserer Erklärungsmethode zu demonstrieren, führen wir Experimente mit einer Reihe von simulierten Nutzern durch, die verschiedene Niveaus des Modellverständnisses repräsentieren, und verwenden informationstheoretische Metriken, um die durch die Erklärungen erzielte Verbesserung des Modellverständnisses zu messen. Da es außerdem notwendig ist um die Richtigkeit der mit unserer Methode erstellten Erklärungen sicherzustellen, identifizieren wir das Problem des *output encoding* in Erklärungen, die mit dem L2X-Framework (Chen et al., 2018) erstellt wurden, entwickeln informationstheoretische Werkzeuge, um es zu beheben, und liefern eine empirische Charakterisierung seines Auftretens. Insgesamt leistet diese Arbeit einen Beitrag dazu, Nutzern ein besseres Verständnis der von ihnen verwendeten AI-Modelle zu ermöglichen und unser informationstheoretisches Verständnis von Erklärungen für AI-Modelle insgesamt zu verbessern.

Contents

1. Introduction

In the last years, the world has seen an ever-increasing use of *Artificial Intelligence* (AI) technology, especially in the form of *Machine Learning* (ML) models. As a result of this development, the necessity of *Explainable Artificial Intelligence* (XAI) has become increasingly obvious, and the demand for it has reached ever-new heights [XUD+19]. In this thesis, we will approach the topic of XAI from a perspective not only focusing on producing explanations in general but aiming to develop a framework that adapts its explanations to the knowledge of its user.

1.1. XAI From a User Perspective

As XAI is becoming an evermore sought-after topic, let us see what some reasons for this are. Until recently, many applications of AI relied on intrinsically interpretable ML models, so-called white-box models. However, as we strive for ever more performant models, we are pushing towards the boundaries of what these simple models can do. Consequentially, we see a constantly growing use of black-box models which are not easily interpretable by humans, and hence require assistance in the form of XAI [XUD+19, GA19].

Further, one can observe AI being introduced in ever more sensitive areas. For example, AI is used in the medical field to aid doctors with their treatment of conditions, or to detect early whether treatments are effective [KHSMP26]. In such applications, XAI is needed to comply with medical ethics guidelines and to make sure that the used models work correctly [TG21, XUD+19]. In finance, AI is employed in areas like risk management or portfolio optimisation where strict regulations demand high levels of transparency and thus make XAI a necessity for the lawful use of AI [WCH23]. This also has the consequence that the use of AI is beginning to be regulated by legislation, which is often demanding explainability of the used models [HP22].

Finally, there is the user aspect in the drive for XAI which is based on the fact that people want to understand the models they are dealing with [SKMT25], especially if the users have to accept consequences of these models' outputs [XUD+19]. For example, users of a streaming service are more willing to accept a recommendation for what content to consume next, if it comes with some justification of why they are presented with it [MLH+18].

Due to the great demand for XAI in the past years, a large number of frameworks for it have been developed [HSM+22, IEGA21]. Some of the most widely used XAI methods include LIME [RSG16], SHAP [LL17], and LRP [BBM+15]. Among these, LIME is based on approximating the model to be explained by a white-box model, while SHAP builds on a strong motivation from game theory to create its explanations. LRP, on the other

hand, makes use of the Taylor decompositions of the layers of neural networks to infer the importance of input features, but it is only applicable to neural networks for this reason.

These examples serve to illustrate some of the variety that exists among the various XAI methods. But, despite the large number of XAI methods already available, there is no single method that is considered to solve the problem of XAI, as each method has advantages and shortcomings [HSM+22]. Therefore, as a starting point for the work presented in this thesis, we raise the following issues:

I1 *Which method to use and why?* There are many different XAI methods, which all have different theoretical motivations, and there is no guarantee that they will produce consistent results. Moreover, in the entire field of XAI, there does not even exist an agreed-upon definition of what an explanation is or what constitutes a good explanation [RTL19, SO23]. Thus, instead of relying on rigorous evaluation, XAI methods are often backed up only by heuristic reasoning of why their explanations provide insights into the model's black box. The review [AB18] found that only 5% of the reviewed XAI papers presented quantitative evaluations of their explanations. As a consequence, XAI methods are sometimes applied in the wrong context, or their explanations are misinterpreted, which counteracts their purpose [MKH+22, LV22]. Altogether, we have the situation that there is no straightforward way to decide which method is the most suitable for a given application and produces the best explanations [SO23]. Thus, it is necessary to create XAI methods for which it is clear which tasks they are suitable for and what kind of explanations they produce so that they can be used effectively.

I2 *Different explanations for different users?* So far, XAI methods have mainly been designed for professional users who, on the one hand, have a solid understanding of the theory behind the involved ML models anyway, and, on the other hand, use them in specific technical contexts [CCN+21, SO23]. However, it has been pointed out that the explanations that such methods produce might not be helpful at all for the majority of potential users since they might either lack the necessary knowledge to interpret them [JN20], they might be interested in different explanation contexts such as regulatory evaluation of models [SO23], or they might not be interested in such technical explanations at all [RTL19, SKMT25]. For this reason, we must develop XAI methods that make AI understandable to a broader population, since it will be essential for both the acceptance of AI by the people and also to enable the people to verify if AI technology is being used fairly.

To address these questions, we propose a new explanation framework that is based on information theory and which builds on the existing XAI frameworks *Learning to Explain* (L2X) [CSWJ18], and the framework proposed by Jung and Nardelli (JN) [JN20]. The L2X method is a local explanation method that makes use of information theory to create explanations maximising the information they convey about the model to be explained. The method proposed by JN is also aiming to maximise the information about the model in its explanations, but it does so taking account of the understanding of the model that the user already has, and it produces only global explanations.

Information theory has proven to be very effective in describing phenomena in a wide range of fields, including communication theory, computer science, mathematics, physics, statistics, and economics, [CT12]. Thus, information theory is supported by both a strong theoretical and empirical motivation, making it a natural choice for designing explanations with clear motivation and application scenarios. Therefore, as pointed out by L2X and JN already, information-theory-based explanations are very suitable to answer the question about which XAI methods are to be used in certain applications raised in issue I1. Furthermore, the user adaptivity of JN enabled by the flexibility of its information-theory-based explanations makes it an ideal candidate to build on towards answering the question of whether different users require different explanations raised in issue I2.

Thus, to define our framework, we make use of the fact that both L2X and JN have advantages that are complementary to each other, and that their information-theory-based explanation objectives are similar enough to be fused together. We propose a unified framework for creating local explanations that combines the advantages of L2X and JN, and which resolves issues I1 through a strong and versatile theoretical motivation provided by information theory, and resolves issue I2 through user-adaptive explanations to personalise the information conveyed by explanations to the knowledge of their users.

1.2. Contributions

Based on the problems in XAI identified in section 1.1 and the criticism of the XAI frameworks L2X [CSWJ18] and JN [JN20] raised in section 3.2, in this thesis, we will attempt to answer the following research question:

> Is it possible to integrate the user-adaptiveness of JN into the L2X framework to obtain an explanation method that offers local user-adaptive explanations based on information theory, and which produces suitable explanations?

To answer this question, we propose a mutual-information-based method to create local user-adaptive explanations based on L2X and JN.

To ensure the truthfulness of the explanations produced by our framework, we raise a theoretical criticism of the explanation objective proposed by L2X. There, we identify the problem of *output encoding* in the explanations produced by L2X, which consists of arbitrary encodings of model outputs in the explanations produced by L2X and degrades their meaningfulness. To mitigate this problem, we develop tools of information theory to resolve it in the explanation objective of L2X, and conduct experiments to demonstrate the occurrence of output encoding in L2X explanations. As a result of these experiments, we are able to derive empirical rules characterising the occurrence of output encoding in L2X explanations based on the size of the used explanation masks. Furthermore, in these experiments, we also evaluate our modifications to the explanation objective of L2X to resolve the problem of output encoding and show that they are successful at doing so.

At last, we implement our proposed explanation method for creating local user-adaptive explanations based on the insights gathered from the problems encountered with L2X and examine the quality of the explanations produced by it. In these experiments, we find that the explanations produced by solving the theoretical objective of our framework exactly are very informative for a wide range of user profiles. However, we also find that it is mathematically impossible to implement our framework in a tractable way so that it does not suffer from output encoding. Finally, we examine the prevalence of output encoding in explanations produced by a tractable implementation of our framework and provide a partial characterisation of its occurrence based on empirical observations.

The remainder of this thesis is structured as follows: In chapter 2 we will discuss the existing literature on personalisation in XAI; in chapter 3 we will introduce the tools from information theory necessary for our method and more closely examine the methods L2X and JN on which our method will build; in chapter 4 we will define the notions necessary to describe our framework; in chapter 5 we will discuss our theoretical concerns about the L2X method, develop the theory necessary to fix it, and empirically validate these steps; and in chapter 6 we will introduce our explanation framework, discuss its implementation, and evaluate the explanations produced by it regarding their quality.

The code implementing the framework and experiments presented in this thesis is publicly available in the repository [1].

[1] https://github.com/LukasKarner/IT4PXAI

2. Related Work

As we pointed out in chapter 1, so far XAI methods have mainly been developed for professional users who have a sound understanding of the theoretical aspects of AI and the ML models to be explained anyways. The research community, however, is aware that this poses a problem for making explanations accessible to a wider population and thus has increasingly focused on creating XAI methods that produce personalised explanations for their users. To provide a broader foundation for our own proposed method, in this chapter, we will give an overview of the research that has been done already in the area of personalised XAI and discuss the aspects of it that will be relevant for our work.

Requirements to Personalised Explanations

First of all, there are papers which lay out general principles for creating personalised explanations and discuss how they can be derived from existing work.

Initial steps in creating personalised explanations were outlined by [RTL19] which provided a characterisation of the goals, contents and types of XAI methods that had been developed so far and further discussed how one can use the different modalities of different explanation methods to cater to the needs of different groups of users, specifically AI researchers, domain experts and lay users.

A more thorough conceptualisation of personalised explanations was then done by [SH19], which, in addition to deriving principles for personalised explanations from existing work in XAI and personalised ML, gives a profound taxonomy of the concepts relevant in each step of the personalised XAI pipeline. Thus, [SH19] makes several important observations about the nature of personalised explanations, provides an overview of the initial work done on personalised explanations and also discusses ways to evaluate them. In particular, we want to highlight the following points raised by [SH19]: They argue that to create personalised explanations, the user must provide data on themself, which is then used together with data for non-personalised explanations as input to the personalised explanation methods. Such a personalised explanation method can make use of the internals of an ML model, its decision, training data, user data and information derived from it. From their point of view, a main problem of the existing methods for personalised explanations is that so far personalised explanations are only considered in settings where personalised predictions are explained (especially in the area of recommendation systems), and there it is done not by collecting and using information about users but just by explaining the personalised prediction model. Furthermore, [SH19] goes on to define an extensive list of desiderata and concepts of personalised explanations. In particular, among the desiderata, we want to highlight fidelity and effort. Fidelity in this context is understood as the degree to which the explanations match the

predictions of the model, and effort is understood as both the effort of the user necessary to create the personalised explanations and the effort necessary to understand them. Among the presented concepts, we would like to highlight personalisation granularity and personalisation automation. By personalisation granularity [SH19] refers to how personalisable an explanation method is, i.e. starting from explanations that are crafted for whole groups of users (e.g. experts and non-experts) down to bespoke explanations for single users' preferences. Personalisation automation describes the range of possibilities of how the personalisation is performed, starting from the user themself (e.g. by setting explanation parameters) up to the explanation method operating fully autonomously without any action required by the user except for providing information on themself. Regarding the topic of information about the user, [SH19] is also providing a taxonomy of the different kinds of user data that can be employed to create personalised explanations and of methods to obtain them. Finally, [SH19] discusses the various ways in which one can use the concepts introduced before to create personalised explanations and how to evaluate these explanations against the listed desiderata. For the two desiderata, fidelity and effort that we highlighted above, the suggested way to evaluate them is, in the case of fidelity, by inspection of the explanations applied to white-box models, and, in the case of effort, by measuring the cognitive load of users consuming the explanations.

Personalisation as an Aspect of General Explanations

Secondly, there are papers with a main focus on general XAI but which also contain useful observations on personalised explanations.

Here we have [LV22] which does not cover personalised explanations directly, but is approaching human-centred XAI from the perspective of research in human-computer interaction and user experience design. Mainly, [LV22] criticises that existing XAI methods, in general, take a too technology-centred approach to the topic of explainability, while it is an inherently human subject. Along these lines, [LV22] discusses how humans understand classical XAI methods, various pitfalls and possibilities for misunderstanding that they pose, and what explanations are from the perspective of cognitive science. Out of the many points that this paper raises, we would like to emphasise the following: Generally, [LV22] argues that there is no "one-fits-all" solution in the vast and rapidly growing collection of XAI methods, and further, they argue that the choice of the XAI method to use should be based on the user's explainability needs. This is a challenging endeavour, though, they argue, as the users of XAI do not constitute a uniform group and their explainability needs can vary significantly depending on their goals, backgrounds, and usage contexts. Furthermore, they lament that XAI methods are primarily developed to help AI researchers inspect the models and not to help users understand them. Thus, they conclude that the helpfulness of XAI to end users is not given and should get more attention in XAI research. As a first step in this direction, [LV22] conducted a user study to map out the common explainability needs of users and compiled them into a system that can be used as a tool to identify appropriate explanation needs for users. To prioritise the explainability needs of users [LV22] suggests reframing the objective of XAI away from the technical space towards the user questions that each XAI method can address.

The meta-survey [SO23] is concerned with mapping out challenges and future research directions in XAI. These are categorised either as general challenges and research directions of XAI or challenges and research directions of XAI based on the different phases of the ML life cycle: design, development, and deployment. First of all, [SO23] highlights that explainability is not a singular concept, but that it is needed and needs to be viewed from various perspectives, namely the scientific, regulatory, developer, industrial, and end-user perspectives. Then, [SO23] goes on to discuss the 39 challenges and research directions in XAI which they identified through a systematic literature review of papers published before September 2021. Among the general issues in XAI that were identified are, for example: XAI for trustworthy AI, multidisciplinary research collaborations, challenges in existing XAI methods, the communication of uncertainties, and natural language explanations. As most notable for personalised explanations, however, we would like to highlight the general challenges of "towards more formalism" and "explanations and the nature of user experience and expertise", and the deployment challenges of "human-machine teaming" and "XAI and privacy". The aspects of the problem of achieving more formalism in XAI that were found by [SO23] are numerous, and many of them can be considered foundational. As main issues in this area [SO23] found the fact that the field of XAI as a whole lacks consistent definitions and usage of notions and concepts, and that there is no satisfying way to evaluate XAI methods and decide which yields the best explanations. Further, [SO23] notes that many methods have been proposed to tackle explanations in very specialised settings and, thus, states that there is a need to build generic XAI frameworks that allow for easily obtaining end-to-end XAI solutions. In the domain of explanations and user experience and expertise [SO23] found that works in XAI generally neglected to account for the knowledge, goals, and skills of users while it is considered important to do so because the users of ML models can vary and the users' knowledge and the complexity that they expect in their explanations greatly influences the user experience. Thus, [SO23] argues that, on the one hand, it is crucial to adapt explanations based on user experience and expertise and that they should be conveyed differently to different users, and, on the other hand, that it is necessary to define and respect the goals of users and explanations. Finally, we would like to mention the issues raised by [SO23] at the deployment stage in the ML life cycle. There, in the problem area of human-machine teaming, they found a large part of XAI methods to lack interactivity and personalisation of the explanations, and that much work is still necessary to adapt interfaces for different users. In the area of XAI and privacy, they found the problem of preserving the users' privacy, which is especially urgent in the context of personalised explanations, as employing user data is seen as necessary there.

Impact of Personalisation on Users

Further, some works engage user studies to learn about the effects of personalised explanations on the perception and experience of the users.

There is [CBPR21] which presents a case study about personalised explanations in tutoring systems and seeks to create personalised XAI by giving XAI methods the ability to be aware of when and how to provide explanations to their users. Based on the

assumption that one-fits-all explanations are not ideal and the question of how users interact with and perceive personalised explanations, [CBPR21] conducted a user study on the reception by students of a system providing various forms of personalised explanations for hints displayed in a tutoring application and found a positive impact of personalised explanations on the students' learning performance and their perception of the hints.

Notably, we find the recent empirical work [NCZ+24], which seeks to shed light on the question of how necessary it is to create personalised explanations. To assess this, the authors conducted a user study with 149 participants to find out whether there exist significant correlations between the users' personality traits and how they perceive, engage with and understand non-personalised explanations of a hate-speech detection model. In summary, [NCZ+24] found that user characteristics influenced the measured metrics very little, only finding a significant negative effect of the properties 'age' and 'openness' on the actual understanding of the explanations. However, the authors admit that a negative effect of the personality trait 'openness' on the understanding of XAI is a counter-intuitive result and that it might very well stem from a known weakness in the questionnaire used to assess the personality traits. Hence, this result might be due to a methodological measuring error, and a follow-up study would be necessary to more thoroughly investigate this result. Based on their finding that the users' age is the only characteristic that significantly impacted their interaction with the provided explanations, [NCZ+24] concludes that lay users of XAI might be much more similar to each other than previously assumed and questions if it is necessary to pursue the personalisation of XAI based on fine-grained user characteristics. Furthermore, we would like to highlight two points about personalised explanations raised in the discussion of [NCZ+24] which are relevant to our own proposed method: On the one hand, they argue that if one pursues fine-grained personalisation techniques, it is so far not clear how to efficiently do this as one has to take into account the effort that a user has to put into building a user profile for every personalised XAI application that they want to use, which might outweigh the expected benefits of them. And, on the other hand, [NCZ+24] points out the general problem that there still is no satisfying and agreed-upon way to design explanations to take user characteristics into account and that further research into this is necessary.

Personalised Explanations (in Special Settings)

Now, let us turn towards works which deal with creating personalised explanations.

Most notably, here we find [JN20] to be the only paper to propose a general-purpose explanation method to create personalised explanations for arbitrary ML models. This is done by considering an information-theoretic approach to relate the model outputs, explanations and user knowledge to each other, which we will discuss in chapter 3.

Other works attempt to introduce personalised XAI in specialised settings where they presume it to be useful. The research note [CCN+21] treats XAI in the context of distributed agents with heterogeneous knowledge that require personalised explanations without a central entity mediating them. To achieve this, they propose a broad framework that is defined by several limitations and objectives in distributed XAI that they identified

and which is supposed to address the diverse requirements, preferences and knowledge of the different agents while also respecting their independence and isolation from the rest of the agents.

In information systems research, there is [CMG22] which discusses possible directions for research on personalised explanations while arguing from the perspective of information systems and the various stakeholders involved in this setting. To this end, they define, on the one hand, a characterisation of the different groups of stakeholders and, on the other hand, a set of objectives for each of these groups that personalised explanations for them should fulfil. While explainability has been a research topic of information systems in the past, [CMG22] laments that only little recent work on this topic exists and thus urges for more effort to be put into this area.

In the setting of financial auditing, where explainability is especially relevant for the application of AI due to regulatory pressure, [RRF+22] attempts to establish a theoretical framework for personalised explanations to serve as a role model for future implementations of these. For this, they conducted interviews with domain experts to identify the needs of stakeholders and further evaluated the proposed framework in focus groups.

As a particular instance of personalised XAI in specialised settings, there are several papers in the area of human-robot teaming researching the use of personalised explanations for explaining the actions of robots to the humans collaborating with them.

Here, [BCF21] proposes a system for generating personalised explanations for robot path planning to incorporate feedback from users and develops a method to generate personalised explanations based on the preferences of the users. Additionally, the proposed system is designed to interact with its users by responding to further questions about the generated explanations, and a user study showed that the generated personalised explanations improve user satisfaction.

Next, [SSK21] proposes a system to generate personalised explanations by learning the different types of users that the robot could interact with. According to this method, during the interaction with the user, the system identifies the user type and adjusts its explanations to it. This is achieved by combining a clustering approach with a partially observed Markov decision process, and a user study is conducted to evaluate user satisfaction with the provided explanations.

In the domain of autonomous vehicle operation, [STSG24] conducts user studies on user preferences for explanations in general and different strategies for adapting explanations to them, where they find significant results in the preferences for and performance of explanations. Further, [STSG24] proposes a personalisation strategy to balance user preferences and explanation performance and shows that this strategy yields significant gains in explanation performance.

Finally, since, as [SH19] pointed out, a large part of the existing work on personalised explanations has been done on explanations for personalised recommender systems, for the sake of completeness, let us review some examples of these works, although they do not directly contribute to our setting.

Here, we find, for example [MCV22] which studies how explanations in a music recommender system should be designed to serve the preferences of different users, or [KSP+19] which focuses on the problem of generating and visualising personalised explanations for recommender systems incorporating many different data sources and develops an approach to generate real-time recommendations with personalised explanations.

In [LZC21], the authors study the personalisation of natural language generation for tasks such as explainable recommendations or dialogue systems. For this, they present a modification of the Transformer architecture that incorporates user information, and which, in addition to creating explanations for recommendations, can also be used to create recommendations themselves.

And, at last, there is [GFG+22] which considers the problem of explanation systems for personalised recommendations, often producing unsatisfying results. To mitigate this problem, they propose to incorporate a visual feedback loop into the explanation system, which allows the system to produce more coherent results.

Summary

In conclusion to this literature review, we would like to summarise two issues which were mentioned by multiple sources and are of special importance for our setting. On the one hand, it appears that [JN20] is the only general-purpose method to create personalised explanations in existence, and, on the other hand, the rigorous evaluation of (non-)personalised explanations is very much an open problem.

The lack of general purpose personalised explanation methods was pointed out by [SH19, SO23, NCZ+24] and is corroborated by our observations as [JN20] is the only existing XAI method we found which can be used to create personalised explanations in general settings for any kind of model without the need of hand-engineered or hard-coded explanation techniques. All the other XAI methods for personalised explanations, which we found and portrayed here, are either only applicable to very specialised settings and need a large amount of engineering effort to be ported to different models or explanation tasks, or they are personalised merely insofar as they explain personalised models.

Regarding the problem of the objective evaluation of explanations, we observe that, while it is considered an open problem even for non-personalised explanations [SO23], apparently all proposed ways to evaluate personalised explanations rely on subjective terms such as "user needs" or "user perception" which can only be measured qualitatively in user studies, and hence no rigorous and objective way to evaluate personalised explanations exists.

Thus, although many works are calling for personalised explanations [RTL19, SH19, SO23, LV22], there appears to be a severe lack of research both in the creation of personalised explanations and their evaluation. For this reason, in this thesis, we will attempt to fill this research gap by building on the existing method [JN20] towards a more comprehensive personalised explanation framework.

3. Background

In the last chapter, we identified [JN20] as the only existing XAI method that is capable of creating personalised explanations in general settings, which is why we will use it as the foundation for our own proposed method. To this end, we will discuss its approach more thoroughly in this chapter. As we will see, this approach uses tools from *Information Theory* [CT12] to describe the behaviour of models and users, which is why we will quickly introduce this subject now.

Note that, throughout this thesis, we use log to denote the logarithm of base 2.

3.1. Information Theory in XAI

When viewed from an abstract mathematical perspective, we can formulate the inputs and outputs of an ML model as random variables that follow a joint distribution. In chapter 4 we will treat this formalism more rigorously, but for now let X denote the random variable describing the input data, Y the random variable describing the model output, and $p(X, Y)$ their joint distribution. Then, since the behaviour on inputs and outputs of the model to be explained is fully represented by the conditional distribution $p(Y|X)$ within p, this probability distribution captures all the aspects of a model that are relevant for a model-agnostic XAI method. Hence, one can formulate the problem of XAI entirely in terms of these random variables and their joint distribution. Information theory provides some excellent tools to study these because it was created to characterise systems transmitting information in a rigorous mathematical way, and it does so by modelling them as random variables.

For these reasons, information theory has already been employed in several XAI frameworks. Most notably, [CSWJ18] is the first work to explicitly use concepts from information theory to design explanations, and further [JN20] is the first work to pick up this idea to adapt its explanation to a given user. We will more thoroughly review these two methods in the next section as their ideas will later form the basis for the framework that this thesis proposes.

Other than that, for example, there are [GWZ+19], which uses information theory to improve explanations in the domain of natural language processing, [ZYK+23] uses information theory to select meaningful data representations, and [BXL+21] uses information theory to create brief and comprehensive explanations.

3.2. Mutual Information – The Key to Explanation

Let us now introduce the main tool from information theory by which one can design (personalised) explanations, the so-called *mutual information* of random variables. Consider the random variables X and Y with joint distribution p again. Then the mutual information $I(X, Y)$ of X and Y is given by

$$I(X, Y) = \mathbb{E}\left[\log \frac{p(X, Y)}{p(X)p(Y)}\right]. \tag{3.1}$$

And further, if there is a third random variable Z that admits a joint distribution p with X and Y, then the *conditional mutual information* $I(X, Y|Z)$ of X and Y under Z is given by

$$I(X, Y|Z) = \mathbb{E}\left[\log \frac{p(X, Y|Z)}{p(X|Z)p(Y|Z)}\right]. \tag{3.2}$$

The mutual information of two random variables has many interpretations; however, the classical meaning of it is that it describes the capacity of a communication channel with input X and output Y [CT12]. More intuitively, one can think of it as the amount of information that one gains about X when observing Y (or vice versa). Extending this interpretation, one can then think of the conditional mutual information as the amount of information that one gains about X when observing Y and knowing the result of Z.

The two frameworks on which the contribution of this thesis builds both have in common that they use the mutual information of the explanation itself and the model's output to design their explanations, albeit they do it in somewhat different ways, each of which has its advantages and shortcomings.

For these discussions, we will use the following notations: The random variables X and Y will denote the input and output of the ML model to be explained, and x and y will denote realisations of these random variables. The input data X is generally d-dimensional and consists of features $X = (X_i)_{i=1}^d$. Further, we will consider the random variables S ranging over subsets $X_S = (X_i)_{i \in S}$ of the features of input data, E denoting explanations, U denoting user summaries (defined below), and realisations of these will be denoted by s, e and u respectively. Any (joint) probability distributions of these random variables will be denoted by p.

***Learning to Explain* (L2X) [CSWJ18]** The L2X framework considers so-called *instance-wise feature selection* to explain the predictions of an ML model. That is, for a given input x, it returns a subset s of this input's features as an explanation for the model's prediction on it. To this end, L2X defines an explainer model $\mathcal{E} : x \mapsto s$ which assigns to x a subset s of its features. Notably, a parameter of the explainer model is the *explanation mask size* k which determines the size of the feature subsets s constituting its explanations. To select which feature subset s constitutes the explanation for an input instance x, this explainer model $\mathcal{E}$ is trained on a mutual-information-based objective, and as such L2X was the first XAI method to use mutual information to find its explanations. This objective considers

the model output Y on the one hand, and on the other hand, it considers the random variable X_S that is to be understood as the random variable obtained from X by masking it such that only the features in $S = \mathcal{E}(X)$ are visible. The optimal explainer model $\widehat{\mathcal{E}}$ of L2X is defined as the one maximising the mutual information of X_S with Y under the constraint $|S| \leq k$ limiting the number of features visible in the explanations, i.e.

$$\widehat{\mathcal{E}} = \arg\max_{\mathcal{E}} I(X_S, Y) \quad \text{where} \quad S = \mathcal{E}(X). \tag{3.3}$$

As this global criterion might seem a bit cumbersome to understand, [CSWJ18] claims to prove[1] that criterion (3.3) is, in fact, equivalent to the local and deterministic criterion

$$\widehat{\mathcal{E}}(x) = \arg\min_{s} \mathbb{E}_{Y|X} \left[\log \frac{1}{p(Y|X_s = x_s)} \middle| X = x \right]. \tag{3.4}$$

Unfortunately, this optimisation problem is intractable, which is why the authors of L2X came up with a sophisticated combination of the Gumbel-softmax trick [JGP17, MMT17] and a variational lower bound to circumvent this problem. The explainer model $\mathcal{E}$ and the variational lower bound are implemented as neural networks that are trained on the inputs and outputs of the model to be explained. The training of the explainer is computationally expensive, but at inference L2X is very efficient as it just requires one forward pass through the neural network to compute an explanation.

***Jung and Nardelli* (JN) [JN20]** The motivation of this explanation method is to provide personalised explanations by considering explanations that minimise the surprisingness of the output of an ML model for a user. Thus, this method considers a model of its user's knowledge, and a mutual-information-based objective conditioned on it, to create its explanations. As such, it was the first method to propose using mutual information conditioned on information provided by the user to personalise its explanations. Like L2X, also JN defines its explanations by an explainer model $\mathcal{E} : x \mapsto e$ that assigns an explanation e to a given input x, but the approach of JN is more general than the one of L2X as it allows for explanations e of a general form instead of just subsets s of the input features. Notably, however, in the default implementation that the authors present, the explanations are also in the form of maskings of the observed input. To select its explanations, the explainer model of JN is trained to maximise the mutual information of the model output Y and explanations $E = \mathcal{E}(X)$ conditioned on a so-called *user summary* U. The user summary is understood to be a quantity characterising the user's knowledge, which describes the user's understanding of the model output, and is formally defined by a map $\mathcal{U} : x \mapsto u$. Thus u can be, e.g. a vector of high-level features derived from the input x, or the output of a model trained on a subset of the features of x selected by the user. In any case, however, some form of user input is required to obtain an estimate of u and to obtain explanations in this framework. The best explainer model $\widehat{\mathcal{E}}$ of JN is the map which yields the maximal conditional mutual information of E with Y under U, i.e.

$$\widehat{\mathcal{E}} = \arg\max_{\mathcal{E}} I(E, Y|U) \quad \text{where} \quad E = \mathcal{E}(X). \tag{3.5}$$

[1] This proof will be discussed in section 5.3.

That said, while their explanation objectives are indeed very similar, there is a large difference in the proposed implementations of this framework and L2X to avoid the intractable optimisation problem arising here. Namely, to obtain an implementation, JN makes several simplifying assumptions on the structure of the data, and the nature of the explainer model $\mathcal{E}$ and the user summary map $\mathcal{U}$ to arrive at a problem that can be solved by Lasso regression [Tib96] with target Y and inputs X and U.

As a result of this, the explanation that the implementation in [JN20] provides does not depend on a realisation x of the input X anymore, but is merely a feature subset selected by Lasso regression to explain the model's behaviour in general. This means that JN provides a so-called *global explanation* of the model, in contrast to L2X, which provides so-called *local explanations*, i.e. explanations for the model's behaviour at every input-output pair individually.

To conclude this chapter, let us now summarise our discussion about L2X and JN and put these two methods in the perspective of the two issues I1 and I2 in XAI that we outlined at the end of chapter 1.

For L2X, we have that it aims to produce information-theoretically optimal solutions for every input instance and, hence, is backed up by the strong theoretical motivation provided by information theory. Further, we should also remark that L2X is model-agnostic, which makes it versatile and applicable in general contexts. The downside of L2X is mainly that it does not offer personalised explanations.

On the upside of JN, we find that it provides personalised explanations, that it is model-agnostic, and also that it is motivated by information theory. Thus, it scores on the issues of motivation and user adaptation. On the negative side of JN, there is, on the one hand, the fact that its demonstrated implementation only provides a single generic explanation for the entire data, and on the other hand, that this implementation is only based on very simplifying assumptions.

In summary, both methods have their merits, but there is potential for an XAI framework to resolve the issues that we raised previously. Namely, still, there is no general-purpose XAI method that allows the creation of personalised local explanations, and thus, we will now explore how one can build on L2X and JN to proceed in this direction.

4. Defining the Environment

In this chapter, we will introduce our personalised explanations. To quickly recapitulate, we found that L2X offers local explanations but is not user-adaptive, whereas JN is user-adaptive but does not offer local explanations. To combine the advantages of both methods, our explanations will be produced in a manner which merges the two mutual-information-based approaches of L2X and JN to obtain an XAI method whose explanations are both local and personalised. To do this, we will now define the basic elements of our framework, which we will use throughout the remainder of this thesis.[1]

4.1. From Models to Explanations

In chapter 3, we quickly introduced how L2X and JN approach XAI from an information-theoretical perspective. We will now refine these notions more rigorously to lay out the foundation for our proposed method.

Definition 1 (Model and Data)**.** We denote the model to be explained as a map

$$\mathcal{M} : \mathbb{R}^d \to \mathbb{R}^{d'}, \quad x \mapsto y \tag{4.1}$$

taking as input data $x \in \mathbb{R}^d$ and producing as output a prediction $\mathcal{M}(x) = y \in \mathbb{R}^{d'}$. We treat the model's input data x as realisation of a random variable X on $\mathbb{R}^d$ following the data-generating distribution and, likewise, consider the model's prediction y as realisation of the random variable $Y = \mathcal{M}(X)$ on $\mathbb{R}^{d'}$ following the distribution induced by $\mathcal{M}$. We observe that the joint probability density p of X and Y is entirely determined by the density of the data-generating distribution p_X and the density $p_{Y|X}$ induced by the model. That is, for Ω the sample space of the random variables, $\mathbb{P}$ the induced probability measure, and $d\mathbb{P}/d\nu$ the Radon-Nikodym derivative with appropriate[2] measure ν, we find

$$p : \Omega \to \mathbb{R}, \quad p(x, y) = p_X(x) \cdot p_{Y|X}(y|x) = d\mathbb{P}(X = x)/d\nu_X \cdot d\mathbb{P}(\mathcal{M}(x) = Y)/d\nu_Y.^3 \tag{4.2}$$

Next, we will define our main tool for creating explanations based on the explainer models that we have seen in the existing methods.

[1] On the notation of random variables: Commonly, random variables X are denoted by upper case letters while instances x of them are written in lower case. This is important in the theoretical discussion of our framework as, in information theory, only random variables X carry information, while single realisations x of them do not do so.

[2] Meaning that the random variables are absolutely continuous with respect to ν so that the Radon-Nikodym derivatives exist and define their densities (cf. Radon-Nikodym Theorem 9.36 in [Axl19]).

[3] For the sake of readability, we will omit the subscripts of the probability distribution p whenever it is clear which conditional distribution we are referring to.

Definition 2 (Explainer Model)**.** To obtain explanations for a given input, we define an *explainer model* $\mathcal{E}$ of explanation mask size $k \in \mathbb{N}$ with $0 < k < d$ as a map

$$\mathcal{E} : \mathbb{R}^d \to \mathcal{P}(\{1, \dots, d\}), \quad x \mapsto s \tag{4.3}$$

with $\mathcal{P}$ being the power set, that sends input data $x \in \mathbb{R}^d$ to a set $s \subseteq \{1, \dots, d\}$ of size $|s| = k$ which corresponds to a subset of the indices of x and is called *explanation mask*. For a prediction $\mathcal{M}(x)$, we denote the *explanation* of the model $\mathcal{M}$ by $\mu(x, s)$, and it is defined as the masking of x in which only the features in s are visible as the explainer model selected them to constitute the explanation. Formally, we define the explanation as

$$\mu(x, s) := x \odot \chi(s) \tag{4.4}$$

where $\chi(s)$ is the characteristic vector of the set of indices s and $\odot$ designates an element-wise operator leaving the features of x indicated by $\chi(s)$ unchanged while hiding[4] the others. Likewise to the data, we consider $s \subseteq \{1, \dots, d\}$ as a realisation of a random variable $S = \mathcal{E}(X)$ and, further, consider the explanation $\mu(x, s)$ as a realisation of the random variable $\mu(X, S)$. Thus, we extend the definition of the probability density p to

$$p : \Omega \to \mathbb{R}, \quad p(x, y, s) = p(x, y) \cdot p(s|x) = p(x, y) \cdot d\mathbb{P}(\mathcal{E}(x) = S)/d\nu_S. \tag{4.5}$$

Please note: Firstly, we will use the indexing notation x_s to reference sub-vectors of x, e.g. we will write

$$p(x_s) = p(x_{i_1}, \dots, x_{i_k}) \quad \text{for} \quad s = \{i_1, \dots, i_k\} \tag{4.6}$$

to indicate the marginal probability of the features s of x assuming their values.

Secondly, note that the probability density of $\mu(x, s)$ satisfies

$$p(\mu(x, s)) = p(x_s)p(s|x_s) = p(x_s, s) \tag{4.7}$$

which follows immediately from its definition since $\mu(x, s)$ is fully determined by the features x_s in it and the explanation mask s defining it.

Thirdly, we note that it is necessary to limit the explanation mask size of an explainer model because, otherwise, the explanation maximising mutual information would always be the trivial explanation that does not mask any features of the input. But this is not desirable because it does not tell us which are the important features for the model's prediction and, thus, we restrict explanation masks to a certain maximal size k.

Finally, we would like to point out an important aspect of explanations which we will need on several occasions and, thus, have cast into the following observation.

[4] In Observation 1 we will see that the purpose of the explanation mask is not only to remove the masked features, but also to indicate which features are masked. To achieve this, the values used for masking features must be distinguishable from actual feature values. For example, if the data X have binary features taking values 0 or 1 with positive probability, then features should not be masked by replacing them with 0 because this would introduce an ambiguity as to which features are masked. If, on the other hand, it holds $P(X_i = 0) = 0$ for any feature X_i of the data, the gate operation can simply be implemented as an element-wise multiplication because, almost surely, an unmasked feature will not take the value 0 and, hence, it is possible to distinguish masked and unmasked features from each other.

Observation 1 (Interpretability of Explanations)**.** For the functioning of our explanation method, it is necessary that in each explanation $\mu(x, s)$ it is possible to deduce what the features s included in it are. This is because otherwise, e.g. in the case of the explanation simply being given as the sub-vector x_s with the masked features deleted, it is impossible to know which entry of x_s belongs to which feature of x, thus rendering the explanation meaningless. To understand this, consider the example of

$$x_1 = (4, 5, 6)^\top, s_1 = \mathcal{E}(x_1) = \{1, 2\} \quad \text{and} \quad x_2 = (4, 6, 5)^\top, s_2 = \mathcal{E}(x_2) = \{1, 3\}. \tag{4.8}$$

If a user is presented with merely the sub-vectors

$$x_{1_{s_1}} = (4, 5)^\top \quad \text{or} \quad x_{2_{s_2}} = (4, 5)^\top \tag{4.9}$$

without further references, it is not possible to infer which features of x are included in the explanations. However, if one is presented with masked-feature vectors

$$\mu(x_1, s_1) = (4, 5, \cdot)^\top \quad \text{or} \quad \nu(x_2, s_2) = (4, \cdot, 5)^\top, \tag{4.10}$$

one can infer from the explanations which are the features relevant for them and, therefore, one is able to correctly interpret them. In this regard, it is also noteworthy that there are equivalent options to formalise the notion of a masked vector to the one used in Definition 2. For instance, one could define $\mu(x, s)$ as the pair (x_s, s), as it is possible to determine which feature takes which value in this case. While this alternative definition is helpful to understand our upcoming discussions, nonetheless, we chose $x \odot \chi(s)$ as our main definition of explanations, as it closely reflects the implementation of our method.

4.2. User Profile

Finally, we define the part of our method that provides information to personalise explanations. To motivate this definition, let us review how the idea of a user profile was originally introduced in JN and what we want it to do.

In its original form introduced by JN, for a given input x, a user summary is a quantity defined by the user to explain the model's behaviour on x based on the user's perspective and knowledge of the model $\mathcal{M}$. More concretely, a user summary is a vector consisting of features derived from the input which the user deems important for the model's prediction. These features, however, are the same for every input x and, thus, in JN the user profile $\mathcal{U}$ can simply be seen as a feature transformation of X chosen by the user. But this means that user summaries in JN are not local in the sense that the selected features are different for different instances x of X.

In contrast to JN, we want user summaries to be local so that users do not have to come up with a set of features that is relevant for all possible inputs, but can choose different features to explain the model's behaviour on different inputs. Thus, we enable user summaries to be local by designing them, like explanations, as masked feature vectors.

Regarding the nature of the features of user summaries, we find that in the default case they can just be the same as the features of the input data X. This is a reasonable approach because it makes the explanations and user summaries perfectly complementary and easily comparable to each other. So for each input x, the user summary would be a masking of x showing features that are relevant for the model's decision according to the user, and the explanation would be a set of additional features of the input further explaining the behaviour of $\mathcal{M}$. In more advanced cases, one could also use other features derived from the input data, such as entire segments of images, the outputs of simpler models trained to solve the same problem as $\mathcal{M}$, or other hard-coded functions of x, as features of user summaries.

As a concrete example of an implementation of user profiles, consider the creation of personalised explanations for an image classification model. In this case, the user summary provided by the user for a single image will be the area of the image that the user views as most salient for the output of the model. For example, if the task of the model is to classify images of different animals, then the user might select a part of the image showing a certain body part of the animal if they deem this body part to be crucial for the model's prediction. Like this, the user profile can capture a comprehensive record of the understanding that the user has of the model to be explained and, in turn, this record can be used to effectively personalise explanations to the user.

At last, we note that it is not strictly necessary for creating local user-adaptive explanations to localise user summaries by masking. Hence, if the user features are not suitable for being masked, one can also drop this requirement again. As a rather exotic example of such a scenario, one could consider user summaries in the form of text, e.g. an explanation written by the user, which is then transformed by text embedding methods [PBGN23] to vectors to be used as user features.

Definition 3 (User Profile)**.** We define a user profile $\mathcal{U}$ as a map

$$\mathcal{U} : \mathbb{R}^d \to \mathbb{R}^{d''}, \quad x \mapsto u \tag{4.11}$$

that maps input data $x \in \mathbb{R}^d$ to a vector $u \in \mathbb{R}^{d''}$, which we call the *user summary* of x. A user summary is understood to be a realisation of the random variable $U = \mathcal{U}(X)$. Thus, we obtain conditional independences $(U \perp\!\!\!\perp Y \,|\, X)$ and $(U \perp\!\!\!\perp S \,|\, X)$ and extend the definition of p to be

$$p : \Omega \to \mathbb{R}, \quad p(x, y, s, u) = p(x, y, s) \cdot p(u|x) = p(x, y, s) \cdot d\mathbb{P}(\mathcal{U}(x) = U)/d\nu_U. \tag{4.12}$$

We note that, while it is not strictly required like for explanations, it can be advantageous for the comparability of user summaries to limit the size of their feature masks. In this case, we denote this limit by the *user summary size* l for $0 < l < d''$.

5. Mutual Information for Local Explanations

After having defined all the elements that make up our framework, we can now turn towards the objective by which our explainer model is trained.

Before making a first attempt to define the objective for training the explainer model, let us recall some insights from previous chapters: In chapter 1 we found that explanations for ML models often lack of personalisation, hindering the use of explainability by a wider range of users. To tackle this issue, we reviewed existing approaches for personalised explanations and found that JN was the only such method that can be used for general purposes. Therefore, we decided to use JN as the foundation for our work and additionally examined the existing mutual-information-based XAI method L2X. We found that both of them have pros and cons and, thus, concluded that our method should merge the approaches of L2X and JN to create explanations that are both local and personalised.

For an explainer model $\mathcal{E}$, this means it should provide the user with local explanations personalised to the user's understanding of the model $\mathcal{M}$, as expressed through the user profile $\mathcal{U}$. To accomplish this, we propose local explanations in the form of masked feature vectors, as defined in the previous chapter, and to train the explainer model we propose to adopt objective (5.1), the maximisation of the mutual information of explanations $\mu(X, S)$ and model output Y conditioned under user summaries U, as target.

$$\max_{\mathcal{E}} I(\mu(X, S), Y \mid U) \tag{5.1}$$

According to the motivations of L2X and JN, it seems a reasonable idea to use mutual information of explanations and model outputs as criterion for explanations. However, there are subtleties in the nature of mutual information which cause certain encoding phenomena to occur in *local* explanations defined by this objective, and which are not aligned with the intentions that motivate it. Describing these phenomena and how to modify objective (5.1) to resolve them will be the main concern of this chapter.

5.1. Back to the Roots – L2X Objective

Since just the combination of a mutual-information-based objective with local explanations by masked feature vectors forms a minimal reproducible example of the issues we will discuss in this chapter, we restrict our discussion of them to this simpler setting, which is just L2X. Nonetheless, the issues raised in this chapter hold in the same way for our method as the additional technique of conditioning on user summaries adopted from JN does not alter them in any way. Thus, let us begin by refreshing the knowledge of what

L2X exactly looks like. From now on, however, we will use our notation introduced in chapter 4 instead of the original notation of L2X that we used in section 3.2.

The core idea of L2X is to choose the 'most informative' subset of the features of an input data instance to serve as explanation for the model's prediction on it. This means that L2X considers, on the one hand, the model to be explained $\mathcal{M} : x \mapsto y$ and, on the other hand, an explainer model $\mathcal{E} : x \mapsto s$ which defines a masking of the input x. This explainer model $\mathcal{E}$ is trained with a reward function approximating the objective

$$\max_{\mathcal{E}} I(\mu(X, S), Y) \tag{5.2}$$

to provide local explanations for $\mathcal{M}$. Notably, the masks S are only allowed to be of a fixed maximal size k (in the sense that k features are visible) because maskings that allow more features to be visible allow more information to be conveyed. Hence, allowing for arbitrary mask sizes k would lead to the mask that does not delete anything to be the trivial but undesired solution to the explanation problem. Now, while objective (5.2) only characterises the behaviour of $\mathcal{E}$ on a global level, to demonstrate that this objective also has a meaningful interpretation at the level of individual data instances, [CSWJ18] presents the following statement, saying that L2X explanations are optimal in the sense that under their knowledge the encoding length of the model output Y is minimised.

Hypothesis 1 (Theorem 1 of [CSWJ18])**.** *Let the random variables X be the input data, Y the model output and let $\mathcal{E}^*$ be the explainer model defined by the local criterion (5.3). Then $\mathcal{E}^*$ is a solution to the global L2X criterion (5.2), and for any other explainer model $\widehat{\mathcal{E}}$ optimising criterion (5.2) it holds that $\widehat{\mathcal{E}}$ coincides with $\mathcal{E}^*$ almost surely over the data generating distribution p_X.*

$$\mathcal{E}^*(x) = \arg\max_{s} \mathbb{E}_{Y|X}\Big[\log p(Y|X_s = x_s)\Big|X = x\Big] \tag{5.3}$$

Albeit this information-theoretic approach seems intuitively appealing – it is supposed to pack as much information about the model output as possible into the explanations – taking a closer look at it reveals that it evokes some unwished-for behaviour.

5.2. Information Encoded in the Feature Mask

Designating Hypothesis 1 as hypothesis, and not as theorem, already gives a cue that not everything is alright with it. As we will see, it does not take into account a way in which information about the model output beyond the explanation features can be added to the explanation via the mask S, which we will call *output encoding*. To understand how this happens, let us contemplate from scratch what kind of explanations objective (5.2) yields.

Example 1. To get acquainted with the issue of output encoding, let us demonstrate its occurrence for a model of the XOR function. Consider data X with 2 features independently and uniformly taking values in $\{-1, 1\}$, the model $\mathcal{M} : \{-1, 1\}^2 \to \{-1, 1\}$, $\mathcal{M}(x) = x_1 x_2$, and let $Y = \mathcal{M}(X)$. Further, let the explainer model $\mathcal{E} : \mathbb{R}^d \to \{\{1\}, \{2\}\}$ be defined by $\mathcal{E}(x) = \{1\} \Leftrightarrow Y = 1$ with explanation mask size $k = 1$.

The maximally possible value of mutual information $I(\mu(X,S),Y)$ in this example is 1 bit as Y is following a uniform distribution on 2 possible values.[1] To calculate the actual mutual information $I(\mu(X,S),Y)$ of the explanations and model outputs, we consider the probability densities involved in its definition. For any $y \in \{-1,1\}$ we find $p(y) = 0.5$; for any $s \in \{\{1\},\{2\}\}$ and corresponding $x_s \in \{(-1),(1)\}$ it holds $p(x_s,s) = 0.25$; and, as S and Y are equivalent, for any values of s, x_s and y we find

$$p(x_s,s,y) = p(x_s)p(s,y|x_s) = p(x_s)p(y|x_s) = 0.5 \cdot 0.5 = 0.25. \tag{5.4}$$

Therefore,

$$I(\mu(X,S),Y) = \mathbb{E}\left[\log \frac{p(x_s,s,y)}{p(x_s,s)p(y)}\right] = \mathbb{E}\log 2 = 1 \text{ bit}, \tag{5.5}$$

and so $\mathcal{E}$ is an optimal explainer model according to criterion (5.2).

But where does the information in its explanations come from? To understand this, we have to consider both the explanation masks S and the features of X visible in the explanations. As S and Y are equivalent, we know that S alone has the maximal possible mutual information of 1 bit with Y, and from Observation 1 we know that S contributes to $I(\mu(X,S),Y)$. Regarding the features visible in the explanations, on the other hand, the fact that Y is independent from each X_i, combined with the explanation mask size being $k = 1$, tell us intuitively[2] that the features themselves do not contribute any information to $I(\mu(X,S),Y)$. So, the explanations created by $\mathcal{E}$ are not informative through the features visible in them, despite having perfect mutual information with the model outputs through an essentially arbitrary encoding in the explanation masks. This renders the explanations effectively meaningless, and it is what we call *output encoding*.

The bottom line of this example is that there is nothing that hinders the explainer model $\mathcal{E}$ from using the feature mask S to increase the mutual information of explanations and model outputs instead of just relying on the features of X for it. Even worse, information in features may be suppressed in explanations because the mask is more adept at expressing it and, thus, acts as dominant information channel. This, however, counteracts the intended purpose of L2X as its explanations should be informative through the values of the visible features, and not by some arbitrary encoding in the feature mask.

Following this first insight into the effects of objective (5.2), let us move to a setting where we can quantify the effects of output encoding and compare it to the information conveyed by the features of explanations. To this end, this example uses linear regression models, for which it is possible to calculate the mutual information of global explanations.

Example 2. In this example, we will compare the amounts of information that explanations and explanation masks can contain for simple linear regression models. Regarding the explanations, we only consider global explanations here because for these it is easily possible to calculate their mutual information with the model output in this setting. One might object that local explanations are more informative than global ones, but we see in

[1] This follows from Example 2.1.1 and Theorem 2.4.1 of [CT12].

[2] Definition 5 will provide a way to rigorously assess this.

appendix B that the additional amount of information conveyed by local explanations is typically less than 1 bit and, thus, global explanations serve as valid benchmark here.

Thus, let data X be on $\mathbb{R}^d$ with independent and standard normally distributed features $X_i \sim \mathcal{N}(0,1)$ for $i = 1, \dots, d$ and consider a linear regression model $\mathcal{M} : x \mapsto \sum_i \alpha_i x_i$ with weights $\alpha_i \in \mathbb{R}$ such that $\sum_i \alpha_i^2 = 1$. That is, the model output $Y = \mathcal{M}(X)$ is also following a standard normal distribution $Y \sim \mathcal{N}(0,1)$. As global explanations are just sub-vectors X_s of X, we will indicate instances of them here just by the respective sub-vector x_s since it is equivalent to $\mu(x, s)$.

Then, as for the mutual information of X_s and Y, a simple calculation (see Example 8.5.1 in [CT12]) shows that for any mask s it holds

$$I(X_s, Y) = -\frac{1}{2} \log \left(1 - \sum_{i \in s} \alpha_i^2 \right) \tag{5.6}$$

which tells us that the mutual information of X_s and Y scales with the logarithm of the variance of Y not explained by X_s. This fact, however, already demonstrates the weakness of using masked feature vectors for conveying information. In table 5.1 we have listed some exemplary values for the mutual information of X_s and Y and the corresponding levels of explained variance, and it demonstrates just how much explained variance is necessary for the explanations to reduce the uncertainty about the outcome of Y.[3] For example, knowing which quartiles of its distribution the outcomes of Y fall in means to gain 2 bits of information about it, which is equivalent to explaining 93.75% of the variance of Y; and it is necessary to explain more than 99.6% of the variance of Y to convey 4 bits of information about it, which is equivalent to knowing which 16-quantiles the outcomes fall in. As a matter of fact, the mutual information of X_s and Y can be arbitrarily large if the explained variance $\sum \alpha_i$ is just close enough to 1. But the explained variance has to be very close to 1, in other words, Y has to be almost completely determined by the observation of X_s, so that their mutual information attains more than a couple of bits.

In contrast to this, let us now consider how much information can be expressed in explanations by the mask S. If we consider masks of size k and have d features in total, then it is well known that there are $\binom{d}{k}$ different masks, hence the masks have a maximal capacity of $\log \binom{d}{k}$ bits of information to convey.[4] By Lemma 1 (see appendix A.1), we find the approximation

$$\log \binom{d}{k} = dH(k/d) + O(\log d) \tag{5.7}$$

where H denotes the binary entropy[5]. Through this, we can see at once that for fixed masking ratio k/d, the information capacity of S is linearly proportional to d. For fixed d and varying k, on the other hand, the maximal amount of information in the mask occurs

[3] The same result holds for multivariate $Y = AX$ for which the variance of Y not explained by X_s is given by $\det(AA^T - A_s A_s^T)$ with AA^T being the covariance matrix of Y and A_s the restriction of the parameter matrix A to the features visible in X_s. Thus, the above result is not just a consequence of Y being univariate, but similarly holds in more general cases.

[4] This results from Theorem 2.6.4 and Equation 8.54 of [CT12].

[5] By Example 2.1.1 of [CT12] it holds $H(p) = -p \log p - (1-p) \log(1-p)$.

with about d bits at $k \approx d/2$ and is slowly falling off towards $\log(d)$ bits at the extremes of $k = 1$ and $k = d - 1$. Table 5.2 provides exact values of $\log \binom{d}{k}$ for some selected values of d and k/d, and one can see there clearly that, except for the smallest choices for d and k, the masks have the capacity to carry information in the range of well above 4 bits.

We emphasise this threshold here because it shows that one can generally expect the number of features d to be large enough so that, for any appropriate mask size k, the mask itself can convey an amount of information which can be overcome by the features visible through the mask only if they almost completely determine the model output.[6] This, however, crucially depends on the explanation mask size being large enough so that enough features are visible through the mask to sufficiently determine the outcome of Y. Because, as we have seen in table 5.1, even the tiniest remaining uncertainty in the outcome of Y is enough to vastly diminish the information contained in this channel and allow it to be many times outnumbered by output encoding in the mask.

With these observations we are now ready to pin down the nature of output encoding. Further, as these observations contradict Hypothesis 1 of [CSWJ18], which does not admit any possibility for output encoding to occur, let us cast them into an opposing hypothesis.

Definition 4 (Output Encoding)**.** By *output encoding* we denote the neglect of information contained in the features of input data X in favour of information encoded in the explanation mask S by an explainer model $\mathcal{E}$.[7]

Hypothesis 2. *Depending on the explanation mask size and the amount of information in the features of explanations, L2X explanations are susceptible to output encoding.*

Having stated Hypothesis 2, we now have to show that it holds. The roadmap for going about this is as follows: In section 5.5 we will provide experimental evidence based on L2X applied to linear regression models which shows clear signs of output encoding happening under certain conditions and, thus, is sufficient to empirically validate Hypothesis 2 and reject Hypothesis 1. Furthermore, we will provide criteria characterising the occurrence of output encoding, which we derive from our observations, and in section 5.6 we will evaluate some approaches to mitigate output encoding in L2X. As the analysis of these experiments requires a deeper understanding of the information-theoretical aspects of the features and masks of local explanations, though, first, we will develop this theory in section 5.3. In the course of this analysis, we will also, already before disproving it, see how to fix Hypothesis 1 to make it valid again.

[6] To make this comparison between values of mutual information involving both discrete and continuous random variables, we note that mutual information is well-defined in all such cases. This is because mutual information can be defined generally for any combination of random variables as the supremum over the values of mutual information of all discretisations of them. Therefore, having $I(S, Y) = I(X_s, Y) = n$ bits means both S and X_s convey an amount of information about Y equivalent to a n-bit quantisation of it. For a more thorough discussion of this we refer to section 8.5 of [CT12] and the 'master definition' of mutual information given there, and a rigorous proof of these ideas can be found in Theorem 4.5 and Remark 4.3 of [PW25].

[7] The result of Proposition 1 will allow us to characterise output encoding more formally, and the remarks following it with Corollary 1 will establish this connection.

$I(X_s, Y)$ in bits	$\sum_{i \in s} \alpha_i^2$
0.5	0.5
1	0.75
2	0.9375
4	≈ 0.9961
8	≈ 0.999985
16	≈ 0.99999999977
∞	1

Table 5.1.: Some exemplary values for the mutual information $I(X_s, Y)$ in bits considered in Example 2 and their corresponding values of variance of Y explained by X_s according to equation (5.6).

k/d \ d	8	16	32	64	128	256	512
1/2	6.13	13.65	29.16	60.67	124.17	251.67	507.17
1/4	4.81	10.83	23.33	48.80	100.22	203.57	410.76
1/8	3	6.91	15.13	32.04	66.34	135.42	274.07
1/16	–	4	8.95	19.28	40.38	83.06	168.91
1/32	–	–	5	10.98	23.35	48.54	99.41
1/64	–	–	–	6	12.99	27.38	56.62

Table 5.2.: Values of the mask information capacity $\log \binom{d}{k}$ in bits rounded to at most 2 decimal places for some exemplary values of number of features d and relative mask size k/d.

Regarding existing literature on output encoding in mutual-information-based explanations, the situation appears to be very sparse.[8] To the best of our knowledge, the line of work [JSAR21] and [PNR24] are the only papers touching upon this topic. Among these, [JSAR21] proposes a method to mitigate encoding, which we will evaluate in section 5.6, and a method to detect output encoding in explanations. This work is picked up by [PNR24], which proposes a modification of the encoding detection method of [JSAR21] and claims to improve its results. While [JSAR21] does not engage any probabilistic methods in its exposition, [PNR24] uses a definition of output encoding equivalent to what we will find in Corollary 1. However, neither of these papers offers the same theoretical insight on the origin of output encoding in L2X as our work and, thus, they do not consider varying explanation mask sizes as a way to affect output encoding in explanations. In particular, to the best of our knowledge, there is no published literature examining the theoretical gaps in the L2X framework which we will discuss in the following section.

5.3. Ignoring the Mask

Having discussed the potential effect of output encoding induced by the mutual-information-based objective (5.2), we have to ask how this fits together with the local interpretation of it that Hypothesis 1 poses and [CSWJ18] claims to prove? After all, the point of Hypothesis 1 is that the explanation provided for any input instance x is determined solely by the conditional probability distributions $p(y|x_s)$, and not by arbitrary encodings of Y in the mask S that are unrelated to the observed data x.

The answer to the question why output encoding occurs under objective (5.2) but not under the allegedly equivalent objective (5.3) is actually very simple and reveals itself once one takes a close look at the proof of Hypothesis 1 presented in [CSWJ18]. The mistake, which occurs there in the step between equations (1) and (2), is shown in equation (5.8) and consists of the authors wrongly equating the probability density $p(x_s)$ of the features visible through the mask with the probability density $p(\mu(x,s))$ of the explanation, which rather is equal to $p(x_s, s)$ as we have seen in equation (4.7).

$$p(x_s) \overset{\not}{=} p(\mu(x,s)) = p(x_s, s) \tag{5.8}$$

Consequently, the attempted proof of Hypothesis 1 by [CSWJ18] ignores the fact that explanations $\mu(X,S)$ are not independent of the mask S itself, which, in turn, is precisely what allows output encoding to occur.

Nonetheless, Hypothesis 1 is not a lost cause as its proof still works if one adjusts the mutual-information-based objective (5.2) accordingly. We will now provide a quick analysis of the decisive parts of the proof in question and introduce the ideas necessary to fix it. The corrected statement adapted for our method will then be presented and proven as Theorem 1 in section 6.1.

[8] One can also contemplate if output encoding was ever not expected to happen – autoencoders with architectures similar to L2X are well-known to learn sparse representations, i.e. encodings, of data.

To get started, let us have a look at the core idea of this proof. It is that the mutual information of explanations $\mu(X, S)$ and predictions Y can be expressed as

$$I(\mu(X,S),Y) = \mathbb{E}_{S,Y,X_S} \log \frac{p(X_S,S,Y)}{p(X_S,S)p(Y)} = \mathbb{E}_{S,Y,X_S} \log \frac{p(Y \mid X_S,S)}{p(Y)}. \tag{5.9}$$

Here we see where the misalignment (5.8) in Hypothesis 1 comes into play as the term $p(Y \mid X_S, S)$ in the right side is demanded by the mutual-information-based objective (5.2), while objective (5.3) of Hypothesis 1 requires the term $p(Y \mid X_S)$ to appear here. The way to fix this, metaphorically speaking, is to get rid of the influence that only the mask induces in the probability $p(\mu(X, S))$ by replacing the full explanation $\mu(X, S)$ with something that represents only the information entailed by the features visible in them.

To deduce this quantity, let us reason about it from first principles. For a fixed sub-vector X_s of X, the mutual information of it with Y is given as

$$I(X_s,Y) = \mathbb{E}_{X,Y}\left[\log \frac{p(X_s,Y)}{p(X_s)p(Y)}\right]. \tag{5.10}$$

Similarly to this case, we want to measure the mutual information that the features X_S of explanations have with Y. To do so, we have to consider that the features of explanations are not fixed subvectors of the data X, but the selected features change with every instance of X instead. Therefore, we modify the expectation in (5.10) so that it takes account of the explanation mask used at each data instance.

Definition 5 (Feature Mutual Information of Explanations)**.** Consider random variables X denoting input data, S denoting explanation masks, and Y denoting the model output. We define the *feature mutual information* $I(X_S, Y)$ of the visible features X_S in the explanations $\mu(X, S)$ with the model output Y as

$$I(X_S,Y) = \mathbb{E}_{X,Y}\left[\sum_s p(s|X) \log \frac{p(X_s,Y)}{p(X_s)p(Y)}\right] = \tag{5.11}$$

$$= \mathbb{E}_{X,Y}\left[\mathbb{E}_{S|X}\left[\log \frac{p(X_S,Y)}{p(X_S)p(Y)}\right]\right] = \mathbb{E}_{X,S,Y}\left[\log \frac{p(X_S,Y)}{p(X_S)p(Y)}\right]. \tag{5.12}$$

With this notion of mutual information involving just the features of explanations, we have obtained the tool allowing us to ignore the information contributed by the masks in local explanations, and thus fix Hypothesis 1. To see how this happens, we just need the following result, the proof of which can be found in appendix A.2.

Proposition 1. *Let the random variables X denote input data, S denote explanation masks, and Y denote the model output. Then, for the mutual information $I(\mu(X, S), Y)$ of explanations $\mu(X, S)$ and model outputs Y we have the partition*

$$I(\mu(X,S),Y) = I(X_S,Y) + I(S,Y \mid X_S). \tag{5.13}$$

According to Proposition 1 the mutual information $I(\mu(X,S),Y)$ of local explanations with model outputs is composed of the following two disjoint parts, an illustration of which is given by the Venn diagram in Figure 5.1.

(1) The feature mutual information $I(X_S,Y)$ of explanations, being the part contributed by the visible features X_S and which we need to make our explanations informative.

(2) The conditional mutual information $I(S,Y \mid X_S)$ of the masks S and output Y under the visible features X_S, which is precisely what constitutes output encoding.

Which leads to the following insights:

Corollary 1. *Output encoding in explanations occurs if and only if* $I(S,Y|X_S) > 0$.

As the original formulation of Hypothesis 1 in [CSWJ18] seeks to maximise $I(\mu(X,S),Y)$, it also allows $I(S,Y|X_S)$ to grow. In the better case, this might allow some information about model outputs to be stored in the masks without harming the information conveyed by the features of explanations. But in the worse case, information in the features can be suppressed by the mask since it is much more expressive than the features, as we have seen in Example 2. Thus, as it captures precisely the information we want to include in explanations, we see that replacing $I(\mu(X,S),Y)$ in Hypothesis 1 by feature mutual information $I(X_S,Y)$ is the correct training objective to get rid of output encoding.

5.4. Evaluation Methods

In section 5.2, we developed the idea that output encoding in L2X explanations may happen depending on the size of the explanation mask and the information entailed by the features visible through the mask. This is Hypothesis 2, which we postulated there, and validating which is the goal of the experiments described in section 5.5. There, we will not only confirm output encoding occurring in L2X explanations, thereby disproving Hypothesis 1 by [CSWJ18], but we will also derive empirical rules characterising for which mask sizes output encoding occurs and evaluate approaches to mitigate it altogether.

To find out under which conditions output encoding occurs, we consider L2X explanations for linear regression models and see how metrics characterising output encoding vary with changing explanation mask sizes. To this end, we restrict the setup of Example 2 to the 16-dimensional case and consider data X on $\mathbb{R}^{16}$ with independent and standard normally distributed features $X_i \sim \mathcal{N}(0,1)$ for $i = 1,\dots,16$ and consider linear regression models $\mathcal{M} : x \mapsto \sum_i \alpha_i x_i$ with weights $\alpha_i \in \mathbb{R}$ such that $\sum_i \alpha_i^2 = 1$. That is, the model output $Y = \mathcal{M}(X)$ is also following a standard normal distribution $Y \sim \mathcal{N}(0,1)$. Without loss of generality, we assume model weights are sorted in descending order by absolute values, i.e. α_1 is always the most and α_{16} the least influential weight. The only difference between the models which we consider in our experiments is in the distribution of their weights. We examine 6 interesting example cases which we introduce along with the observed results. A summary of the used models is provided in appendix C.3.

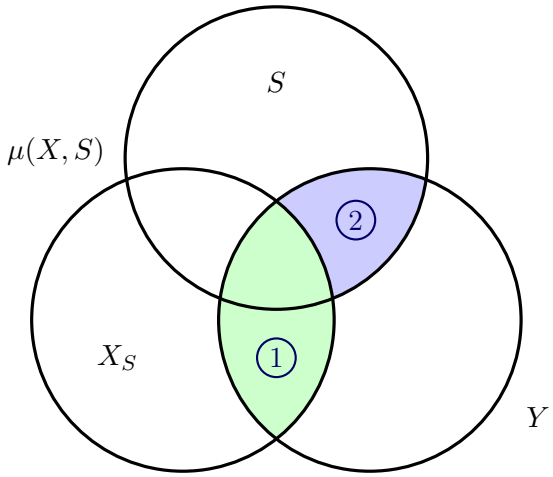


Figure 5.1.: An illustration of how the mutual information of the pair of random variables $(\mu(X, S), Y)$ describing the explanations and model outputs is shared among the components contributing to it. The green and blue shaded areas combined correspond to the mutual information $I(\mu(X, S), Y)$ of explanations $\mu(X, S)$ and model outputs Y; whereas the green shaded area alone corresponds to (1), the feature mutual information $I(X_S, Y)$ of the visible features X_S with the model outputs Y; and the blue shaded area alone corresponds to (2), the conditional mutual information $I(S, Y|X_S)$ of the masks S and outputs Y under X_S.

Regarding the explanations themselves, we consider two kinds of them. On the one hand, we employ the default L2X explanation method, where explanations are provided by a neural network explainer model trained jointly with another neural network, acting as a regressor[9] from explanations to the model output, to maximise the mutual information of explanations and model outputs. On the other hand, we make use of 16-dimensional linear regression models being simple enough so that the implementation proposed by L2X is not necessary to obtain mutual-information-based explanations of them. Instead, one can compute explanations of them by directly solving[10] criterion (5.3). As we will prove in Theorem 1, this is equivalent to maximising the feature mutual information of explanations. Thus, these explanations are provably free of output encoding and serve as baseline against which to compare default L2X. We will refer to explanations obtained by directly solving criterion (5.3) as *exact explanations*. The full details about implementation and training of explainer models used here can be found in appendix C.1.

To detect output encoding in explanations, we keep track of 3 metrics across explanation mask sizes. These are 1.) the feature mutual information of explanations, and 2.) the *output prediction losses* of explanations on a) full explanations, and b) explanation masks.

The first quantity we observe is the feature mutual information of explanations with model outputs. Its intended purpose is to measure how information conveyed by exact explanations compares to the information capacity of explanation masks, as we presumed this an indicator of output encoding, but this conjecture did not manifest itself. Therefore, we mainly use feature mutual information for its secondary purpose of measuring how well L2X explanations fulfil their objective compared to explanations solving it exactly. The method we use to measure feature mutual information is discussed in appendix C.1.

The other two properties of explanations which we will measure are what we call *output prediction losses* of full explanations and explanation masks only. These metrics are motivated by the mechanism through which output encoding is enabled in L2X and exploit it to reveal the presence of output encoding instead.

As outlined before, L2X explainer models are trained jointly with regressor models, which attempt to predict the actual model outputs Y from explanations $\mu(X, S)$. Thus, the assumed presence of output encoding in L2X explanations suggests that the corresponding regressor model can use this encoding to predict the model output. Otherwise, there would be no reason for output encoding to persist throughout the training of the explainer model. Therefore, our premise is that regressor models used in training L2X explainer models are capable of exploiting output encoding to predict model outputs, and we want to make use of this capability to detect it. To this end we take explanations $\mu(X, S)$ and use them to train two new regressor models to predict model outputs Y, but just one regressor is provided with full explanations $\mu(X, S)$ while the other one is given only explanation masks S as input data for this task.

[9] We introduce the notion of *regressor* models here as [CSWJ18] does not use a specific name for them.

[10] The feasibility of mathematically solving criterion (5.3) is due to linear regression models allowing to easily describe conditional densities $p(y|x_s)$ for all feature subsets s. The practical feasibility of solving this criterion, however, is due to the low dimension of the models, permitting searches over all feature subsets, and the data following normal distributions, which are easy to describe mathematically.

The metrics we record are test losses measuring the performance of these regressor models in predicting model outputs after their training. The idea behind this is that, for linear regression models, just knowing the explanation masks does not provide any information about the model output, as any feature value can be positive or negative and, thus, influence the output either way. Hence, it is not possible to predict the output from just the explanation masks unless there is output encoding present in them. For full explanations, on the other hand, it should be very well possible to reconstruct the model output from them as, by definition, they are maximally informative about it. In this way, we have two models at hand, both of which can make use of output encoding, but only one of them can also work without it. Hence, by comparing the two output prediction losses for each explanation type and size, we are able to assess the quality of the explanations and whether there is output encoding occurring in them or not.

5.5. Output Encoding Experiments

The results of our experiments can be seen in Figure 5.2, the 6 subplots of which each correspond to one examined linear regression model. For each of these examples, we will discuss its motivation, observations in the results, and conclusions to draw from them.

Model 1 Our initial attempt at probing output encoding is directly motivated by the observations we made in Example 2, where we demonstrated the information capacity of explanation masks to be generally much higher than the feature mutual information conveyed by explanations. Thus, we hypothesise that the proportion of these two quantities affects the occurrence of output encoding. To test this, we need to construct a linear regression model in which this proportion changes significantly with the explanation mask size. To do this, note that we have shown in appendix B that feature mutual information of local explanations is closely bounded to mutual information of global explanations. As given by equation (5.6), we see that mutual information of global explanations can be made arbitrarily large by shrinking the sum of squared weights outside the explanation mask. Thus, to observe high feature mutual information of explanations, we need to examine a linear regression model with several very small weights, and so we furnished our first examined model, Model 1, with a weight pattern commonly associated with ridge regression models. More precisely, we let the weights of Model 1, as seen in the top plot in Figure 5.2a, consist of a decaying tail of 10 weights 2^{-i} for $i = 8, \dots, 17$ and let the remaining 6 weights $\approx \sqrt{1/6}$ be equally large to make the squared weights sum up to 1.

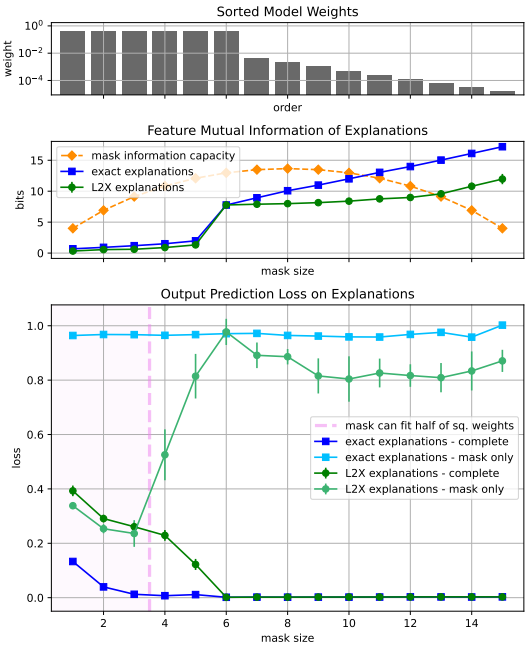
In the results of our trials with Model 1, displayed in Figure 5.2a, there are several interesting things to observe. Firstly, there is the almost perfect quality of exact explanations as exhibited by their output prediction losses. While their mask-only output prediction loss is almost constant at the level of the output variance, indicating a complete absence of output encoding, their full explanation output prediction loss deviates from 0 only for the smallest explanation mask sizes. Thus, it shows that exact explanations are very good at determining the model output with a high degree of certainty even for small explanation masks. Secondly, we observe L2X explanations to be much less effective. While their

full-explanation output prediction loss is much higher for small mask sizes and their feature mutual information lags behind that of exact explanations for larger mask sizes, their mask-only output prediction loss reveals a very strong presence of output encoding. For small explanation mask sizes the mask-only loss is in the same range as the loss incurred from full explanations, thus indicating that the entire information in these explanations is solely due to output encoding. Particularly, we would like to highlight the following distinguishable phases in the observed output prediction losses for L2X explanations: As mentioned before, at first we observe very strong output encoding for mask sizes $k \leq 3$, where the masks carry all relevant information in the L2X explanations. After that, output encoding vanishes quickly until at mask size $k = 6$ L2X explanations achieve the same performance as exact ones. For even larger mask sizes $k \geq 7$ we see again a slight decrease in mask-only output prediction losses while full-explanation output prediction losses stay perfectly flat. These phases are important for our further experiments as we will encounter similar effects in all of them. Hence, these effects demonstrate already the criteria for the occurrence of output encoding, which we will derive from them. As final observation in the results of Model 1, we remark that our initial hypothesis about output encoding being mediated by the proportion of mask information capacity and feature mutual information of exact explanations did not materialise in any of our observations. To asses this we added the mask information capacity to the feature mutual information plot in Figure 5.2a. There, one can see no significant change in output encoding between the intervals of mask size $k \leq 10$ and $k \geq 11$ defined by the dominance of either mask information capacity or feature mutual information of exact explanations.

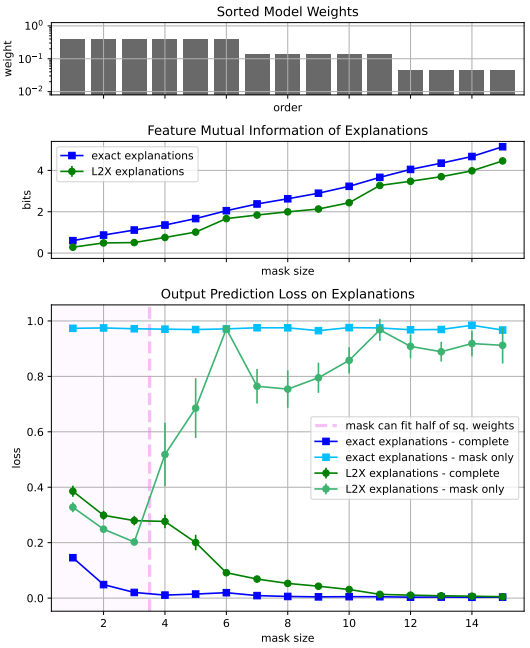
We draw 2 immediate conclusions from the experiments on Model 1. On the one hand, they show that output encoding does in fact occur, but on the other hand, it does not happen in the way we hypothesised. This, however, is favourable because it frees us from the constraints imposed on the weights of Model 1 by our initial hypothesis and allows us to examine more general models instead while still being able to study output encoding.

Model 2 The next model we examine, Model 2, is a modification of Model 1 to take advantage of the removed requirement of having several very small weights. Instead, we take a closer look at the complete absence of output encoding at explanation mask size $k = 6$ for Model 1. As this is the exact mask size where there is a large gap in the list of descendingly sorted weights of Model 1, we conjecture these two facts to be related. To check this, we let Model 2 have 2 such weight gaps to see if they are indeed causing the absence of output encoding. We let Model 2 have 3 sets of equal weights such that the 6 largest squared weights sum up to 0.9 and the 11 largest squared weights sum up to 0.99, meaning that it has 6 weights $\sqrt{0.15}$, further 5 weights $\sqrt{0.018}$ and 5 weights $\sqrt{0.002}$.

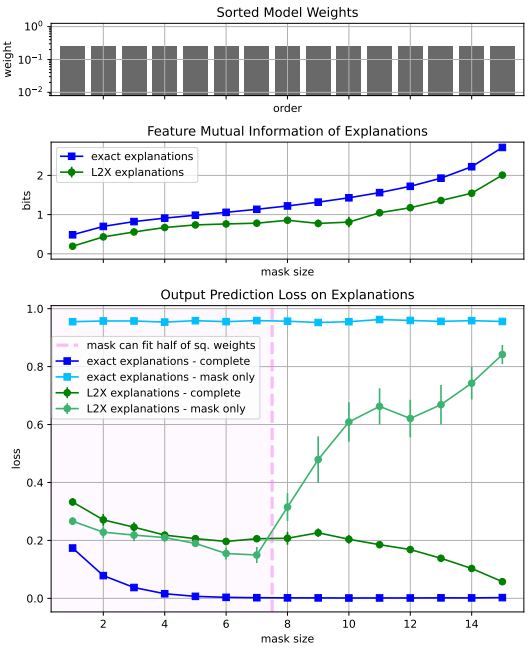
The results of the experiments on Model 2 are shown in Figure 5.2b, and they confirm our expectations for them. While we again see an almost perfect performance of exact explanations, L2X explanations show the same phased behaviour as for Model 1. In particular, we observe very strong output encoding in L2X explanations for mask sizes $k \leq 3$. But, most importantly, the mask-only output prediction losses indicate a complete lack of output encoding for mask sizes $k = 6$ and $k = 11$ — the locations of the weight



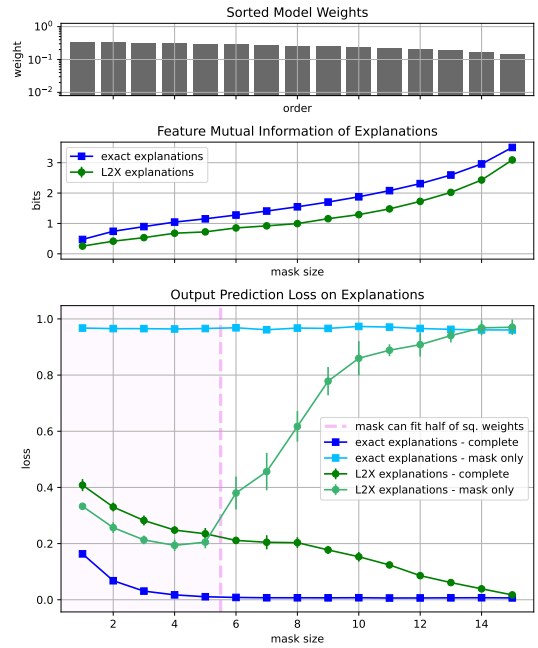
(a) Model 1



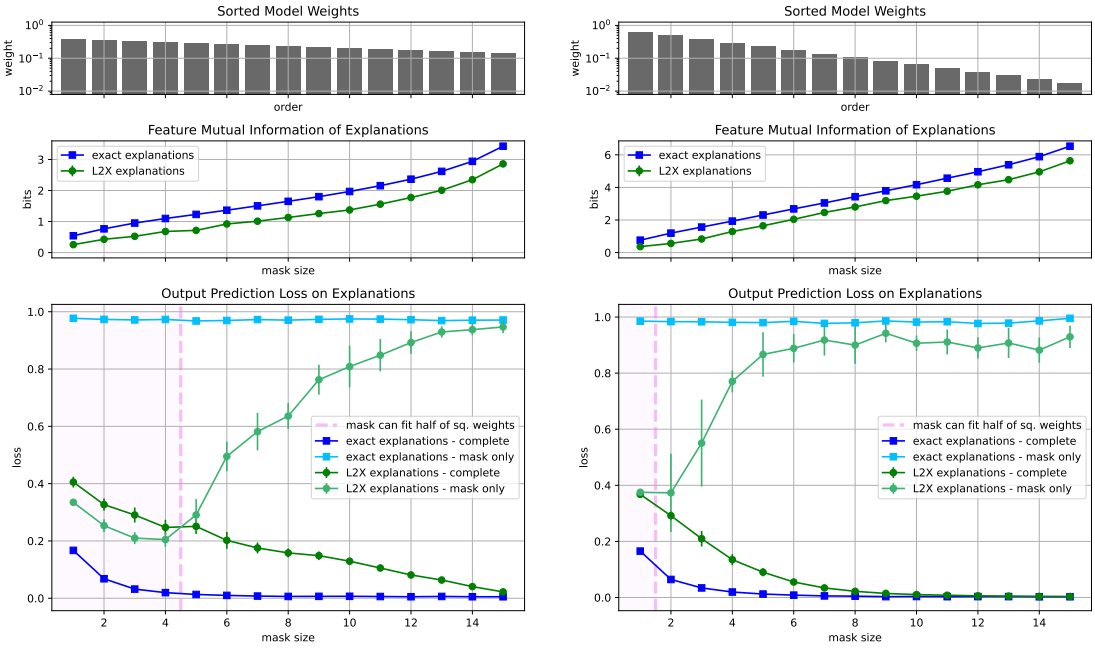
(b) Model 2



(c) Model 3



(d) Model 4



(e) Model 5

(f) Model 6

Figure 5.2.: Plots showing the results of the experiments discussed in Section 5.5. Each of the subfigures corresponds to one linear regression model, the sorted weights of which are shown in the top subplots. The middle and bottom subplots for each model show the feature mutual information and output prediction losses plotted against explanation mask sizes for exact and L2X explanations, respectively. In the graphs belonging to L2X explanations, the vertical error bars show the standard deviation of the results observed in 10 randomly initialised and trained explainer models. Additionally, for Model 1 we also show the information capacity of the explanation masks, and for all models we highlight the explanation mask sizes at which criterion (5.14) is satisfied, as we find this area to be important for the occurrence of output encoding in L2X explanations. As we ignore the trivial explanation mask sizes 0 and 16, the x-axis ranges only from 1 to 15. Consequently, the top subplots only show 15 of the 16 weights of the models. The remaining weights are determined by the displayed ones, though, as the squared weights sum to 1 for each model.

gaps. Furthermore, we also see a slight decrease in mask-only output prediction losses for mask sizes in between and above the weight gap locations. Interestingly, the feature mutual information of L2X explanations shows the same undulating pattern, whereas the exact explanations demonstrate a more consistent increase in feature mutual information.

We can view these results as an elementary empirical confirmation of our conjecture that gaps in the weight distribution of models can cause an absence of output encoding for certain explanation mask sizes. As an explanation for this effect, we hypothesise that such weight gaps, if the explanation mask size matches the number of larger weights, make explanations consisting of the features belonging to the larger weights more favourable to the explainer models than alternative explanations. This is because in linear regression models there is a direct correspondence between weights and feature relevance. Thus, explanations consisting of the features belonging to the larger weights are likely to be much better at determining the model output than alternative explanations, as these necessarily include less relevant features. Hence, weight gaps present explainer models with one distinctly relevant explanation option and, consequently, explainer models learn to almost only use this singular explanation. In turn, this causes the explanation masks to be void of any encoded information. Nonetheless, Model 2 can only be a first step in investigating this effect, and we should be cautious to interpret our observations as further research in this area is necessary.

Model 3 Next, we turn to more general linear regression models to study output encoding also in models without the weight gaps found in Models 1 and 2. As starting point for this, we let Model 3 be the linear regression model with all weights having the value of 0.25 because this model forms a straightforward baseline for all linear regression models.

Running our experiments on Model 3 yields the results shown in Figure 5.2c. There, we again observe almost perfect results by the exact explanations. Noticeably, though, the feature mutual information of exact explanations only amounts to at most 2.5 bits here, which is caused by the absence of very small weights in Model 3. Meanwhile, the performance of L2X explanations appears again in a phased behaviour similar to the previous models. The main differences being that there is no explanation mask size for which we observe a complete absence of output encoding and the phase of very strong output encoding now lasting until mask size $k = 7$. Furthermore, contrary to the previous models, full-explanation output prediction losses of L2X explanations tend towards 0 very slowly without actually reaching it, and around mask size $k = 9$, there appears to be an interval of decreased explanation quality. While the weaker decrease of full-explanation output prediction losses can be explained by the absence of very small weights, because this causes the features to be less relevant, the increased loss around mask size $k = 9$ does not admit an obvious explanation. Even more so, the worse explanation performance at mask size $k = 9$ is also reflected in the feature mutual information of L2X explanations, as explanations of this mask size have a lower feature mutual information than those of mask size 8. This is especially notable as it is the only case of a decrease of feature mutual information with increasing explanation mask size, which we observe across all our examined models, as feature mutual information is otherwise strictly increasing.

The most important observation in these results is clearly that the phase of very strong output encoding lasts until mask size $k = 7$ for Model 3, in contrast to $k = 3$ for the previous models. Consequently, we hypothesise the duration of this phase to be correlated to a threshold of squared weights able to fit inside the explanation masks. More precisely, we note that for all 3 models examined so far, the phase of very strong output encoding ranges exactly over all explanation mask sizes k for which the condition

$$\max_{|s|=k} \sum_{i \in s} \alpha_i^2 < 0.5 \tag{5.14}$$

is satisfied. Therefore, we hypothesise that condition (5.14) can characterise the occurrence of very strong output encoding in our experiments.

Regarding the worse quality of L2X explanations around mask size $k = 9$, we suspect it is related to the end of very strong output encoding at $k = 7$. At this point between output encoding and feature information, both these influences on the explanations could interfere with each other and cause them to be less informative. However, this is very speculative and further research is necessary to understand this phenomenon.

Further Models Finally, we investigate 3 more models, but as none of the experiments conducted on these models result in previously unobserved effects and only confirm the hypotheses we made so far, we present them jointly now.

For Model 4 we assume the squared weights to be uniformly distributed on an interval such that the largest one is 8 times the size of the smallest. With the constraint of the squared weights having sum 1, this results in the weights of Model 4 being $\sqrt{1 + 7i/15}/6\sqrt{2}$ for $i = 0, \ldots, 15$. For Models 5 and 6, we assume the squared weights to follow a log-uniform distribution as this allows the weights to cover a greater range of values. We define the squared weights of Model 5 such that the largest squared weight is 8 times the size of the smallest, which yields weights $2^{\beta - i/10}$ for $i = 0, \ldots, 15$ with normalisation factor $\beta \approx -1.391665$. The same holds for Model 6 where we let the largest squared weight be 2048 times the smallest, leaving us with weights $2^{\beta - 11i/30}$ for $i = 0, \ldots, 15$ with $\beta \approx -0.663485$. The factors 8 and 2048 determining the weights of Models 5 and 6 were selected because the results observed from them are representative of results observed in experiments with more general models.

The results obtained from these 3 models can be seen in Figures 5.2d, 5.2e, and 5.2f respectively and reflect what we have seen for previous models. The exact explanations demonstrate almost perfect quality with no information encoded in the explanation masks, and positive full-explanation output prediction losses occurring only for the smallest explanation mask sizes. The L2X explanations, however, display the behaviour familiar from our previous experiments. An initial phase of very strong output encoding is followed by an increase in mask-only output prediction losses and a, depending on the model, more or less quick decrease of full-explanation output prediction losses. Most importantly, we see in all 3 models that the initial phase of very strong output encoding lasts exactly until the mask size predicted by criterion (5.14), thereby conforming with our hypothesis of criterion (5.14) determining the occurrence of this phase. Furthermore, we have 2 remarks

about Models 4 and 5. On the one hand, one can see a plateau in the full-explanation output prediction losses at mask sizes $6 \leq k \leq 8$ and $k = 5$ respectively, which one might interpret as a reappearance of the transitory loss of explanation quality seen in Model 3. On the other hand, we note that mask-only output prediction losses of L2X explanations for Models 4 and 5 are special as they reach very high levels unlike other models, indicating very little information encoded in the masks for mask sizes $k \geq 13$.

The most important conclusion we can draw from these observations is that they further strengthen our hypothesis of criterion (5.14) being able to accurately predict the initial phase of very strong output encoding, as this is the case for all 3 models. Apart from this, we note that the L2X explanations of mask size $k \geq 13$ in Models 4 and 5 are the second case in which we observe L2X explanations to have so little information encoded in the explanation masks besides the weight gaps discussed for Models 1 and 2. The possible cause of this is unclear, though, and remains to be examined more closely.

Limitations

The experiments presented here serve as a proof-of-concept of the issues we raised about L2X explanations, and they can be extended in several ways:

1. The experiments have a limited scope, only covering linear regression models. This is because the simplicity of linear regression models permits to work with exact explanations and to measure feature mutual information of explanations. For further research with more general models one will only have access to output prediction losses of L2X explanations and one will not be able to compare them to exact explanations, making these experiments more difficult to interpret.

2. We only consider the quality of explanations from the global perspective, feature mutual information and output prediction losses, while the local point of view could provide more insights into output encoding. In future work, one could examine, for both exact and L2X explanations, correlations among features included in explanations and model inputs to uncover aspects hidden from the global perspective.

3. The experiments only involve few models, which, although carefully selected to show the most interesting phenomena, can only provide limited insights. Through a more systematic approach to examine the full generality of linear regression models one can hope to quantify which weight gaps imply an absence of output encoding, and to gain more insights on the loss of explanation quality in Model 3, and the particularly high mask-only output prediction losses seen in Models 4 and 5.

4. Further, all data and models used in our experiments use 64-bit floating-point precision. While this is not common for ML models, especially for neural networks, it was necessary as the initial objective of the experiments (cf. Model 1) was to study linear regression models with weights small enough to be not well representable with 32-bit precision. However, as all meaningful observations made manifested at weight ranges well representable with 32-bit floating point numbers, we are very confident that the obtained results would be equally observable in 32-bit implementations.

Discussion

In summary, we have seen in our experiments across all 6 models presented here very consistent results for both types of explanations that we considered. The exact explanations, maximising the feature mutual information of explanations, continuously demonstrated to be highly informative and free of output encoding. Meanwhile, throughout all of our experiments, the L2X explanations showed phases of very strong output encoding and generally worse explanation quality than the exact explanations. This empirical evidence, although elementary, is substantial enough so that we can draw important conclusions about Hypotheses 1 and 2 from it.

Hypothesis 1, which in the scope of our experiments alleges that L2X explanations are equivalent to exact explanations, can be flatly rejected based on our observations. It is not only the case that L2X explanations are of significantly worse quality than exact explanations, which might just indicate a flawed training procedure of the explainer models. But instead, we observed strong and consistent contrasts in the behaviours of explanations, where exact explanations were continuously highly informative while L2X explanations performed poorly and showed various output encoding effects. In particular, this applies to the prominent occurrences of output encoding observed in L2X explanations, which contradict the absence of output encoding implied by Hypothesis 1 and demonstrated by the exact explanations. Therefore, we can confidently conclude that exact and L2X explanations are not following equivalent objectives. Furthermore, this confirms the theoretical deliberations we made in section 5.3, where we pointed out the flaw in the attempted proof of Hypothesis 1 by [CSWJ18] and derived the notion of feature mutual information to obtain explanations free of output encoding instead.

Hypothesis 2, which claims that L2X explanations are susceptible to output encoding, is empirically strongly confirmed within the scope of our experiments by the observation of the phases of output encoding made for all considered models. Additionally, based on our observations, we derived criterion (5.14), which precisely characterises the explanation mask sizes for which output encoding occurs for linear regression models and, thus, further specifies Hypothesis 2 in this context.

While the insights gained from our experiments form a substantial improvement in our understanding of output encoding already, 2 subsequent questions arise immediately from them. On the one hand, among the other observed encoding effects, there is the absence of output encoding related to weight gaps seen in Models 1 and 2. This is not accounted for in Hypothesis 2 and, thus, shows that our understanding of the occurrence of output encoding is not complete yet. On the other hand, while criterion (5.14) and the existence of weight gaps are straightforward to assess for linear regression models, there is the problem of translating these criteria to more general models. This is because the notions of squared weights, feature importance, and explained output variance are equated by the simplicity of linear regression models for explanations of them, and, hence, it is not clear how to substitute them in the context of more general ML models. Potentially, other XAI methods could be suitable for this, but this remains an open question for now.

5.6. Mitigation of Output Encoding

Having gained a fundamental understanding of output encoding, at least for linear regression models, the next logical step is to use this understanding to prevent output encoding in L2X explanations altogether. To this end, we conceived and implemented two modifications of the L2X algorithm and evaluated them by re-running our experiments with these modified explanations. Further details regarding the implementations are provided in appendix C.2.

The first attempt, called *passive mitigation of output encoding*, is motivated by the observation that the joint training of explainer and regressor models in L2X allows for the exchange of information encoded in the explanation masks between them. Thus, instead of training them jointly, the idea behind this method is to first train the regressor model with a prepended dropout layer so that it is exposed to all possible explanation masks equally, and then freeze the weights of the regressor model before training the explainer model. In this way, the flow of information about the occurrence of explanation masks between the models should be stopped, which should cause the explainer model to focus exclusively on the features of explanations. Incidentally, this method is equivalent to the method proposed by [JSAR21] to prevent output encoding in L2X explanations.

The results of this attempt are displayed in Figure C.3 and are disappointing as they show no significant reduction of output encoding, but it even appears to be more pronounced in some cases. The reason for this is unclear, although we suspect the fact that models can always recognise masks as part of the explanations to be related to it.

As this fundamental fact cannot be changed, our second attempt, called *active mitigation of output encoding*, uses more profound measures to mitigate output encoding. The idea behind it is that the mask-only output prediction loss used in our experiments to detect output encoding can also be used as a loss function to influence the training of explainer models. Thus, additionally to the changes introduced by passive mitigation of output encoding, we introduce a second regressor model, called *detector* model, in the training procedure of the explainer model. The detector model shall interact with the explainer model through the explanation masks in a fashion similar to GAN models [GPAM+14] to actively prevent information about the model outputs to be encoded in the masks.

The results of this attempt, displayed in Figure C.4, are not satisfying as well. While the mask-only output prediction losses of these explanations generally indicate less information encoded in the masks, which means that the mechanism is working, the training is extremely unstable and difficult to tune. Most significantly, however, the feature mutual information and full-explanation output prediction losses of these explanations show that the GAN-like mechanism is interfering with the explanation features and making the explanations less informative, counteracting their purpose.

6. User-Adaptive Explanations

Having identified the problem of output encoding in local mutual-information-based explanations, and having theoretically resolved its origin in the objective of L2X, we can now finish what we set out to do at the beginning of chapter 5. As discussed there, we want to combine the advantages of the XAI frameworks JN and L2X to create local user-adaptive explanations to make AI models more accessible and understandable to their users. This is now possible due to feature mutual information, defined in section 5.3, which permits us to define explanations that capture only relevant information in them.

6.1. Explanations Broadening User Knowledge

We are now ready to define the objective to use for the training of explainer models. Specifically, to provide the user with explanations that are local, personalised to the user's understanding of a model, as expressed in a user profile, and free of output-encoding, we propose to train explainer models based on the maximisation of feature mutual information of explanations and model outputs conditioned under the user profile.

Definition 6. Let the random variables X denote input data, Y be the model output, and U be the user profile, and let the explanation mask size be $k \in \mathbb{N}$. We define the *optimal explainer model* $\widehat{\mathcal{E}}$ as the explainer model $\mathcal{E}$ which maximises the feature mutual information of the explanations $\mu(X, S)$ and the model output Y conditioned under the user summaries U, i.e. it is given by the criterion

$$\widehat{\mathcal{E}} = \arg\max_{\mathcal{E}} I(X_S, Y|U) \quad \text{where} \quad S = \mathcal{E}(X). \tag{6.1}$$

This definition provides a global criterion for the optimal explainer model, but it does not indicate a solution for individual data points. Thus, we present the following theorem, which adapts Hypothesis 1 to user-adaptive explanations together with the corrections to it we proposed in section 5.3 to prevent output encoding in them.

Theorem 1. *Let the random variables X denote input data, Y be the model output, and U be the user summary, and let the explanation mask size be $k \in \mathbb{N}$. Further, let $\mathcal{E}^*$ be the explainer model defined by the local criterion*

$$\mathcal{E}^*(x) = \arg\max_{s} \mathbb{E}_{Y,U|X}\Big[\log p(Y|X_s = x_s, U)\Big|X = x\Big]. \tag{6.2}$$

Then $\mathcal{E}^*$ is a solution to the global criterion (6.1), and for any other explainer model $\widehat{\mathcal{E}}$ optimising criterion (6.1), it holds that $\widehat{\mathcal{E}}$ coincides with $\mathcal{E}^*$ almost surely over the data-generating distribution p_X.

The proof of this theorem can be found in appendix A.3.

To get an intuitive understanding of the meaning of criterion (6.1), one has to consider the following quantities of information that are shared among the data, model outputs, and user profile, and which are illustrated in Figure 6.1.

(1) The conditional feature mutual information $I(X_S, Y|U)$ of the visible explanation features X_S with the model outputs Y under the user profile U.

(2) The part $I(X, Y|U) - I(X_S, Y|U)$ of conditional mutual information $I(X, Y|U)$ of X and Y under U which is not contained in (1).

(3) The part $I(X, Y) - I(X, Y|U)$ of mutual information $I(X, Y)$ of X and Y that is contained neither in (1) nor in (2).

These three quantities of information are all non-negative and, thus, form a partition of the entire mutual information $I(X, Y)$ of data X and model outputs Y. The non-negativity of (1) and (2) is a direct consequence of Proposition 1, while (3) is guaranteed to be non-negative under assumption (6.3), which holds for our framework as we define both model outputs Y and user profile U as functions of data X.

$$I(Y, U|X) = 0 \tag{6.3}$$

Furthermore, while the interpretation of (1) as the information contained in user-adaptive explanations is clear, and (2) is just the information about model outputs Y in data X that is not (yet) part of the user-adaptive explanations, the interpretation of (3) is provided by assumption (6.3). This is because, under (6.3), it holds

$$I(X, Y) - I(X, Y|U) = I(Y, U) - I(Y, U|X) = I(Y, U) \tag{6.4}$$

and, hence, it shows us that (3) is the mutual information of model outputs Y and user profile U, which can be interpreted as the knowledge about the model that the user has expressed in the given user profile.

The goal of objective (6.1) is to maximise the feature mutual information (1) of user-adaptive explanations, or equivalently, to minimise the information (2) remaining outside of these explanations. The user-adaptivity of the explanations is introduced by the fact that (1) and (2) only consist of information about model outputs that is not yet known to the user, as the user's knowledge about the model outputs is conveyed in (3), which does not affect objective (6.1). It is important to note, however, that the only constraint to the explanations is the explanation mask size k and, thus, objective (6.1) will not actively avoid including the information (3) in the explanations. Therefore, while the explanations defined by objective (6.1) will be personalised to the user by targeting information not yet known to them, objective (6.1) does not actively avoid redundancy among the explanations and the user's knowledge. Thus, the best way to think intuitively about the user-adaptive explanations defined by our framework is that they aim to broaden the user's knowledge of the model by targeting aspects of the model that the user does not yet know about, without abstaining from repeating things already familiar to the user.

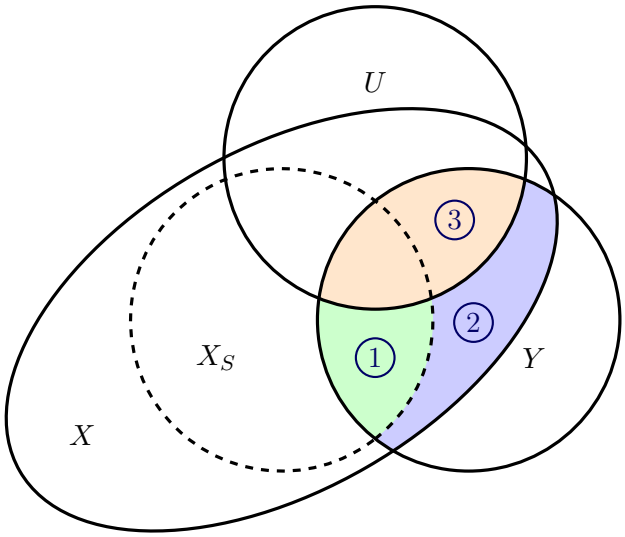


Figure 6.1.: An illustration of the information contents of the three random variables X being the data, Y being the model outputs, and U being the user profile under assumption (6.3). Furthermore, the illustration indicates the feature mutual information of the features X_S contained in user-adaptive explanations of the model. While the relations among the three random variables X, Y, and U are fixed, the feature mutual information of X_S is variable as it is determined by the specification of an explainer model $\mathcal{E}$. The green, blue, and orange shaded areas combined correspond to the mutual information $I(X, Y)$ of data X and model outputs Y, whereas the green and blue shaded areas alone correspond to the conditional mutual information $I(X, Y|U)$ of data X and model outputs Y conditioned under the user profile U. These areas can be further split up into the green shaded area corresponding to (1) the conditional feature mutual information $I(X_S, Y|U)$ of the visible explanation features X_S with the model outputs Y under the user profile U, the blue shaded area alone corresponding to (2) the part of mutual information $I(X, Y|U)$ that is not included in (1), and the orange shaded area corresponding to (3) the mutual information $I(Y, U)$ of model outputs Y and user profile U.

6.2. Implementation of the Framework

Altogether, there are 2 key differences between our framework for user-adaptive explanations and the L2X method, which are

1. the use of feature mutual information in the explanation objective to mitigate output encoding in the explanation masks, and
2. conditioning the explanation objective on user profiles to personalise explanations.

Apart from these 2 aspects, the implementation of our explanation framework coincides with the implementation of the L2X method proposed in [CSWJ18]. Thus, we will first discuss how these two changes relate to the implementation of L2X and, then, provide an overview of the implementation of our framework. Further technical details on the implementation of our experiments are provided in appendix C.1.

The implementation of L2X in [CSWJ18] is facilitated by 3 main components. These are (*i*) the approximation of the global objective (in our case, objective (6.1)) by a variational lower bound, (*ii*) the combination of the different variational models of each explanation mask into a single regressor[1] model, and (*iii*) the joint training of explainer and regressor models through the Gumbel-softmax trick [JGP17, MMT17]. As the Gumbel-softmax trick is only relevant for the back-propagation of gradients from the regressor model to the explainer model, though, it is entirely unaffected by the modifications induced by our framework. Hence, the variational lower bound and the combination of variational models are the only parts of L2X we have to consider to implement user-adaptive explanations.

The variational lower bound itself works perfectly fine within our framework and, compared to L2X, there is only one significant difference with it. This is that the variational models take the user summary as additional input besides the explanation, which is necessary for the personalisation of the explanations. Generally, the variational lower bound is necessary because it is not possible to access conditional probability distributions $p_{Y|X_S,U}$, which would be necessary to implement the local explanation criterion (6.2). Therefore, one has to find an approximation to them to implement feature-mutual-information-based explanations. We do so in the following Proposition, the proof of which can be found in appendix A.4.

Proposition 2. *Let the random variables X denote input data, Y be the model output, and U be the user summary. Further, let $\mathcal{E}$ be an explainer model with explanation mask size $k \in \mathbb{N}$, and let $\mathcal{Q} = \{q_S \,|\, S \subset \{1, \dots, d\}, |S| = k\}$ be a family of variational models. Then, optimising both the explainer model $\mathcal{E}$ and the family of variational models $\mathcal{Q}$ to fulfil*

$$\arg\max_{\mathcal{E},\mathcal{Q}} \mathbb{E}\Big[\log q_S(Y|X_S, U)\Big] \quad \textit{where} \quad S = \mathcal{E}(X) \tag{6.5}$$

is sufficient to optimise the explainer model $\mathcal{E}$ to fulfil criterion (6.1).

[1] As done in section 5.4, we call these models *regressor* models here, as [CSWJ18] does not introduce a specific name for them.

The combination of the different variational models q_s into a single regressor model q

$$(\mu(X,S),Y,U) \mapsto q_S(Y|X_S,U) \iff (\mu(X,S),Y,U) \mapsto q(Y|\mu(X,S),U) \tag{6.6}$$

is introduced by [CSWJ18] to reduce the number of required models from $\binom{d}{k}$ to 1. This, however, poses a problem for our framework because it is not compatible with the use of feature mutual information to mitigate output encoding. To see why, note that in any implementation of the resulting regressor model q, it is not possible to only pass the explanation features X_S to it. This is because that would amount to using $q(Y|X_S,U)$ as explanation objective, but, as we already noted in Observation 1, it is not possible to interpret this quantity without knowing which are the features S. Instead, the explanation mask S is always passed to q as well through the explanation $\mu(X,S)$. But this is precisely the opposite of the idea of feature mutual information as explanation objective to avoid the influence of output encoding on the explanation masks.

Moreover, this problem cannot be resolved by skipping the combination of the variational models. If we were to just pass the visible explanation features X_S to the respective variational model q_S dedicated to a single explanation mask S, then this model would only ever encounter data biased by the occurrence of the mask S. Hence, the mask is implicitly known to the model, and output encoding can be facilitated by it again. Thus, even if we do not combine the different variational models, we cannot avoid that the regressor model always sees the mask of an explanation and, thus, makes the implementation vulnerable to output encoding.

At this point, we would also like to point out that [CSWJ18] commits the same mistake in the derivation of its variational lower bound (5.8) as in its proof of Hypothesis 1 that we pointed out in section 5.3. Due to this, the variational lower bound of L2X ignores the influence of the explanation mask in the mutual information of explanations and model outputs and, thus, it takes the same shape as our variational lower bound derived for the feature-mutual-information-based objective (6.1). Without this error, the regressor model of L2X would take the explanation masks as additional input next to the explanation features, but as discussed before, it is a consequence of Observation 1 that this does not make a difference for the occurrence of output encoding as the explanation mask is always known to the regressor model anyways.

This, however, leaves us with an implementation of our framework that is different from L2X only in the single aspect that the regressor model takes user summaries as additional input next to the explanations. Obviously, this is problematic for our goal to mitigate output encoding in explanations, which is the reason we developed feature mutual information in the first place. Thus, even if we theoretically resolved the problem of output encoding, and despite the two approaches to mitigate output encoding in L2X that we unsuccessfully tested in section 5.6, we are still stuck with an implementation of user-adaptive explanations that allows output encoding to occur, and we have to resort to the empirical rules derived in section 5.5 to mitigate it.

Algorithm 1: One training step of user-adaptive explanations.

Input: ModelToExplain, UserProfile, GumbelGate, LossFunction
Data: Input data x_i, mask size $k \in \mathbb{N}$
Models: ExplainerModel, RegressorModel

1 **Compute explanation**;
2 $\text{FeatureScores}_i \leftarrow \text{ExplainerModel}(x_i)$;
3 $\text{Mask}_i \leftarrow \text{GumbelGate}(\text{FeatureScores}_i, k)$;
4 $\text{Explanation}_i \leftarrow x_i \odot \text{Mask}_i$;
5 **Compute user summary**;
6 $\text{UserSummary}_i \leftarrow \text{UserProfile}(x_i)$;
7 **Compute loss**;
8 $\text{RegressorInput}_i \leftarrow [\text{Explanation}_i \mid \text{UserSummary}_i]$;
9 $y_{\text{pred}} \leftarrow \text{RegressorModel}(\text{RegressorInput}_i)$;
10 $y_{\text{true}} \leftarrow \text{ModelToExplain}(x_i)$;
11 $\text{Loss} \leftarrow \text{LossFunction}(y_{\text{true}}, y_{\text{pred}})$;
12 **Update models**;
13 Compute gradients of Loss;
14 Update ExplainerModel and RegressorModel;

Algorithm 2: Gumbel-Softmax Trick.

Function: GumbelGate
Input: FeatureScores_i, mask size $k \in \mathbb{N}$
Data: data dim. $d \in \mathbb{N}$, temperature $t > 0$
Output: ExplanationMask_i

1 $\text{Noise}_{ij} \leftarrow \text{GumbelNoiseGenerator}((d, k))$;
2 $\text{NoisyScores}_{ij} \leftarrow \text{FeatureScores}_i + \text{Noise}_{ij}$;
3 $\text{ExpScores}_{ij} \leftarrow \exp(\text{NoisyScores}_{ij} \cdot t)$;
4 $\text{SoftMax}_{ij} \leftarrow \text{ExpScores}_{ij} / \sum_i \text{ExpScores}_{ij}$;
5 $\text{ExplanationMask}_i \leftarrow \max_j \text{SoftMax}_{ij}$;
6 **return** ExplanationMask_i;

Algorithm 3: Getting user-adaptive explanations from an explainer model.

Input: ExplainerModel
Data: Input data x_i, mask size $k \in \mathbb{N}$
Result: Explanation_i

1 $\text{FeatureScores}_i \leftarrow \text{ExplainerModel}(x_i)$;
2 $\text{Mask}_i \leftarrow [1 \text{ if } s \text{ in top } k \text{ of } \text{FeatureScores}_i, \text{ else } 0, \text{ for } s \text{ in } \text{FeatureScores}_i]$;
3 $\text{Explanation}_i \leftarrow x_i \odot \text{Mask}_i$;

Altogether, taking into account the use of feature mutual information, conditioning on user summaries, and their interaction with the implementation of L2X, for user-adaptive explanations we thus arrive at the explanation objective

$$\arg\max_{\mathcal{E},q} \mathbb{E}\Big[\log q(Y|\mu(X,S),U)\Big] = \mathbb{E}\Big[\log q(\mathcal{M}(X)|\mu(X,\mathcal{E}(X)),\mathcal{U}(X))\Big] \tag{6.7}$$

which can be solved by methods of empirical risk minimisation to train the explainer model $\mathcal{E}$ and the regressor model q by sampling from the data-generating distribution p_X. An exemplary implementation of our explanation framework based on the implementation of L2X by [CSWJ18] is provided in Algorithms 1 to 3, showing the steps required to train an explainer model and use it for the creation of user-adaptive explanations.

Algorithm 1 shows the instructions to execute one training step in the joint training of the explainer and regressor models on a single data point. To do so, first, in lines 2 to 4 we compute a differentiable explanation by taking the explainer model output as feature importance scores, transforming them into a differentiable explanation mask through the Gumbel-Softmax trick (see Algorithm 2), and applying this mask to the data by element-wise multiplication. After obtaining the user summary in line 6, we concatenate explanation and user summary to feed them to the regressor model and get a prediction of the model output on lines 8 and 9. On lines 10 and 11, we get the true model output and compute the loss between it and its prediction by the regressor model. Finally, on line 13 we compute the gradients of the loss with respect to the regressor and explainer models, where we use the differentiability of the explanation given by the Gumbel-Softmax trick, and on line 14 we update the weights of these models.

Algorithm 2 shows the implementation of the Gumbel-Softmax trick, which we adopt from [CSWJ18], as definition of the respective function used in Algorithm 1. We start by drawing $d \cdot k$ random samples from a standard Gumbel distribution in line 1 for data dimension d and explanation mask size k. This noise is added to the feature importance scores in line 2, where the 1-dimensional feature scores are broadcast to the 2-dimensional noise. Then, on lines 3 and 4, we apply the softmax function with temperature t to the noisy scores along the first dimension, and finally take the maximum of the resulting scores over the second dimension to obtain the differentiable explanation mask in line 5.

Algorithm 3 shows how to obtain an explanation for a data point from a given explainer model. Unlike the training of the explainer model, this does not require a differentiable explanation mask and, thus, the procedure does not make use of the Gumbel-Softmax trick. In line 1 we start by obtaining the feature importance scores from the explainer model, and in line 2 we build an explanation mask by determining which are the k scores with the highest values, for k the explanation mask size. Finally, in line 3, we construct the explanation by element-wise multiplication of the data with the explanation mask.

6.3. Evaluation Methods

Having defined our framework, determined its implementation, and found that it is inherently susceptible to output encoding, there are two questions to answer:

Q1 Does the method work on a fundamental level, i.e. are user-adaptive explanations defined by the feature mutual-information-based objective (6.1) informative about the model outputs for a given user profile?

Q2 How prevalent is output encoding in the user-adaptive explanations produced by the implementation described in section 6.2, and can one derive similar empirical rules for its occurrence as we did for L2X in section 5.5?

We will answer these questions in sections 6.4 and 6.5 respectively, and to do so, we will again use the methods introduced in section 5.4 to examine output encoding in L2X.

More specifically, to answer question Q1 about the suitability of explanation objective (6.1), we employ 16-dimensional linear regression models, define several user profiles of user summary sizes $l = 2, 4, 8$, and compute exact user-adaptive explanations for the combinations of these models, user profiles, and explanation mask sizes $k = 1, \ldots, 15$. We only consider exact explanations, meaning explanations determined by analytically solving objective (6.2) as introduced in section 5.4, to assess this question because these are, by definition, free of output encoding. Thus, they allow us a clear and isolated view of the performance of objective (6.1) without the distortions induced by output encoding.

To evaluate user-adaptive explanations created in this manner, we measure feature mutual information and full-explanation output prediction losses that the explanations have together with the respective user profiles. For each combined explanation mask and user summary size $c = k + l$, we compare these metrics across user summary sizes, and to

the respective non-user-adaptive explanations. Here, it is important to understand that, for a given value of c, non-user adaptive explanations always achieve the best possible metrics because they lack the constraint of user-adaption. For user-adaptive explanations, on the other hand, their metrics at $k = 0$ are constrained to the metrics of the respective user profiles, and the explanations have to improve starting from this point. Therefore, we compare user-adaptive explanations to non-user-adaptive explanations for fixed values of c because, like this, we see how the quality of user-adaptive explanations is impacted by the constraint of personalisation to user profiles, and how well the explanations build on the user's knowledge towards the overall optimum.

To answer question Q2 about the prevalence of output encoding in user-adaptive explanations implemented according to our method, we use the same linear regression models and user profiles as for question Q1, but now compute learned explanations for them. To measure the effects of output encoding in learned user-adaptive explanations, we record full-explanation and mask-only output prediction losses that explanations have together with the respective user profiles. We compare these metrics for each combined explanation mask and user summary size across user summary sizes, and to the respective non-user-adaptive explanations, to study the intensity of output encoding in these explanations and under which conditions it occurs.

We study user-adaptive explanations personalised to the following user profiles, each with user summary size l and using the features of the input data.

First user profile: Always selects the l features corresponding to the largest absolute value weights of the respective linear regression model. It is a global user profile, the best such user profile as measured by its mutual information with model outputs, and simulates a use case where the user has a good understanding of the model in general.

Last user profile: Always selects the l features corresponding to the smallest absolute value weights of the respective linear regression model. It is a global user profile, the worst such user profile as measured by its mutual information with model outputs, and simulates a use case where the explanations have to provide a lot of information.

Random user profile: Selects l features at random out of the available features every time. It is a local user profile and serves as a baseline user profile simulating the case of a completely uninformed user.

Absolute user profile: For each data point, selects its l features with the largest absolute values. It is a local user profile and, depending on the model, somewhat better than user profile 'random'. Thus, it simulates a user who has some understanding of the model.

Greedy user profile: For each data point, selects its l features that yield the highest absolute value when multiplied by the respective weight of the linear regression model. It is a local user profile and, depending on the model, somewhat worse than user profile 'exact'. Thus, it simulates a user with near-optimal understanding of the model.

Exact user profile: Selects the exact explanation of explanation mask size l according to objective (6.1) as user summary for each data point. It is a local user profile, the overall best user profile as measured by its mutual information with model outputs, and simulates the use case of optimal model understanding by the user.

Finally, we consider two linear regression models for our experiments, which we have already examined as Models 1 and 5 in section 5.5. For the experiments presented in the following sections we will denote them as Model A (= Model 1 in section 5.5) and Model B (= Model 5 in section 5.5) respectively. We only consider these two linear regression models here because the results observed on them are representative of the results observed also on the remaining models. In general, the observations we make for user-adaptive explanations are less diverse than the observations made in section 5.5 and the regularised Model A and unregularised Model B are sufficient to represent them.

The full technical details of the evaluation methods used are provided in appendix C.1, and an overview of the linear regression models and user profiles is given in appendix C.3.

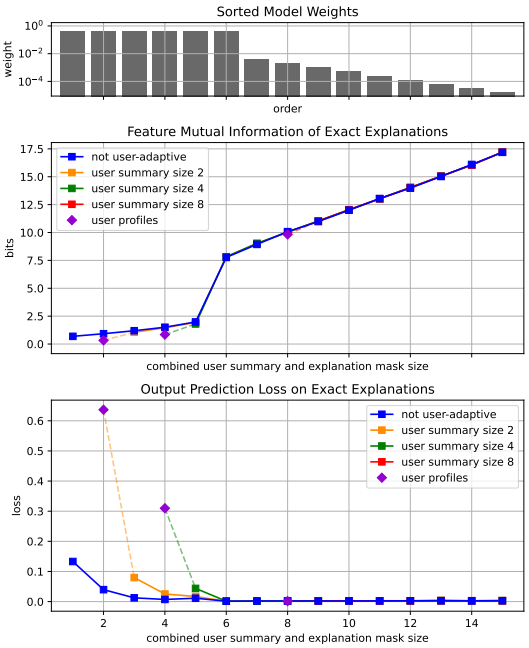
6.4. Performance of User-Adaptive Explanations

In Figure 6.2 we present the results of our experiments assessing the performance of exact user-adaptive explanations to answer question Q1. We will now discuss what one can observe in these results for each of the examined user profiles.

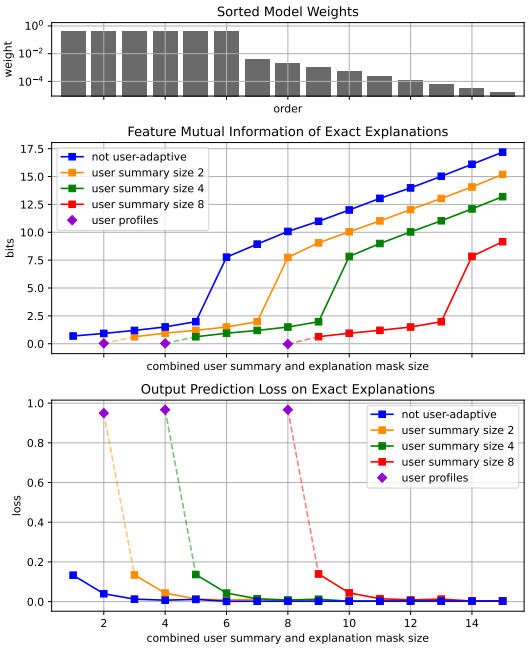
First In the results of user profile 'first', displayed in Figures 6.2a and 6.2c, we see an excellent performance of the user adaptive explanations. As for Model A the user profiles already show quite a good understanding of the model in most cases, the user-adaptive explanations need to provide only very few additional features to provide the user with the same understanding of the model as is given in the non-user-adaptive explanations. For Model B, on the other hand, the user profiles show a significantly worse understanding of the model than the non-user-adaptive explanations, and, accordingly, it takes explanations of around size 3 to match the user profiles with the non-user-adaptive explanations.

Last In the results obtained with user profile 'last', shown in Figures 6.2b and 6.2d, we can see how user-adaptive explanations handle the low feature mutual information of this user profile. While the output prediction losses recorded for both models show again that user-adaptive explanations only require small explanation mask sizes to make up for the lack of information provided in the user profiles, the measurements of feature mutual information reveal a different view of the situation. In contrast to other user profiles, the feature mutual information observed for explanations adapted to user profile 'last' is growing at roughly the same rate as for non-user-adaptive explanations, but not able to catch up with them. This tells us, on the one hand, that user-adaptive explanations struggle to fill the lack of information in these user profiles. But, on the other hand, this also demonstrates that user-adaptive explanations are capable of delivering the information necessary to complement the user's knowledge even in the most difficult cases.

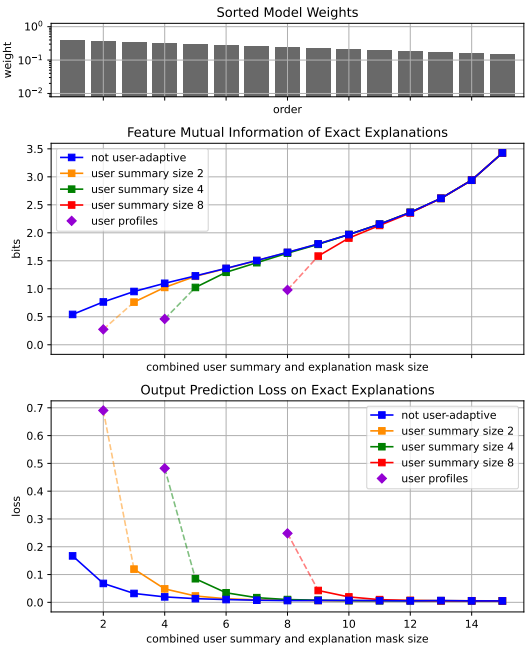
Random The results regarding user profile 'random', which can be found in Figures 6.2e and 6.2g, provide us with insights into the behaviour of user-adaptive explanations in the case of an uninformed user who is selecting features (approximately) at random.



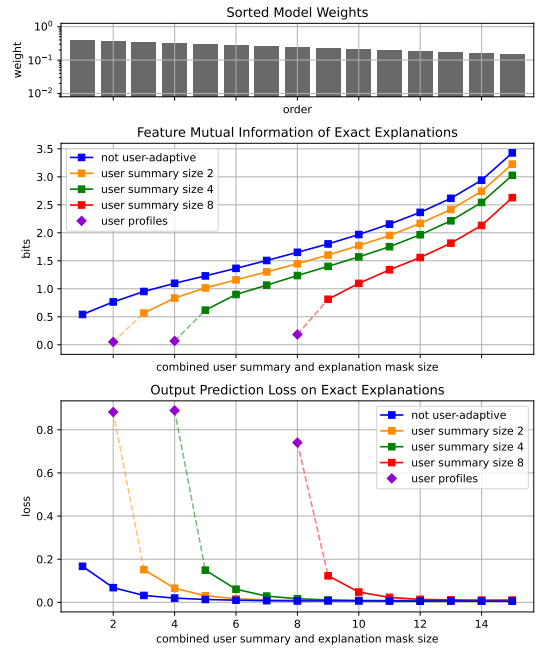
(a) Model A, User Profile 'first'



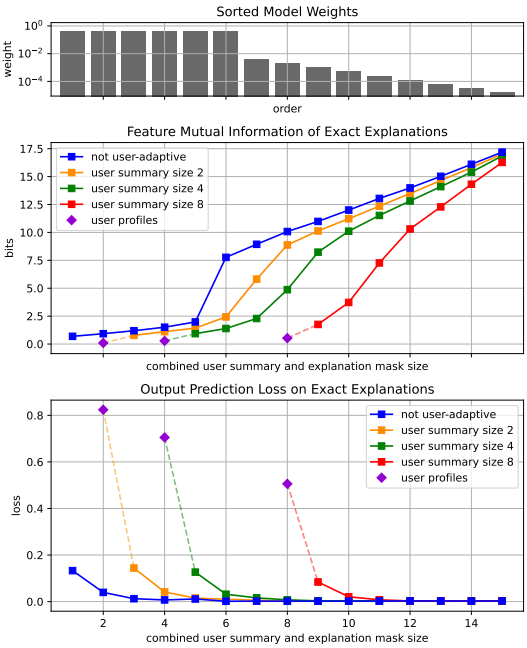
(b) Model A, User Profile 'last'



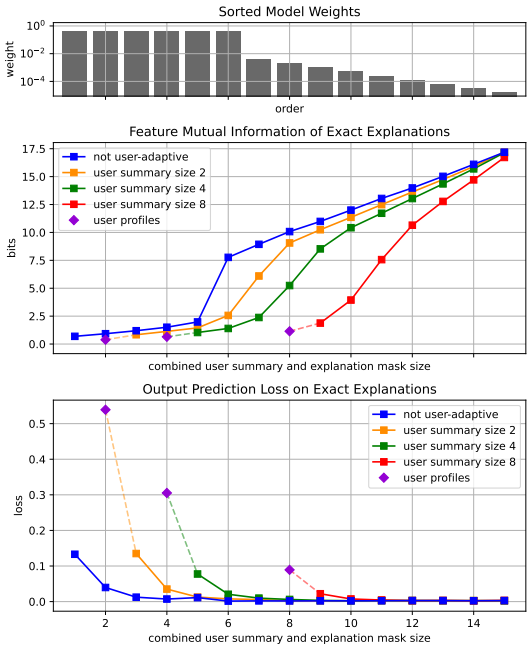
(c) Model B, User Profile 'first'



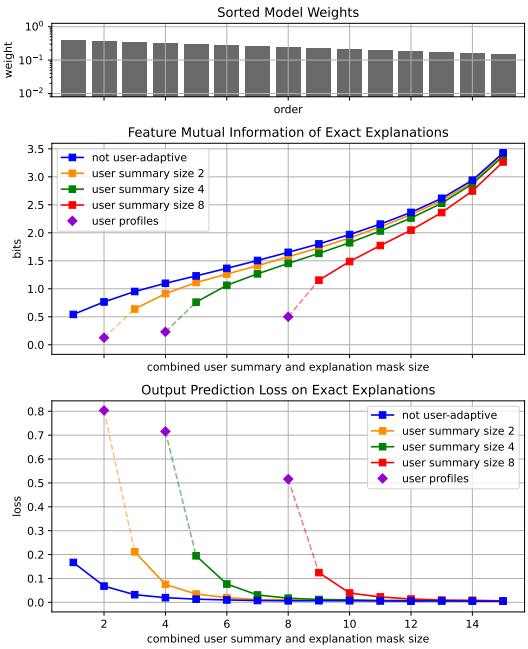
(d) Model B, User Profile 'last'



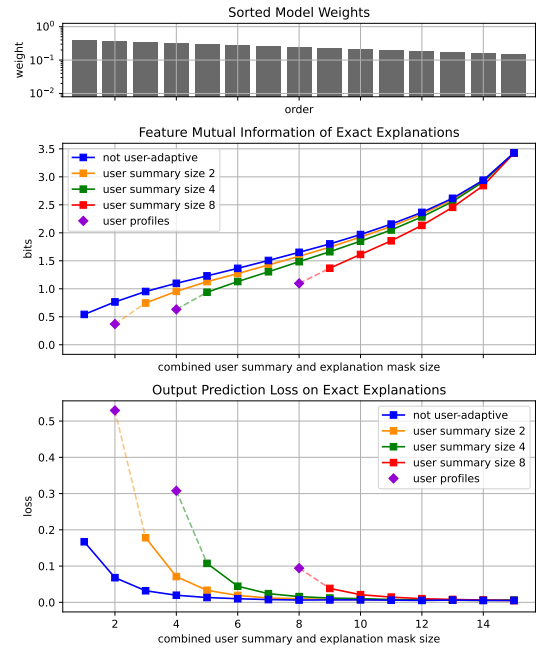
(e) Model A, User Profile 'random'



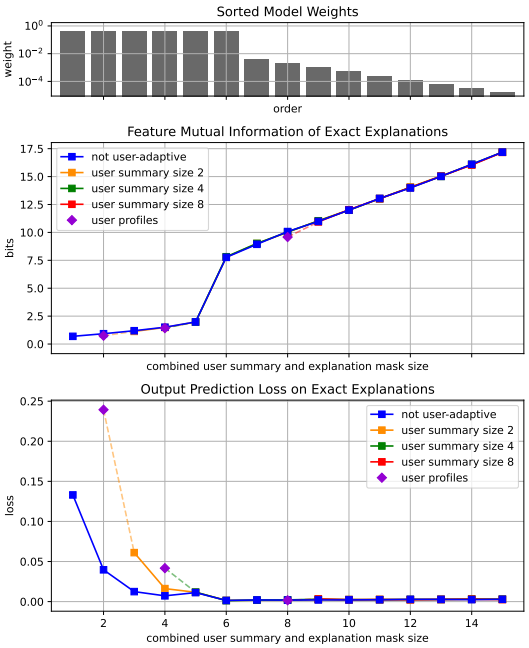
(f) Model A, User Profile 'absolute'



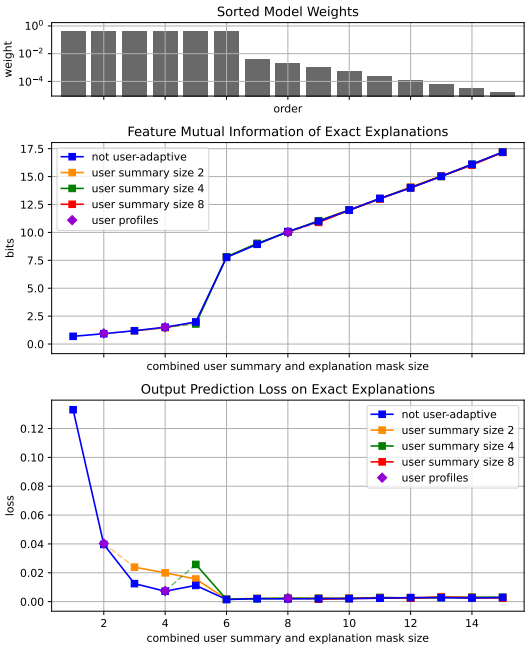
(g) Model B, User Profile 'random'



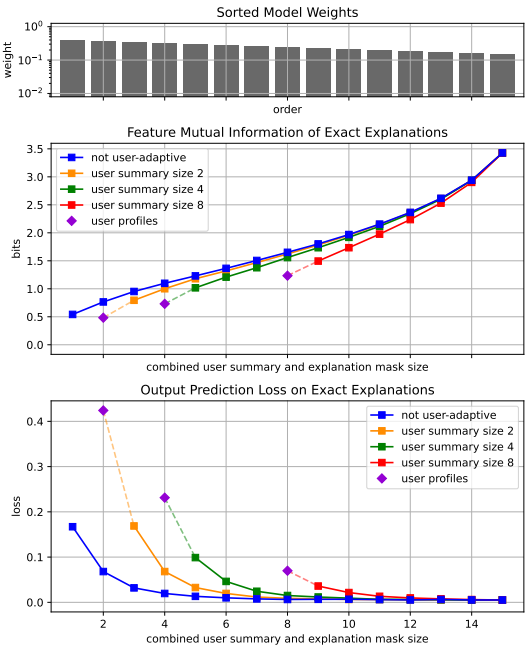
(h) Model B, User Profile 'absolute'



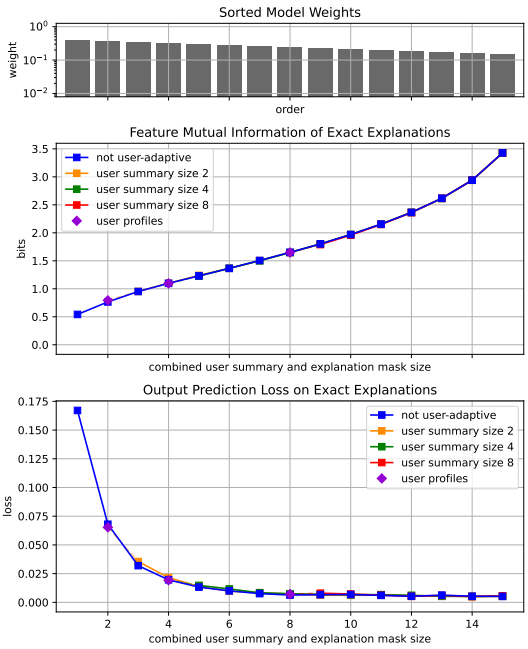
(i) Model A, User Profile 'greedy'



(j) Model A, User Profile 'exact'



(k) Model B, User Profile 'greedy'



(l) Model B, User Profile 'exact'

Figure 6.2.: Plots showing the results of the experiments discussed in Section 6.4. Each subfigure corresponds to one combination of linear regression model and user profile. The sorted weights of the respective model are shown in the top subplots, the middle and bottom subplots show feature mutual information and output prediction losses of exact user-adaptive explanations and the respective user profiles plotted against the combined user summary and explanation mask sizes. As we ignore the trivial explanation mask sizes 0 and 16, the x-axis ranges only from 1 to 15. Consequently, the top subplots only show 15 of the 16 weights of the models. The remaining weights are fully determined by the displayed ones, though, as the squared weights sum to 1 for each model.

In this case, we find again a very good performance of the user-adaptive explanations, as measured by the output prediction losses quickly approaching the scores of non-user-adaptive explanations despite relatively little information contained in the user profiles. Importantly, the feature mutual information scores of explanations adapted to this user profile, while starting out low due to the uninformedness of the user profile, grow more quickly than the feature mutual information of non-user-adaptive explanations and eventually catch up with them. This is a critical insight, however, as it proves that user-adaptive explanations can, even if they are provided with a completely uninformed user, overcome the lack of information found in such a user profile and still keep up with the quality of non-user-adaptive explanations.

Absolute The results of user profile 'absolute', given in Figures 6.2f and 6.2h, mostly reflect our observations made for user profile 'random'. While user profile 'absolute' is noticeably better than user profile 'random', as indicated by higher feature mutual information scores and lower output prediction losses, the metrics of the user-adaptive explanations themselves follow similar trajectories as for user profile 'random'. This is unsurprising, but it shows us that, also in the case of a user profile which is somewhat better informed than a random baseline, user-adaptive explanations are able to effectively inform the user about the model at hand.

Greedy The results obtained for user profile 'greedy', displayed in Figures 6.2i and 6.2k, are again similar to what we have seen for user profile 'absolute', but also show us that they form an easier setting for user-adaptive explanations. As these user profiles show a quite good understanding of the model already, the difference in model understanding in them compared to the non-user-adaptive explanations is getting ever smaller, which is especially evident for Model A where user profile 'greedy' basically coincides with non-user-adaptive explanations already. Nonetheless, the explanations adapted to these user profiles show a very good performance as measured by the output prediction losses, and also the feature mutual information scores are either identical to those of non-user-adaptive explanations or quick to catch up to them.

Exact Finally, as for the results about user profile 'exact', shown in Figures 6.2j and 6.2l, there is really not much to say about them in the context of explanation performance. This is because the user profiles are already on the level of the non-user-adaptive explanations, and, thus, the performance of the user-adaptive explanations is basically indistinguishable from the non-user-adaptive explanations. An interesting exception to this occurs, however, for the cases of Model A where several weights have the same value. There, one can find some irregularities in the observed output prediction losses, and it is not clear what causes them to occur.

Summary

In total, we have seen in these experiments that user-adaptive explanations have a good performance on a wide range of application cases. While user-adaptive explanations show a very good performance of overcoming the restrictions imposed by and building on existing user knowledge for user profiles 'first', 'greedy' and 'exact', where it is easy to do so, user-adaptive explanations also show a satisfying performance when applied to user profile 'last', constituting the most difficult case. Furthermore, when applied to user profiles 'random' and 'absolute', which are especially important because they realistically reflect users with little to no understanding of the model to be explained, user-adaptive explanations show that they are capable of providing these users with the information that is necessary to improve their understanding of the models. Therefore, we can conclude that user-adaptive explanations, as defined by objective (6.1), are informative about the model outputs for a given user profile, which positively answers question Q1 that we set out to resolve with these experiments.

6.5. Output-Encoding in User-Adaptive Explanations

In Figure 6.3, we present the results of our experiments assessing the occurrence of output encoding in learned user-adaptive explanations to answer question Q2. We will now discuss what one can observe in these results for each of the examined user profiles.

First In the results obtained for user profile 'first', which are shown in Figures 6.3a and 6.3c, we find that output encoding in learned user-adaptive explanations for this user profile largely follows the occurrence of output encoding in non-user-adaptive explanations. More specifically, we observe that, for both models and all user summary sizes, both the full-explanation and mask-only output prediction losses of user-adaptive explanations closely follow the output prediction losses of non-user-adaptive explanations after quickly approaching them initially. This is especially interesting, as the user profiles themselves are free of output encoding, which can be seen from the fact that all mask-only output prediction losses of the user profiles are at the level of the output variance. Nonetheless, the user-adaptive explanations only need explanation mask sizes of at most 3 to make up for the lack of output encoding in the user profiles before their output prediction losses reach the same level as seen in the respective non-user-adaptive explanations. Based on

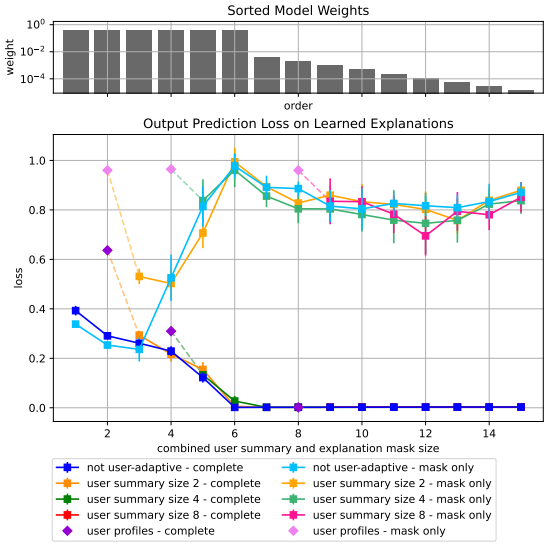
this, we form an initial conjecture that, at least for global user profiles, output encoding in user-adaptive explanations follows the same patterns that we described in section 5.5, where we characterised the occurrence of output encoding based on the relation of explanation mask size and model weights in criterion (5.14). The only difference in this case, it seems, is that for user-adaptive explanations the user profile size contributes to this threshold and, thus, the occurrence of output encoding is offset accordingly.

Last The results observed for user profile 'last', which can be found in Figures 6.3b and 6.3d, strongly corroborate our observations made for user profile 'first'. This comes as, for user profile 'last', output encoding is strongly present in the user-adaptive explanations and, for all user-summary sizes, appears in an identical pattern as in the case of non-user-adaptive explanations. Also, we see again that there is no output encoding occurring in the user profiles themselves, while it takes place in all user-adaptive explanations with an intensity comparable to that of the respective non-user-adaptive explanations. The only difference to the case of user profile 'first' is that here the progression of output encoding over the explanation mask sizes is not aligned with its progression in non-user-adaptive explanations, but it is offset according to the lack of model output variance explained by user profile 'last'. This effect is more prominent for Model A having less uniform weights and smoothly transitions towards the patterns seen for the 'first' profile for models with more uniform weights (as the two user profiles are identical there), which is why the effect is less pronounced in Model B.

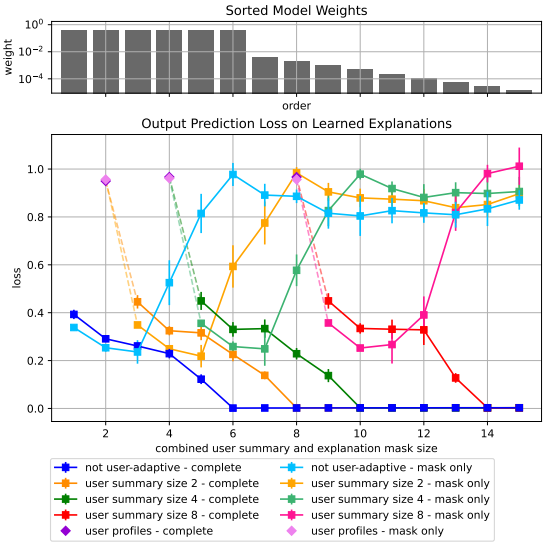
This means that we have to adapt our initial conjecture about criterion (5.14) holding in the same way for user-adaptive explanations with global user profiles so that the user profile does not contribute to the threshold in the same way as the explanations. Instead, it merely offsets the threshold by the weights belonging to the features contained in the global user profile. Thus, we postulate that for user-adaptive explanations with explanation masks s and a global user profile with feature subset u, output encoding occurs for explanation mask sizes k for which the adapted criterion (6.8) holds.

$$\max_{|s|=k} \sum_{i \in s \cup u} \alpha_i^2 < 0.5 \tag{6.8}$$

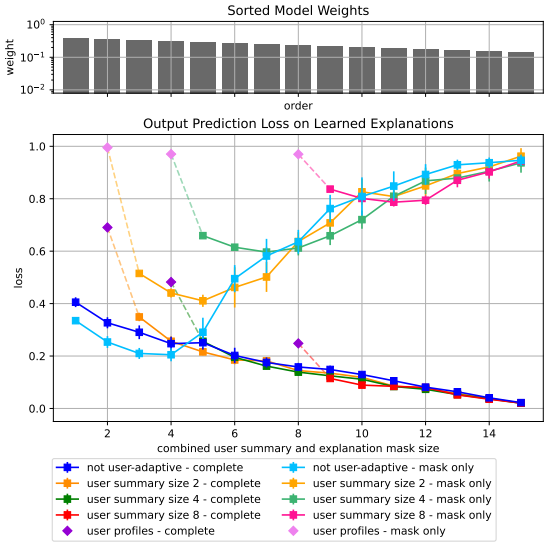
Random In the experiments conducted with user profile 'random', shown in Figures 6.3e and 6.3g, we make new interesting observations. While the results overall resemble the measurements made for user profile 'last', we also notice some differences to the previously examined global user profiles. On the one hand, the output prediction losses of the user-adaptive explanations do not follow the losses of the non-user-adaptive explanations as closely as is the case for the global user profiles. On the other hand, the output encoding occurring here appears to weaken with growing user summary size. These observations do not already permit new conclusions, but they give us a first insight into how local user profiles generally behave differently from global ones.



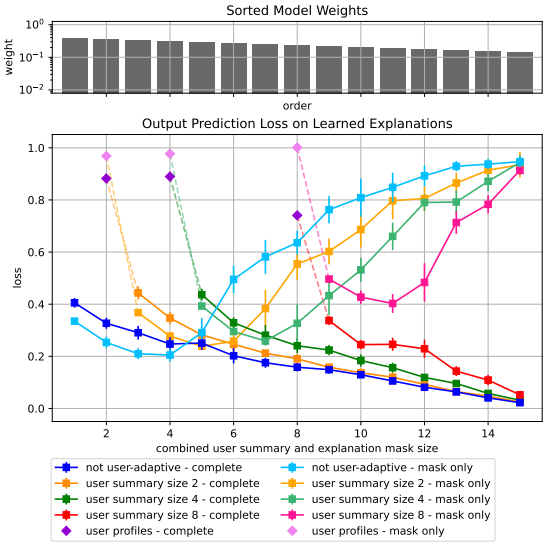
(a) Model A, User Profile 'first'



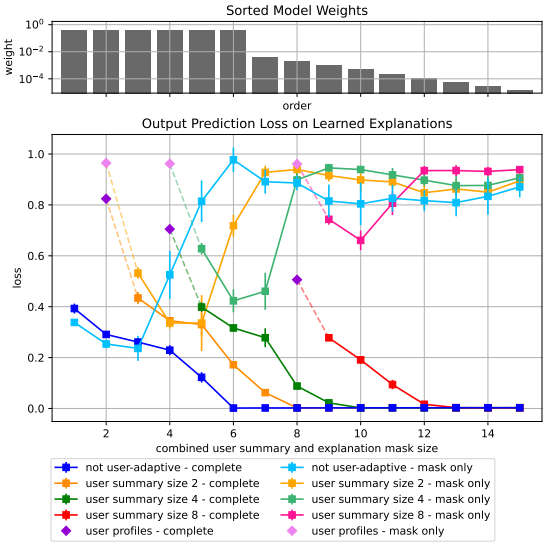
(b) Model A, User Profile 'last'



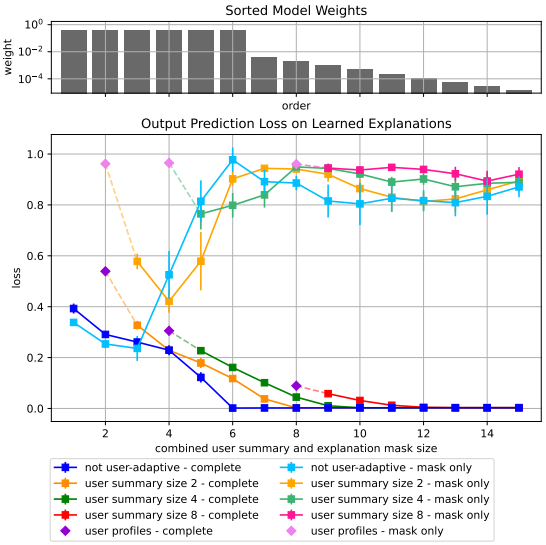
(c) Model B, User Profile 'first'



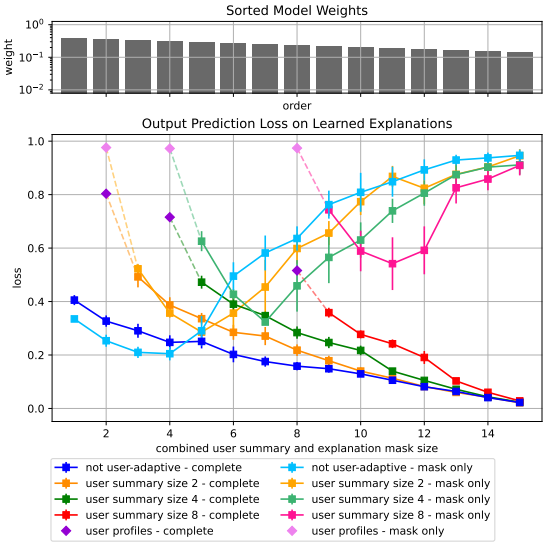
(d) Model B, User Profile 'last'



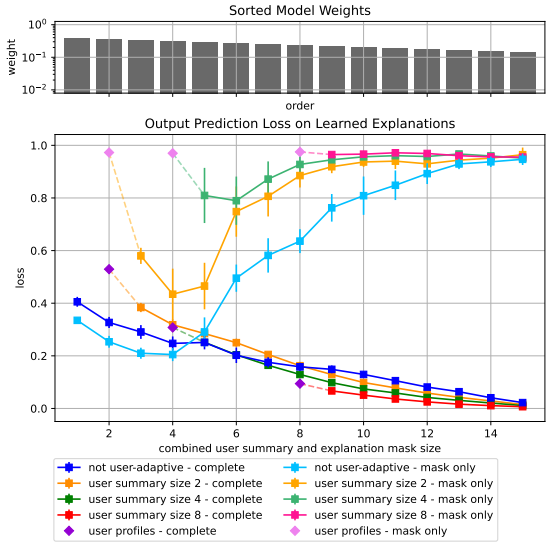
(e) Model A, User Profile 'random'



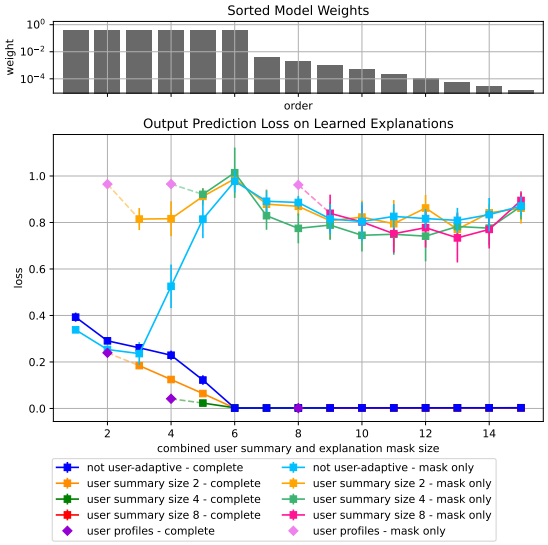
(f) Model A, User Profile 'absolute'



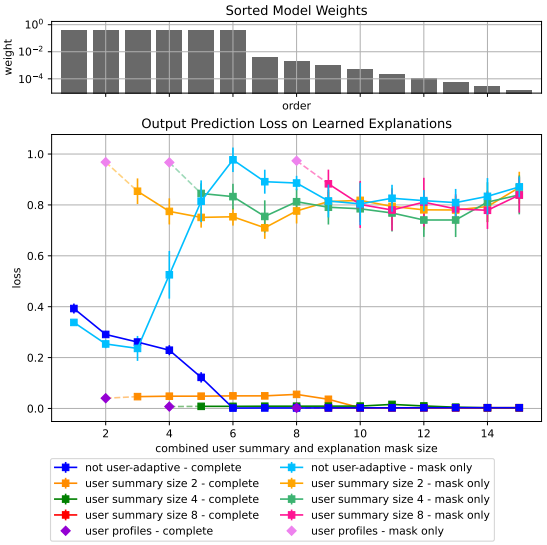
(g) Model B, User Profile 'random'



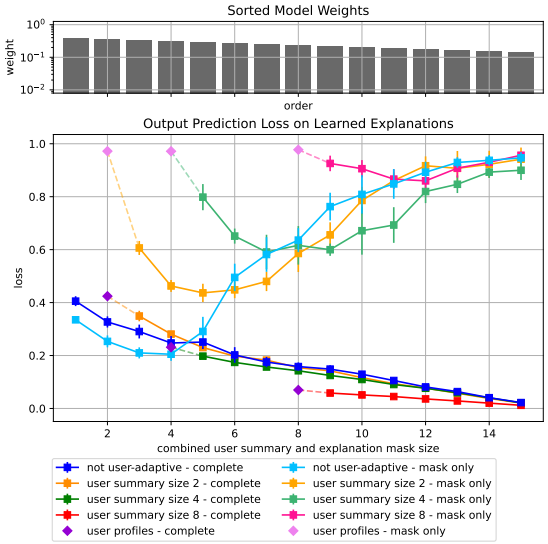
(h) Model B, User Profile 'absolute'



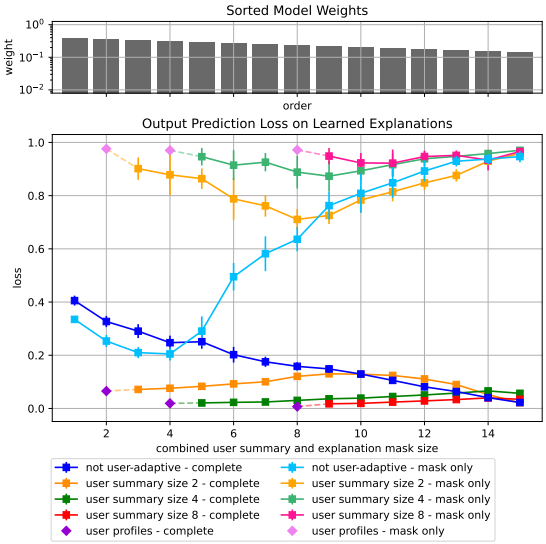
(i) Model A, User Profile 'greedy'



(j) Model A, User Profile 'exact'



(k) Model B, User Profile 'greedy'



(l) Model B, User Profile 'exact'

Figure 6.3.: Plots showing the results of the experiments discussed in Section 6.5. Each subfigure corresponds to one combination of linear regression model and user profile. The sorted weights of the respective model are shown in the top subplots, the bottom subplots show output prediction losses of learned user-adaptive explanations and the respective user profiles plotted against the combined user summary and explanation mask sizes. Vertical error bars show the standard deviation of the results observed in 10 randomly initialised and trained explainer models. As we ignore the trivial explanation mask sizes 0 and 16, the x-axis ranges only from 1 to 15. Consequently, the top subplots only show 15 of the 16 weights of the models. The remaining weights are fully determined by the displayed ones, though, as the squared weights sum to 1 for each model.

Absolute The results regarding user profile 'absolute', which are displayed in Figures 6.3f and 6.3h, show a continuation of the situation that we have found in the results of user profile 'random'. While the overall situation in these results looks more like the results obtained for user profile 'first', the differences between user profiles 'first' and 'absolute' appear to be the same as between user profiles 'last' and 'random'. More precisely, we observe that the output prediction losses of user-adaptive explanations do not follow the output prediction losses of non-user-adaptive explanations as tightly as with the global user profiles, and we observe that output encoding seems to become weaker with larger user summary sizes. Most importantly, we see here for the first time a clear separation between the mask-only output prediction losses of user-adaptive and non-user-adaptive explanations, meaning that user-adaptive explanations for user profile 'absolute' have much less output encoding than those of all user profiles examined so far.

Therefore, we can identify two trends in output encoding in user-adaptive explanations with local user profiles: On the one hand, output encoding in user-adaptive explanations appears to decrease with increasing user summary sizes. On the other hand, output encoding also appears to decrease with increasing knowledgeability of the user profiles, as we have found significantly less output encoding in the results for user profile 'absolute' than with user profile 'random'. Combining these two observations, we hypothesise that the information about model outputs conveyed by the user profile is the main factor modulating output encoding in user-adaptive explanations with local user profiles.

Greedy The results of the experiments employing user profile 'greedy', provided in Figures 6.3i and 6.3k, show us a more ambiguous perspective on the occurrence of output encoding. The results obtained for Model A show an excellent quality of the user-adaptive explanations with very low full-explanation output prediction losses and very high mask-only output prediction losses, meaning that they contain almost no output encoding altogether. In contrast to this, the results for Model B are rather bad as the mask-only output prediction losses of user-adaptive explanations are mostly below those of non-user-adaptive explanations, meaning that, compared to user profile 'absolute', output

encoding is stronger here again. While this does not affirm the part of our conjecture which says that more knowledgeable user profiles should experience less output encoding, it aligns better with the other part of it, as user-adaptive explanations for larger user summary sizes show less output encoding.

Exact Finally, in the results obtained with user profile 'exact', which are given in Figures 6.3j and 6.3l, we can observe very little output encoding. In both examined models, the full-explanations output prediction losses of user-adaptive explanations are generally much lower than for all previously examined cases, although the measurements for user summary size 2 show some strangely high values compared to the other user summary sizes, and it is not clear what causes this. Similarly, all user-adaptive explanations for this user profile have mask-only output prediction losses that indicate very little to no output encoding occurring in them, with user summary size 2 in Model B showing also here an unusual anomaly in deviating from the other user summary sizes. Lastly, we can again observe in these results the effect that user-adaptive explanations with larger user summary sizes tend to show much less output encoding going on in them, and, thus, these results collectively strengthen our hypothesis about the relation of user profile quality and output encoding in user-adaptive explanations again.

Summary

In conclusion, the experiments conducted in this section to examine the occurrence of output encoding in learned user-adaptive explanations show us a two-fold picture.

On the one hand, we found in user-adaptive explanations with global user profiles that the occurrence of output encoding largely follows the empirical rules we derived for L2X explanations in section 5.5. This can be explained by global user profiles having basically the same effect on learned explanations as blocking the respective input features of the model. This influence, we have formalised in criterion (6.8), which takes account of the features represented in the user profile and allows for the prediction of output encoding in user-adaptive explanations.

For local user profiles, on the other hand, we found that the occurrence of output encoding resembles the situation for global user profiles, but the rules derived in section 5.5 do not work well for them anymore. Instead, we attempted to characterise output encoding in these explanations by relating it to the information about the model outputs contained in the user profile, as it appeared to increase with user summary size and knowledgeability of the user profile. Indeed, we were able to gather some evidence for this hypothesis as the user profiles 'random', 'absolute' and 'exact' showed largely promising results. But, ultimately, this evidence is not very strong and the results of user profile 'absolute' do not align with it, leaving the validity of our hypothesis an open question.

Therefore, we have to conclude that, based on the results obtained in the experiments conducted in this section, it is only partially possible to predict the occurrence of output encoding in user-adaptive explanations implemented according to our proposed method. Thus, we are not able to fully answer question Q2 that we sought to resolve here, and it remains an open problem whether it is possible to do so.

7. Discussion and Future Work

In the course of this thesis, we raised several questions and obtained a number of results to answer them. For each of the contributions of this work, let us discuss how they answer the questions we encountered, their limitations, and further research necessary in them.

Feature Mutual Information

As a result of our analysis of the flawed proof of Hypothesis 1 by [CSWJ18] in section 5.3, we introduced the so-called feature mutual information of explanations. Through Proposition 1, this information-theoretic tool allows us to consider the mutual information that the features of explanations have with model outputs without the influence that explanation masks have on them. Therefore, the notion of feature mutual information permits us to rigorously assess the quality of explanations independently of the effects that output encoding can have on them, and it enabled us to fix Hypothesis 1 and transform it into the foundation of our user-adaptive explanations in the form of Theorem 1.

The main limitation of the approach used to derive feature mutual information is that it is not founded on a deeper information theory of masked random vectors, but is rather an ad hoc definition that proved to work well. We made several attempts to conceive approaches that could provide a broader theoretical underpinning of feature mutual information, among which were analytic approaches to find a meaningful definition of the entropy of masked random vectors, attempts to generalise the existing information theoretic definitions by introducing modified probability densities, and methods to quantify information flow in graphs. However, none of the approaches we examined led to a satisfying information theory of masked random vectors, as, in each case, we ran into the same problem of requiring an exact description of the co-occurrence of the values of features visible through the mask. The existence of such a description seems plausible, albeit beyond the scope of this thesis, and it is certainly possible to come up with further approaches to solve this problem and examine them in future work.

Output Encoding in L2X

In the course of chapter 5, we have thoroughly investigated the theoretical and practical ramifications of the explanation objective of the L2X method. To do so, we started by reasoning from first principles about the consequences of optimising local explanations to fulfil a mutual-information-based objective and, thereby, discovered the phenomenon of output encoding in explanations. As it poses a serious risk towards the integrity of explanations, we went on to quantify this phenomenon and to understand it theoretically. Output encoding was not accounted for in the publication [CSWJ18] introducing L2X,

but it was previously examined in [JSAR21] and [PNR24]. Ultimately, in the experiments presented in section 5.5, we were able to clearly demonstrate that output encoding does occur in L2X explanations. Furthermore, we were able to characterise the occurrence of output encoding in L2X explanations of linear regression models based on the used explanation mask size and model weights. Due to this, we could successfully validate Hypothesis 2, show that [CSWJ18] is severely lacking in understanding of its method, and lay an important foundation for the evaluation of our own explanation method. Finally, we also investigated two modifications of the implementation of L2X to mitigate output encoding and found that they do not yield satisfying results.

As discussed in full detail in sections 5.5 and 5.6, the main limitation of this contribution is its limited scope, comprising only a few hand-crafted linear regression models and global evaluations of their explanations. The future work required in this area consists of widening the scope of these experiments and improving our understanding of the output encoding effects observed in them.

Evaluation of Explanations

As we have pointed out in our literature review, the rigorous evaluation of explanations is still a major open problem in XAI. This is due to there not being a generally established definition of what an explanation is or what properties it should have. Consequently, evaluations of XAI methods are usually insufficient, only consisting of examples or being absent altogether. With the evaluation methods used in our experiments of (i) measuring feature mutual information and full-explanation/mask output prediction losses, and (ii) comparing these across different explanation mask sizes, we provide novel methods to rigorously evaluate explanations, especially regarding output encoding, and offer a way to clearly distinguish informative explanations from bad ones.

The limitations of our evaluation methods are, on the one hand, that the feature mutual information of explanations can only be computed if one has access to the probability densities $p(y|x_s)$ of the model output under the different feature subsets, or an approximation thereof. On the other hand, its high computational complexity is a limitation of our evaluation of explanations because both the Monte-Carlo estimation of feature mutual information and the training of regressor models to compute output prediction losses require a large number of explanations to be computed. Thus, the main objective for further research to improve our evaluation method should be to generalise them to models for which it is not straightforward to obtain $p(y|x_s)$, and to reduce their computational complexity. In this way, our evaluation methods could also be used to assess the quality of explanations for general models other than the linear regression models that we examined in this thesis. Furthermore, specifically regarding the evaluation of explanations produced by our framework, further research is necessary on how to ensure that explanations for each data point are of the same quality despite the explainer model optimising a global criterion.

User-Adaptive Explanations

Based on our previous findings on output encoding in L2X, in chapter 6 we introduce our proposed explanation framework to combine the advantages of the mutual-information-based XAI frameworks L2X and JN to create a method capable of creating local user-adaptive explanations. Furthermore, we evaluate the explanations produced by our method in terms of their quality and the occurrence of output encoding in them.

On the one hand, we find that the explanations produced by solving the objective of our framework exactly are informative about the model outputs for a given user profile and, thus, fulfil our goal of creating useful local and user-adaptive explanations. Regarding output encoding in explanations learned according to our specified implementation, on the other hand, we find a two-fold situation. While for global user profiles the occurrence of output encoding can be characterised by similar rules as for plain L2X explanations, in the case of local user profiles, we only find weak evidence that a similar characterisation is possible. Nonetheless, with the help of the evaluation methods employed to detect output encoding in our experiments, it is possible to ensure that the learned explanations produced by our framework are of satisfactory quality.

Therefore, we conclude that we have tentatively confirmed our research question, although it is evident that much work is still necessary to completely solve it.

The limitations of this contribution are essentially the same as for our work on output encoding, as both use the same setup for their experiments.

1. The main limitation of our contribution is that we only consider linear regression models in our evaluation, as only these are tractable for our evaluation methods. Additionally, our evaluation requires having user profiles for the models to explain, which are much easier to define for linear regression models than for other types of models, which also favours the sole use of linear regression models here.

2. Although user profiles are easy to define for linear regression models, another limitation is that we only evaluate explanations using synthetic user profiles defined by mathematical criteria, in contrast to inputs of real users of such models. Thus, we only cover user-adaptive explanations crafted with user profiles having the precision of mathematical criteria, while explanations for profiles of human users might behave differently, as one can expect human-defined user profiles to be less coherent.

3. We only examine user-adaptive explanations for very few different linear regression models, and we only do so from the global perspective. To obtain a more complete picture of user-adaptive explanations, even just for linear regression models, a more systematic approach for model selection and a more thorough investigation of the explanations produced for individual data points are necessary.

4. The evaluation of this contribution is implemented in 64-bit floating-point precision. But, as all observed phenomena occur in weight ranges well representable in 32-bit precision, we are confident that this does not affect the validity of our results.

The future work necessary to improve our framework for user-adaptive explanations mostly concerns aspects of the personalisation of explanations.

1. The most important objective for further research in this area is to completely answer question Q2 raised in section 6.3, that is, to provide a complete characterisation of the occurrence of output encoding in user-adaptive explanations.

2. In this work, we did not examine how to obtain user profiles from human users. Researching how to create user profiles for real users, particularly concerning the goals of personalised XAI discussed in chapter 2 (e.g. minimising required effort), would exceed the scope of this thesis. Nonetheless, research on obtaining real user profiles is crucial for user-adaptive explanations to fulfil their purpose for human users. Also, conducting a real-life user study on the suitability of user-adaptive explanations is an important target for further research in our framework.

3. There is the possibility to leverage a collaborative approach for obtaining user profiles and learning user-adaptive explanations. As Theorem 1 can be adapted to this setting, one could employ one *user model* to learn user profiles of multiple users by adding a *user embedding* as input to the user and explainer models to separate different users. In scenarios where multiple users need explanations for the same model, this would have several advantages. On the one hand, only one user and explainer model would be required for an entire instance of our framework with multiple users and, on the other hand, it would allow for a quick initiation of new users into an instance of our framework. This is, as it would only be necessary to train the user model once, when initially setting up the instance and, afterwards, to add a new user, it would only be necessary to train a new user embedding for them. Hence, finding a collaborative approach for user-adaptive explanations would provide significant improvements for realistic application scenarios of our framework.

4. So far, in our evaluation of user-adaptive explanations, we examine only user profiles that are truly local or global (i.e. changing with every data point or being completely constant). But we have not yet considered user profiles that are intermediate between these two opposites, e.g. user profiles selecting one set of features in half of the cases and another set in the other half of cases, or a user profile that is almost global and only sporadically selects different features than the usual ones. As user profiles of real users can plausibly be of this kind, it is necessary to also evaluate user-adaptive explanations under these conditions.

5. Other open questions regarding user-adaptive explanations are the use of general user features in user summaries, and whether one can apply similar user-adaption methods also to other XAI methods? Also, in our evaluation of user-adaptive explanations we assume that the user does not commit output encoding in the masks of user summaries. As one can imagine that human users might unconsciously introduce a correlation to model outputs in the masks of user summaries, one should examine how this would affect user-adaptive explanations as well.

8. Conclusion

At the beginning of the present thesis, we set out to create an XAI framework providing personalised explanations with a strong theoretical motivation to help users understand the ever-growing number of AI models that they encounter in their everyday lives. To solve this challenge, we identify the two mutual-information-based explanation methods L2X and JN as promising approaches, both of which have disadvantages that can be overcome by combining them into a single framework. Thus, we postulate the research question to create a mutual-information-based XAI method capable of producing meaningful local user-adaptive explanations.

To answer this research question, first, we identify the problem of output encoding in explanations produced by the L2X method. As we find that output encoding is a serious problem compromising the truthfulness of local mutual-information-based explanations in the form of masked feature vectors, we develop feature-mutual-information as the information-theoretic notion required to create mutual-information-based explanations that are free of output encoding. Furthermore, we introduce evaluation methods allowing us to rigorously evaluate explanations, and we conduct experiments, based on which we can give a precise characterisation of the occurrence of output encoding in L2X explanations. With this groundwork, we establish important prerequisites for creating our own explanation framework, but also make important contributions to our understanding of mutual-information-based explanations and the evaluation thereof.

Based on these insights, we then go on to define our proposed framework for creating local user-adaptive explanations. As we find it mathematically impossible to implement our explanation framework in a way that is devoid of output encoding, we separately evaluate the explanations produced by our framework regarding their quality and the prevalence of output encoding in them. On the one hand, we find in these experiments that user-adaptive explanations produced by solving the explanation objective of our framework exactly are indeed capable of providing the user at hand with information that improves their understanding of the model to be explained. The experiments investigating output encoding in user-adaptive explanations, on the other hand, yield significant results only for the case of global user profiles. In the case of local user profiles, the results are less clear but provide promising clues for further research on them.

Therefore, we have made significant contributions towards answering our research question, especially regarding the theoretical understanding of mutual-information-based explanations, their evaluation, and the creation of local user-adaptive explanations. Nonetheless, many open questions remain for future work, such as the development of a well-founded information theory of masked random vectors, a better understanding of the occurrence and causes of output encoding in explanations, and various possibilities to improve user-adaptive explanations and enhance our understanding of them.

Bibliography

[AB18] Amina Adadi and Mohammed Berrada. Peeking inside the black-box: A survey on explainable artificial intelligence (xai). *IEEE Access*, 6:52138–52160, 2018.

[Axl19] Sheldon Axler. *Measure, Integration & Real Analysis*, volume 282 of *Graduate Texts in Mathematics*. Springer Nature, Cham, 1 edition, 2019.

[BBM+15] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLOS ONE*, 10(7):1–46, 07 2015.

[BCF21] Kayla Boggess, Shenghui Chen, and Lu Feng. Towards personalized explanation of robot path planning via user feedback. *arXiv preprint arXiv:2011.00524*, 2021.

[BXL+21] Seojin Bang, Pengtao Xie, Heewook Lee, Wei Wu, and Eric Xing. Explaining a black-box by using a deep variational information bottleneck approach. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35:11396–11404, 05 2021.

[CBPR21] Cristina Conati, Oswald Barral, Vanessa Putnam, and Lea Rieger. Toward personalized xai: A case study in intelligent tutoring systems. *Artificial Intelligence*, 298:103503, 2021.

[CCN+21] Davide Calvaresi, Giovanni Ciatto, Amro Najjar, Reyhan Aydoğan, Leon Van der Torre, Andrea Omicini, and Michael Schumacher. Expectation: Personalized explainable artificial intelligence for decentralized agents with heterogeneous knowledge. In Davide Calvaresi, Amro Najjar, Michael Winikoff, and Kary Främling, editors, *Explainable and Transparent AI and Multi-Agent Systems*, pages 331–343, Cham, 2021. Springer International Publishing.

[CMG22] Johannes Schneider Christian Meske, Enrico Bunde and Martin Gersch. Explainable artificial intelligence: Objectives, stakeholders, and future research opportunities. *Information Systems Management*, 39(1):53–63, 2022.

[CSWJ18] Jianbo Chen, Le Song, Martin Wainwright, and Michael Jordan. Learning to explain: An information-theoretic perspective on model interpretation.

In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 883–892. PMLR, 10–15 Jul 2018.

[CT12] Thomas M Cover and Joy A Thomas. *Elements of Information Theory*. John Wiley & Sons, Ltd, 2012.

[GA19] David Gunning and David Aha. Darpa's explainable artificial intelligence (xai) program. *AI Magazine*, 40(2):44–58, Jun. 2019.

[GFG+22] Shijie Geng, Zuohui Fu, Yingqiang Ge, Lei Li, Gerard de Melo, and Yongfeng Zhang. Improving personalized explanation generation through visualization. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 244–255, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[GPAM+14] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014.

[GWZ+19] Chaoyu Guan, Xiting Wang, Quanshi Zhang, Runjin Chen, Di He, and Xing Xie. Towards a deep and unified understanding of deep neural models in NLP. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2454–2463. PMLR, 09–15 Jun 2019.

[HP22] Philipp Hacker and Jan-Hendrik Passoth. *Varieties of AI Explanations Under the Law. From the GDPR to the AIA, and Beyond*, pages 343–373. Springer International Publishing, Cham, 2022.

[HSM+22] Andreas Holzinger, Anna Saranti, Christoph Molnar, Przemyslaw Biecek, and Wojciech Samek. *Explainable AI Methods - A Brief Overview*, pages 13–38. Springer International Publishing, Cham, 2022.

[IEGA21] Sheikh Rabiul Islam, William Eberle, Sheikh Khaled Ghafoor, and Mohiuddin Ahmed. Explainable artificial intelligence approaches: A survey. *ArXiv*, abs/2101.09429, 2021.

[JGP17] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In *International Conference on Learning Representations*, 2017.

[JN20] Alexander Jung and Pedro H. J. Nardelli. An information-theoretic approach to personalized explainable machine learning. *IEEE Signal Processing Letters*, 27:825–829, 2020.

[JSAR21] Neil Jethani, Mukund Sudarshan, Yindalon Aphinyanaphongs, and Rajesh Ranganath. Have we learned to explain?: How interpretability methods can learn to encode predictions in their interpretations. In Arindam Banerjee and Kenji Fukumizu, editors, *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pages 1459–1467. PMLR, 13–15 Apr 2021.

[KHSMP26] Melanija Kraljevska, Kateřina Hlaváčková-Schindler, Lukas Miklautz, and Claudia Plant. Eeg-based classification in psychiatry using motif discovery. *Neuroscience Informatics*, 6(1):100242, 2026.

[KSP+19] Pigi Kouki, James Schaffer, Jay Pujara, John O'Donovan, and Lise Getoor. Personalized explanations for hybrid recommender systems. In *Proceedings of the 24th International Conference on Intelligent User Interfaces*, IUI '19, page 379–390, New York, NY, USA, 2019. Association for Computing Machinery.

[LL17] Scott M. Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 4768–4777, Red Hook, NY, USA, 2017. Curran Associates Inc.

[LV22] Q. Vera Liao and Kush R. Varshney. Human-centered explainable ai (xai): From algorithms to user experiences. *arXiv preprint arXiv:2110.10790*, 2022.

[LZC21] Lei Li, Yongfeng Zhang, and Li Chen. Personalized transformer for explainable recommendation. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4947–4957, Online, August 2021. Association for Computational Linguistics.

[MCV22] Millecamp Martijn, Cristina Conati, and Katrien Verbert. "knowing me, knowing you": personalized explanations for a music recommender system. *User Modeling and User-Adapted Interaction*, 32(1):215–252, Apr 2022.

[MKH+22] Christoph Molnar, Gunnar König, Julia Herbinger, Timo Freiesleben, Susanne Dandl, Christian A. Scholbeck, Giuseppe Casalicchio, Moritz Grosse-Wentrup, and Bernd Bischl. *General Pitfalls of Model-Agnostic Interpretation Methods for Machine Learning Models*, pages 39–68. Springer International Publishing, Cham, 2022.

[MLH+18] James McInerney, Benjamin Lacker, Samantha Hansen, Karl Higley, Hugues Bouchard, Alois Gruson, and Rishabh Mehrotra. Explore, exploit, and explain: personalizing explainable recommendations with bandits. In *Proceedings of the 12th ACM Conference on Recommender Systems*, RecSys '18, page 31–39, New York, NY, USA, 2018. Association for Computing Machinery.

[MMT17] Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. In *International Conference on Learning Representations*, 2017.

[NCZ+24] Robert Nimmo, Marios Constantinides, Ke Zhou, Daniele Quercia, and Simone Stumpf. User characteristics in explainable ai: The rabbit hole of personalization? In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, CHI '24, New York, NY, USA, 2024. Association for Computing Machinery.

[PBGN23] Rajvardhan Patil, Sorio Boit, Venkat Gudivada, and Jagadeesh Nandigam. A survey of text representation and embedding techniques in nlp. *IEEE Access*, 11:36120–36146, 2023.

[PNR24] Aahlad Manas Puli, Nhi Nguyen, and Rajesh Ranganath. Explanations that reveal all through the definition of encoding. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[PW25] Yury Polyanskiy and Yihong Wu. *Information Theory: From Coding to Learning*. Cambridge University Press, 2025.

[RRF+22] Jonas Rebstadt, Florian Remark, Philipp Fukas, Pascal Meier, and Oliver Thomas. Towards personalized explanations for ai systems: Designing a role model for explainable ai in auditing. In *17th International Conference on Wirtschaftsinformatik*, Nürnberg, Germany, 2022.

[RSG16] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should i trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 1135–1144, New York, NY, USA, 2016. Association for Computing Machinery.

[RTL19] Mireia Ribera Turró and Agata Lapedriza. Can we do better explanations? a proposal of user-centered explainable ai. In *Joint Proceedings of the ACM IUI 2019 Workshops*, Los Angeles, USA, 03 2019. ACM, New York, NY, USA.

[SH19] Johannes Schneider and Joshua Peter Handali. Personalized explanation for machine learning: A conceptualization. In *Proceedings of the 27th European*

Conference on Information Systems (ECIS), Stockholm & Uppsala, Sweden, June 2019.

[SKMT25] Timothée Schmude, Laura Koesten, Torsten Möller, and Sebastian Tschiatschek. Information that matters: Exploring information needs of people affected by algorithmic decisions. *International Journal of Human-Computer Studies*, 193:103380, 2025.

[SO23] Waddah Saeed and Christian Omlin. Explainable ai (xai): A systematic meta-survey of current challenges and future opportunities. *Knowledge-Based Systems*, 263:110273, 2023.

[SSK21] Utkarsh Soni, Sarath Sreedharan, and Subbarao Kambhampati. Not all users are the same: Providing personalized explanations for sequential decision making problems. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6240–6247, 2021.

[STSG24] Andrew Silva, Pradyumna Tambwekar, Mariah Schrum, and Matthew Gombolay. Towards balancing preference and performance through adaptive personalized explainability. In *Proceedings of the 2024 ACM/IEEE International Conference on Human-Robot Interaction*, HRI '24, page 658–668, New York, NY, USA, 2024. Association for Computing Machinery.

[TG21] Erico Tjoa and Cuntai Guan. A survey on explainable artificial intelligence (xai): Toward medical xai. *IEEE Transactions on Neural Networks and Learning Systems*, 32(11):4793–4813, 2021.

[Tib96] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288, 1996.

[WCH23] Patrick Weber, K. Valerie Carl, and Oliver Hinz. Applications of explainable artificial intelligence in finance—a systematic review of finance, information systems, and computer science literature. *Management Review Quarterly*, Feb 2023.

[XUD+19] Feiyu Xu, Hans Uszkoreit, Yangzhou Du, Wei Fan, Dongyan Zhao, and Jun Zhu. Explainable ai: A brief survey on history, research areas, approaches and challenges. In Jie Tang, Min-Yen Kan, Dongyan Zhao, Sujian Li, and Hongying Zan, editors, *Natural Language Processing and Chinese Computing*, pages 563–574, Cham, 2019. Springer International Publishing.

[ZYK+23] Maede Zolanvari, Zebo Yang, Khaled Khan, Raj Jain, and Nader Meskin. Trust xai: Model-agnostic explanations for ai with a case study on iiot security. *IEEE Internet of Things Journal*, 10(4):2967–2978, 2023.

List of Tables

List of Figures

Acronyms

AI Artificial Intelligence. iii, v, 1, 2, 5–7, 9, 39, 63

JN Jung and Nardelli. 2–4, 13–15, 17, 19, 39, 61, 63

L2X Learning to Explain. iii, v, vii, 2–4, 12–15, 19–21, 23, 25, 27, 29–31, 33–39, 42–45, 58–61, 63, 73, 75, 85, 87–89, 91, 93, 95

ML Machine Learning. 1, 2, 5, 7, 8, 11–13, 19, 36, 37

XAI Explainable Artificial Intelligence. iii, v, vii, 1–12, 14, 15, 19, 37, 39, 60–63

A. Proofs

A.1. Logarithmic Approximation of the Binomial Coefficient

Lemma 1. *Let $d, k \in \mathbb{N}$ with $0 < k < d$, and let $H : [0,1] \to \mathbb{R}$ denote the binary entropy $H(x) = -x\log(x) - (1-x)\log(1-x)$. Then, for the binomial coefficient $\binom{d}{k}$ it holds that*

$$\log \binom{d}{k} = dH(k/d) + O(\log d). \tag{A.1}$$

Proof. By Lemma 17.5.1 of [CT12] we have

$$\frac{1}{2}\log\left(\frac{d}{8k(d-k)}\right) \leq \log\binom{d}{k} - dH(k/d) \leq \frac{1}{2}\log\left(\frac{d}{\pi k(d-k)}\right). \tag{A.2}$$

Therefore,

$$-1.5 \leq \log\binom{d}{k} - dH(k/d) - \frac{1}{2}\log\left(\frac{d}{k(d-k)}\right) \leq -\log\sqrt{\pi} \tag{A.3}$$

and hence

$$\log\binom{d}{k} = dH(k/d) + \frac{1}{2}\log\left(\frac{d}{k(d-k)}\right) + O(1). \tag{A.4}$$

Further, as k only assumes integer values $1 \leq k \leq d-1$, we find that

$$d - 1 \leq k(d-k) \leq \frac{d^2}{4} \tag{A.5}$$

and thus

$$1 \geq \log\left(\frac{d}{k(d-k)}\right) \geq 2 - \log d \tag{A.6}$$

from which the result follows. □

A.2. Proposition 1

Proof.

$$\begin{aligned}
I(\mu(X,S),Y) &= \int_{X,S,Y} \log\frac{p(x_s,s,y)}{p(x_s,s)p(y)} p(x,s,y)dxsy && \text{(A.7)}\\
&= \int_{X,S,Y} \log\frac{p(x_s,y)p(s|x_s,y)p(y|x_s)}{p(x_s)p(s|x_s)p(y)p(y|x_s)} p(x,s,y)dxsy && \text{(A.8)}\\
&= \int_{X,S,Y} \left(\log\frac{p(x_s,y)}{p(x_s)p(y)} + \log\frac{p(s,y|x_s)}{p(s|x_s)p(y|x_s)}\right) p(x,s,y)dxsy && \text{(A.9)}\\
&= I(X_S,Y) + I(S,Y|X_S) && \text{(A.10)}
\end{aligned}$$

□

A.3. Theorem 1

Proof. Let us begin by showing that $\mathcal{E}^*$ is an optimal explainer model according to the global criterion (6.1). By definition of $\mathcal{E}^*$ through criterion (6.2) it holds for all $x \in \mathbb{R}$ and explainer models $\mathcal{E}$ that

$$\mathbb{E}_{Y,U|X}[\log p(Y|X_{\mathcal{E}(x)} = x_{\mathcal{E}(x)}, U)|X = x] \leq \mathbb{E}_{Y,U|X}[\log p(Y|X_{\mathcal{E}^*(x)} = x_{\mathcal{E}^*(x)}, U)|X = x]. \tag{A.11}$$

Taking the expectation over X and subtracting $\mathbb{E}[\log p(Y|U)]$ on both sides of this inequality yields

$$I(X_{\mathcal{E}(X)}, Y|U) \leq I(X_{\mathcal{E}^*(X)}, Y|U) \tag{A.12}$$

which is just the desired result.

Conversely, let us prove that any optimal explainer model $\widehat{\mathcal{E}}$ coincides almost surely with $\mathcal{E}^*$ with respect to the distribution of X. Without loss of generality, we may assume that $\widehat{\mathcal{E}}$ and $\mathcal{E}^*$ are maximally coinciding; that is, for all $x \in \mathbb{R}^d$ where criterion (6.2) does not uniquely determine $\mathcal{E}^*(x)$ and $\widehat{\mathcal{E}}(x)$ is among the possible values, we may assume that $\mathcal{E}^*(x) = \widehat{\mathcal{E}}(x)$. Now, let us assume that there exists a set $M \subset \mathbb{R}^d$ of positive measure with respect to the data-generating distribution p_X on which $\widehat{\mathcal{E}}$ and $\mathcal{E}^*$ take different values. Then, for any $x \in M$ we find according to the definition of $\mathcal{E}^*$ that

$$\mathbb{E}_{Y,U|X}[\log p(Y|X_{\widehat{\mathcal{E}}(x)} = x_{\widehat{\mathcal{E}}(x)}, U)|X = x] < \mathbb{E}_{Y,U|X}[\log p(Y|X_{\mathcal{E}^*(x)} = x_{\mathcal{E}^*(x)}, U)|X = x], \tag{A.13}$$

while for any $x \notin M$ it holds

$$\mathbb{E}_{Y,U|X}[\log p(Y|X_{\widehat{\mathcal{E}}(x)} = x_{\widehat{\mathcal{E}}(x)}, U)|X = x] \leq \mathbb{E}_{Y,U|X}[\log p(Y|X_{\mathcal{E}^*(x)} = x_{\mathcal{E}^*(x)}, U)|X = x]. \tag{A.14}$$

Thus, by taking the expectation over X and subtracting $\mathbb{E}[\log p(Y|U)]$ on both sides of the above two inequalities, since M has positive measure with respect to the distribution of X, we find that

$$I(X_{\widehat{\mathcal{E}}(X)}, Y|U) < I(X_{\mathcal{E}^*(X)}, Y|U)$$

contradicting the fact that $\widehat{\mathcal{E}}$ is an optimal solution to criterion (6.1). Thus, we have to reject our assumption that M has positive measure, which in turn proves the claim. □

A.4. Proposition 2

Proof. Consider the feature mutual information $I(X_S, Y|U)$ of criterion (6.1), which one can decompose as

$$I(X_S, Y|U) = \mathbb{E}\left[\log \frac{p(X_S, Y|U)}{p(X_S|U)p(Y|U)}\right] = \mathbb{E}\left[\log \frac{p(Y|X_S, U)}{p(Y|U)}\right] =$$
$$= \mathbb{E}\Big[\log p(Y|X_S, U)\Big] + \text{const.} = \mathbb{E}_S \mathbb{E}_{X_S,U} \mathbb{E}_{Y|X_S,U}\Big[\log p(Y|X_S, U)\Big] + \text{const.} \quad \text{(A.15)}$$

where the constant term is to be understood with respect to the explanation masks S as it only depends on Y and U. From this decomposition we can see two things: On the one hand, we see that maximising $\mathbb{E}[\log p(Y|X_S, U)]$ is equivalent to maximising $I(X_S, Y|U)$. On the other hand, we find that $\mathbb{E}[\log p(Y|X_S, U)]$ can be further decomposed into an expectation with respect to $p_{Y|U,X_S}$ over $\log p_{Y|U,X_S}$, which one can use to substitute $p_{Y|U,X_S}$ with the variational models $\mathcal{Q}$. To see this, for a fixed explanation mask S we consider the Kullback-Leibler divergence $D(p_{Y|X_S,U} \,|\, q_S)$ of $p_{Y|U,X_S}$ and q_S, and find

$$0 \leq D(p_{Y|X_S,U} \,|\, q_S) \quad \text{(A.16)}$$
$$0 \leq \mathbb{E}_{Y|X_S,U}\Big[\log p(Y|X_S, U) - \log q_S(Y|X_S, U)\Big] \quad \text{(A.17)}$$
$$\mathbb{E}_{Y|X_S,U}\Big[\log q_S(Y|X_S, U)\Big] \leq \mathbb{E}_{Y|X_S,U}\Big[\log p(Y|X_S, U)\Big]. \quad \text{(A.18)}$$

As this holds for all explanation masks S, it follows that

$$\mathbb{E}\Big[\log q_S(Y|X_S, U)\Big] \leq \mathbb{E}\Big[\log p(Y|X_S, U)\Big]. \quad \text{(A.19)}$$

Thus, we find that, in expectation, $\mathcal{Q}$ yields a lower bound to $I(X_S, Y|U)$ (up to a constant with respect to S), which attains equality if and only if q_S and $p_{Y|X_S,U}$ are equal in distribution for every S. Therefore, optimising $\mathcal{E}$ and $\mathcal{Q}$ over criterion (6.5) is sufficient to optimise $\mathcal{E}$ to fulfil criterion (6.1). □

B. Mutual Information of Local and Global Explanations

In this appendix, we use the techniques introduced in section 5.4 to demonstrate through numerical experiments that the feature mutual information of mutual-information-maximising local explanations for linear regression models of various sizes is closely bounded to the mutual information of mutual-information-maximising global explanations of the same model and size.

To this end, we choose model dimensions $d = 2, 4, 8, 16$ and for each d randomly choose 1000 linear regression models as specified in Example 2, i.e. having squared weights summing to 1, which means that for d-dimensional standard normal input data the model outputs are standard normal as well. For each of these models, we then determine the feature mutual information of the model output with mutual-information-maximising local and global explanations of sizes $k = 1, \ldots, d/2, 3d/4$.

What we call local explanations here are the same as the exact explanations of section 5.4; the global explanations are, as discussed in Example 2, the subvectors of the input data maximising the mutual information with the model output.

The results of these experiments can be seen in Figure B.1, and summary statistics of the observed differences in feature mutual information for each combination of d and k can be seen in Table B.1. Most importantly, in these results, one can see that depending on the model dimension and explanation size, it is always the case that the (feature) mutual information of local explanations only exceeds the mutual information of global explanations by an amount of about 0.5 to 0.7 bits. Furthermore, due to the fact that the mutual information of global explanations is essentially just a measure of the concentration of the weights of a linear regression model, in Figure B.1 one can observe the effects that weight concentration has on the mutual information of explanations. As models with more equal weights (low global mutual information, to the left) allow for more opportunities for the local explanations to exhibit their flexibility than models with more concentrated weights (high global mutual information, to the right) where local and global explanations more often coincide, one can observe how the differences in these metrics gradually decline across the plots. The only case where this effect seems less pronounced is in the case of $d = 16$; however, this can be explained by the curse of dimensionality as models with more concentrated weights become more scarce in high dimensions and, thus, the respective areas of the 15-dimensional unit sphere were not thoroughly covered by our sampling scheme. The code to reproduce these experiments can be found in the notebook [1].

[1] https://github.com/LukasKarner/IT4PXAI/blob/main/mi_local_v_global.ipynb

d	k	min	mean ± std	max
2	1	-0.110	0.168 ± 0.128	0.470
4	1	0.023	0.310 ± 0.113	0.525
	2	-0.035	0.308 ± 0.138	0.588
	3	-0.049	0.203 ± 0.141	0.574
8	1	0.021	0.376 ± 0.071	0.514
	2	0.136	0.461 ± 0.082	0.616
	4	0.060	0.449 ± 0.110	0.656
	6	-0.007	0.340 ± 0.143	0.644
16	1	0.193	0.391 ± 0.044	0.493
	2	0.289	0.512 ± 0.045	0.613
	4	0.323	0.578 ± 0.046	0.681
	8	0.298	0.569 ± 0.061	0.682
	12	0.034	0.475 ± 0.105	0.672

Table B.1.: Summary statistics of the differences in (feature) mutual information of local and global explanations resulting from numerical experiments on linear regression models with dimension d and explanation size k. For each combination of d and k, 1000 models were randomly sampled uniformly from the $d-1$-dimensional unit sphere. Negative values for the minima are attributed to sampling error in the estimation of the mutual information of local explanations.

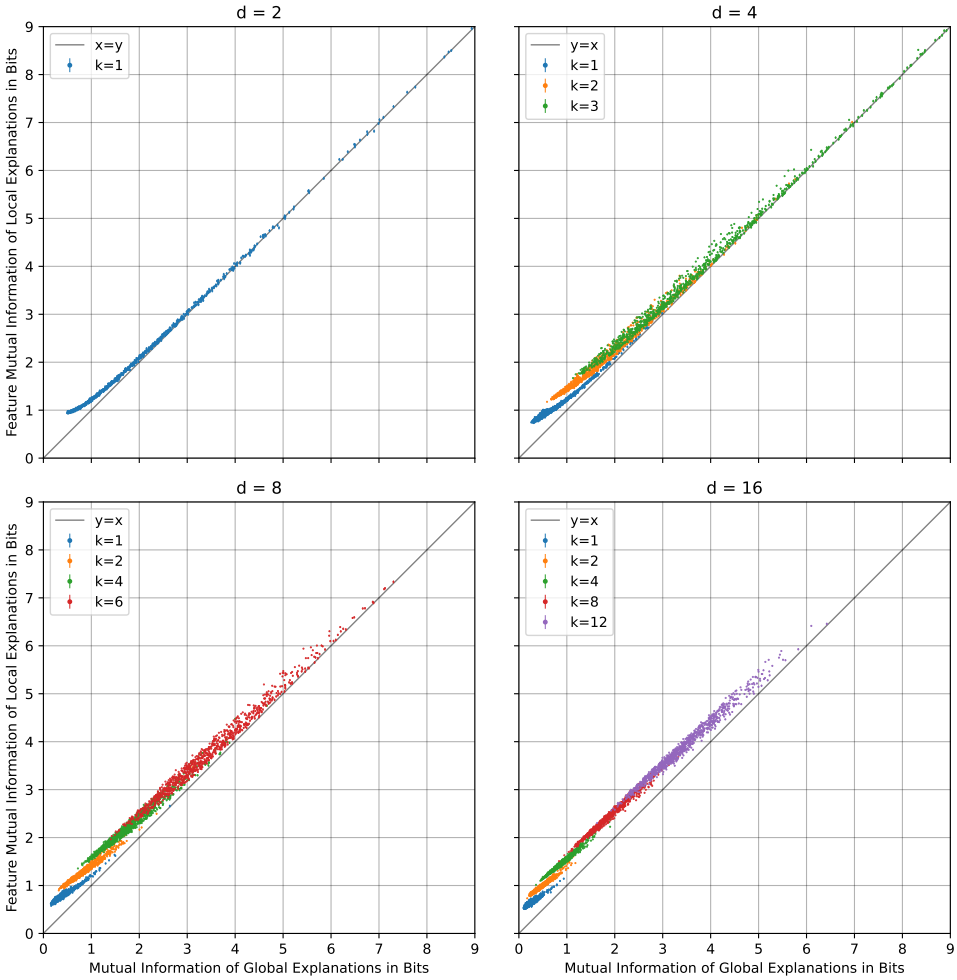


Figure B.1.: Scatter plots showing the (feature) mutual information $I(\mu(X,S),Y)$ of global and local explanations, each with mask size k, of 1000 linear regression models of dimension d. The weights of the models are drawn randomly from the uniform distribution on the $d-1$-dimensional unit sphere, the input data to the models follow the standard normal distribution in d dimensions, the global explanations were selected to maximise the mutual information for the given mask size, and the local explanations were selected according to the L2X criterion. The mutual information of the global explanations was calculated analytically, while a Monte Carlo approach was used for the local explanations, where vertical error bars indicate the standard deviation of the estimate.

C. Experiment Details

This appendix provides further information on the evaluation methods and experiments described in chapters 5 and 6.

C.1. Technical Details

The code to reproduce the experiments of section 5.5 can be found in the notebook [1], and for sections 6.4 and 6.5 it can be found in the notebook [2].

Data

For our experiments, we use a synthetic dataset of 16-dimensions following a standard normal distribution. The training and test sets have 100.000 and 1000 samples each.

Exact Explanations

We compute exact L2X explanations by directly solving criterion (5.3) for linear regression models. To do this for a given data point x and model $x \mapsto \sum_i x_i \alpha_i = y$, we compute likelihoods $p(y|x_s)$ for all possible feature subsets s of the explanation mask size. This can be done as for standard normally distributed input data $X \sim \mathcal{N}(0, 1)$, it holds

$$Y|x_s \sim \mathcal{N}\left(\sum_{i \in s} x_i \alpha_i, \sqrt{\sum_{i \notin s} \alpha_i^2}\right) \tag{C.1}$$

for the partially conditioned model output. Given these likelihood values, the feature subset associated with the maximal value is the exact explanation.

To compute exact user-adaptive explanations, the same procedure is applied to objective (6.2). To do so, we note that user summaries and data have the same features and, hence, can be identified. For example, consider a data point x with explanation mask s and let $\mathcal{U}(x) = u = x_{s'}$ for some index subset s', then it holds $p(y|x_s, u) = p(y|x_{s \cup s'})$. With this, one can compute explanations from likelihoods as described above.

[1] https://github.com/LukasKarner/IT4PXAI/blob/main/output_encoding.ipynb
[2] https://github.com/LukasKarner/IT4PXAI/blob/main/user_explanations.ipynb

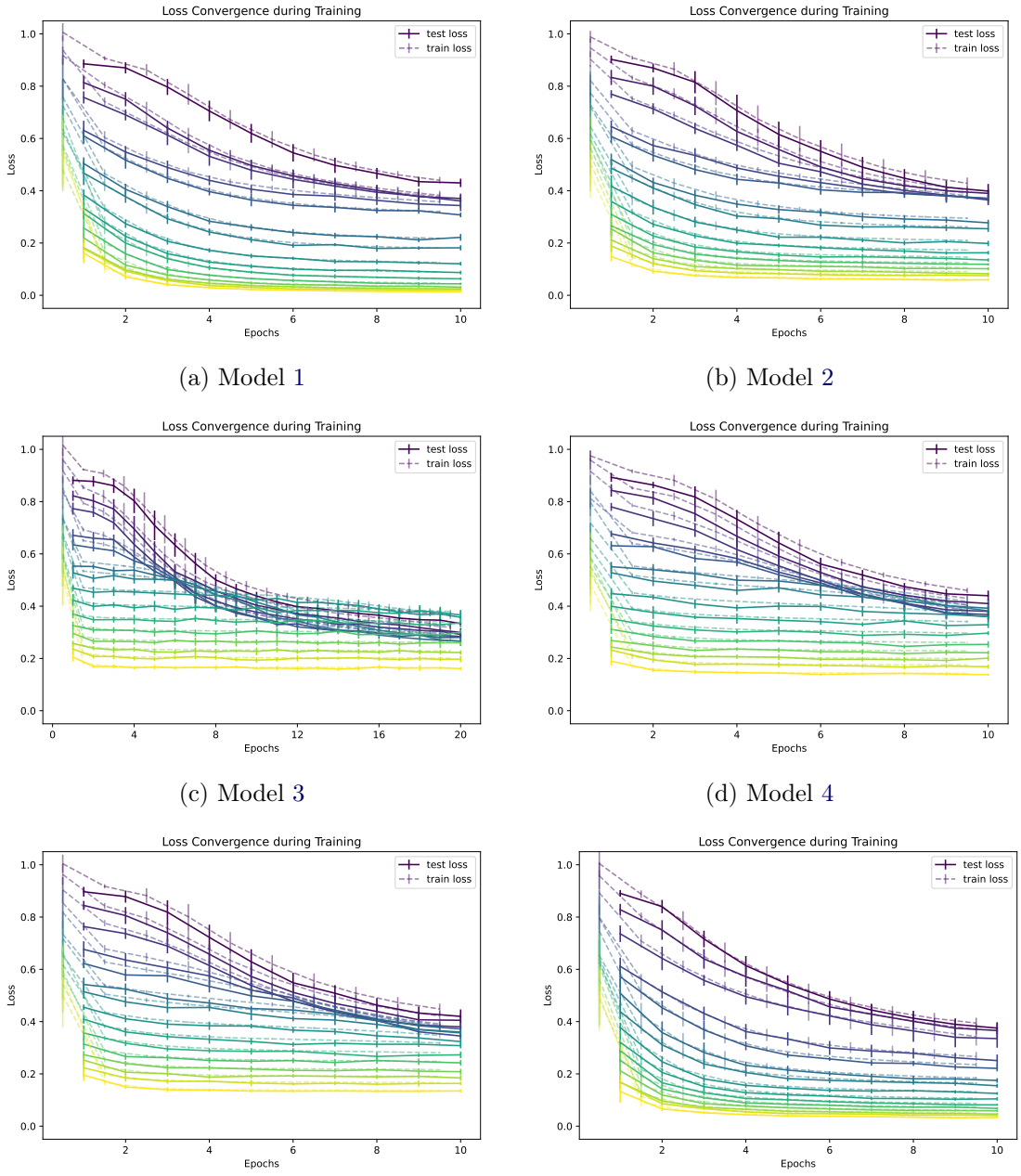
Learned Explanations

To obtain learned L2X explanations, we mostly follow the implementation proposed by [CSWJ18], and for user-adaptive explanations we follow additionally the modifications to L2X discussed in section 6.2. As explainer models, we use neural networks with 1 hidden layer of width 100 and SELU activation, the same holds for regressor models, but they are also equipped with a BatchNorm layer after the activation. All models are trained with mean-square-error as loss function due to the linear regression objective of the models to be explained, and the training was conducted using the Adam optimiser with a learning rate of $3 \cdot 10^{-4}$. The learning rate was selected based on trials with different linear regression models and explanation mask sizes, where it performed well on all the tested cases, while other learning rates had less consistent performance.

Determining the training duration of L2X explanations is not straightforward, as in [CSWJ18] there are no precise statements regarding the required duration of training. Thus, we estimate an appropriate training duration based on the code implementing the experiments presented in [CSWJ18]. There, we find the following training configurations, each using a learning rate of 10^{-3}: 1. Explanations for complex synthetic datasets of 100.000 samples being trained for 1 epoch, 2. explanations for word-level sentiment analysis on a dataset of 25.000 pieces of text being trained for 5 epochs, and 3. explanations for sentence-level sentiment analysis on a dataset of 25.000 pieces of text being trained for 10 epochs. As all 3 of these tasks are arguably more difficult to model than linear regression, our dataset has 100.000 samples, and our learning rate is about 3 times smaller than the one used by [CSWJ18], we deduce that training our explainer models for approximately 10 epochs should satisfy the training requirements intended by [CSWJ18]. Nonetheless, we ensure that the training of L2X explanations used in our experiments lasts long enough so that the models show convergent behaviour, which is indeed the case after 10 epochs for all models but Model 3, which is trained for 20 epochs. The training and test loss curves for the training process of the L2X models discussed in section 5.5 can be seen in Figure C.1.

Gumbel-Softmax Sampling

The only aspect in the implementation of learned explanations in which we significantly diverge from the implementation of [CSWJ18] is the number of draws to make from the Gumbel distribution in the application of the Gumbel-Softmax-trick. The problem there is that, during the joint training of the explainer and regressor models, the explanation masks of size k are determined by randomly drawing k times with replacement from the set of all features. This operation, however, is highly biased towards creating explanation masks of sizes much smaller than k as, in general, drawing k times from a finite set of size not much larger than k will not yield k distinct features. As our experiments critically depend on accurate explanation mask sizes, we correct for this bias in our implementation. As it is a mathematically difficult problem to determine the number of required draws,



(a) Model 1

(b) Model 2

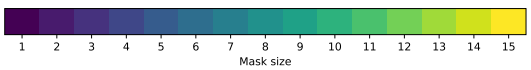
(c) Model 3

(d) Model 4

(e) Model 5

(f) Model 6

Figure C.1.: Training and test losses of L2X explanations over their training on Models 1 to 6. Training losses were recorded during epochs while test losses were recorded after each epoch. Vertical bars show standard deviations observed over 10 trials. Colours show explanation mask sizes as indicated below.



we cannot solve this problem exactly. Instead, we rely on the quadratic model

$$n = \left\lfloor k \left(1 + \frac{\gamma k}{d}\right) + 0.5 \right\rfloor \tag{C.2}$$

with n the corrected number of draws, d the model dimension, and γ a tuning factor as an approximate solution to this problem. The intention behind it is that, for small values of k compared to d, it is more likely to draw k distinct features and, thus, less oversampling is needed, while it becomes increasingly necessary for larger k. The tuning factor γ we set such that for $k \approx d/2$ drawing n times with replacement from d features yields on average k distinct elements. For $d = 16$, we numerically determined that drawing $n = 11$ times yields, on average, about 8 distinct features, which requires the factor to be $\gamma = 0.75$. With this approach, we make sure that our approximate solution produces exact results at the most sensitive explanation mask sizes around $k \approx d/2$. Furthermore, we verified numerically that it also provides excellent estimates for all explanation mask sizes up to $k = 11$. For $k \geq 12$, the quadratic model starts to lag behind the actual required number of draws. However, as this value range is less relevant for our experiments, this is an acceptable tradeoff.

For the experiments with user-adaptive explanations conducted in chapter 6 it is necessary to use different tuning factors γ as conditioning on user summaries with sizes $l = 2, 4, 8$ means that we have to sample explanation features out of a reduced number of 14, 12 or 8 features. Thus, we use $\gamma = 5/7$ to sample out of 14 features, $\gamma = 2/3$ to sample out of 12 features, and $\gamma = 1/2$ to sample out of 8 features.

Output Prediction Losses

To assess output prediction losses for a certain explanation type, we train and evaluate regressor models using either complete explanations or just the explanation masks of said explanations. The used regressor models and training parameters are the same as described in the previous paragraph for the training of learned explanations.

Feature Mutual Information

To compute the feature mutual information of explanations, we note that it holds

$$I(X_S, Y) = \mathbb{E} \log \frac{p(X_S, Y)}{p(X_S)p(Y)} = \mathbb{E} \log \frac{p(Y|X_S)}{p(Y)} = \mathbb{E} \log p(Y|X_S) + H(Y). \tag{C.3}$$

The term on the rightmost side consists, on the one hand, of the entropy of the model output Y, thus given by $H(Y) = \frac{1}{2} \log 2\pi e$ for the standard normally distributed data used in our experiments. On the other hand, the term consists of the expected log-likelihood of model outputs given the feature values of explanations. Therefore, we can use equation (C.1) to access conditional densities $p(y|x_s)$ of linear regression models for all feature subsets s and compute the feature mutual information of explanations in a Monte-Carlo fashion.

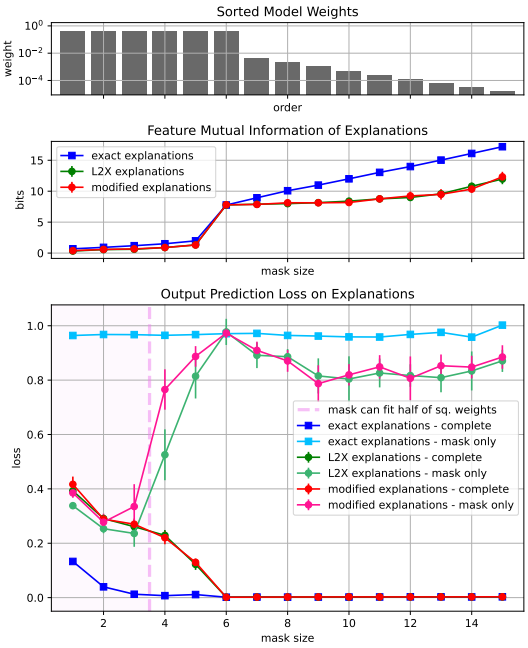
C.2. Mitigation of Output Encoding

Figures C.3 and C.4 show the results of the experiments in section 5.5 with the modifications of L2X, passive and active mitigation of output encoding, proposed to prevent output encoding in explanations. The implementations of these methods are given by the classes `L2XPassiveCounterEncoding` and `L2XActiveCounterEncoding` in the script [3].

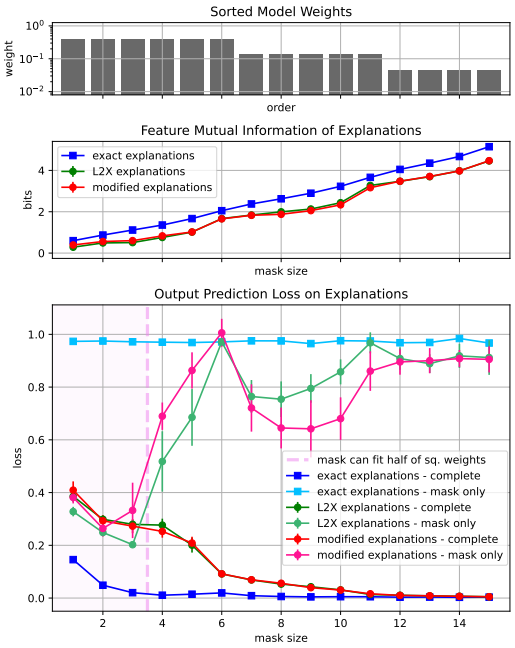
As for the technical details of these experiments, the explanations with passive mitigation of output encoding are created in the same way, except for one modification, as it is described for the learned explanations in appendix C.1. This sole modification is that regressor and explainer models are not trained jointly, but sequentially in a two-step procedure. In the first step, to train the regressor model, the explainer model and Gumbel-Softmax-trick are replaced by a Dropout layer appropriate for the chosen explanation mask size. In the second step, the weights of the regressor model are frozen, as only the explainer model is trained with the regressor model acting as part of its loss function. Notably, up to the used dropout probability, this procedure is identical with the approach to mitigate output encoding proposed by [JSAR21].

The additional modifications implemented for active mitigation of encoding consist of the addition of the detector model and its interaction with the explainer model during the second training step. A detector model is architecturally identical to the regressor models used for computing mask-only output prediction losses, as described in appendix C.1, and also its training parameters are the same. The interaction of explainer and detector models in each of the update steps of their simultaneous training is as follows: At first, the explainer model produces an explanation, the mask of which is then passed to the detector model to execute a training step on it. Then, the explanation is used by the frozen regressor model to compute the loss for the training of the explainer model. Additionally, the explanation mask is also used by the momentarily frozen detector model to compute an additional mask-loss as regulariser for the training of the explainer model. The mask-loss is computed with a target randomly drawn from the model output distribution so that the explainer model learns not to associate model outputs with explanation masks. Finally, the explainer model executes an update step based on the sum of these two losses, where the loss contributed by the detector model is multiplied by a factor of 0.03.

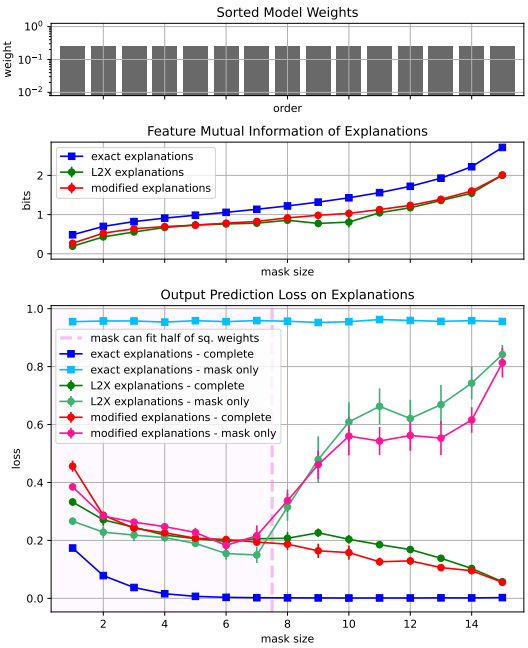
[3] https://github.com/LukasKarner/IT4PXAI/blob/main/explanations.py



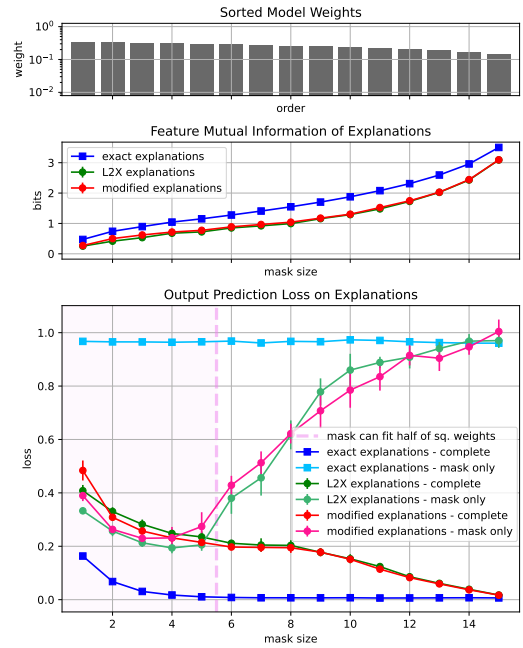
(a) Model 1



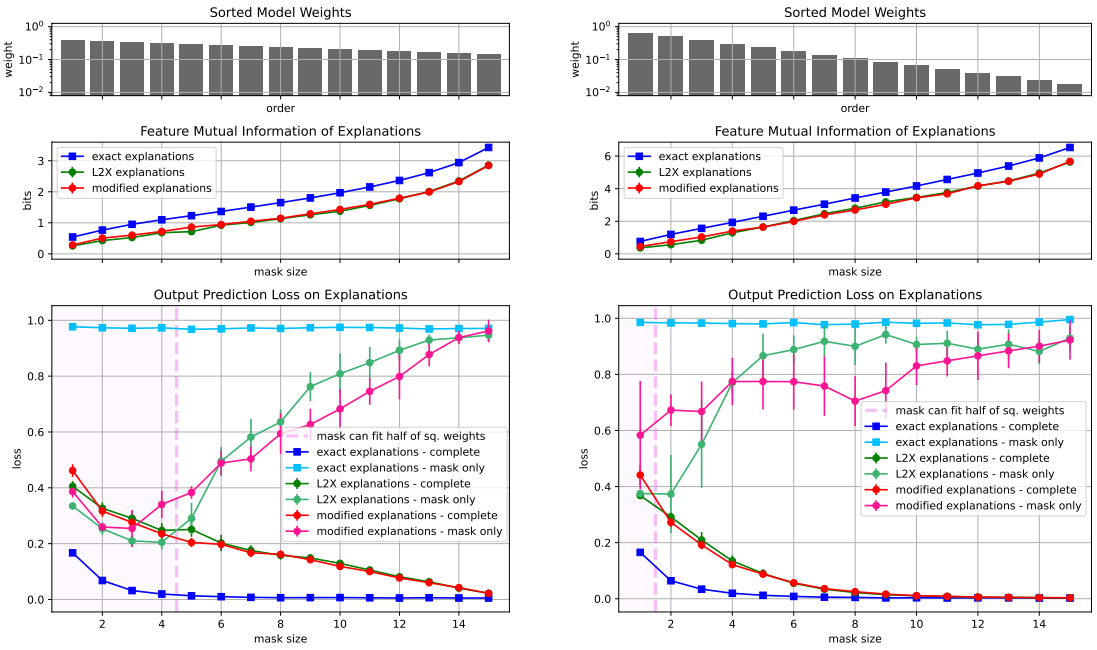
(b) Model 2



(c) Model 3



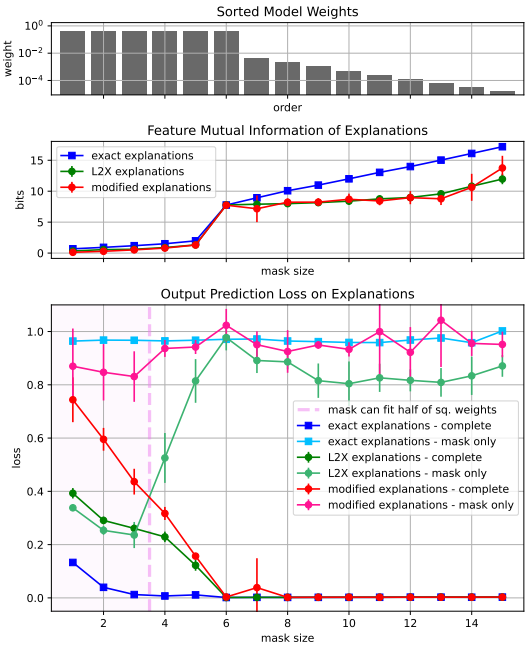
(d) Model 4



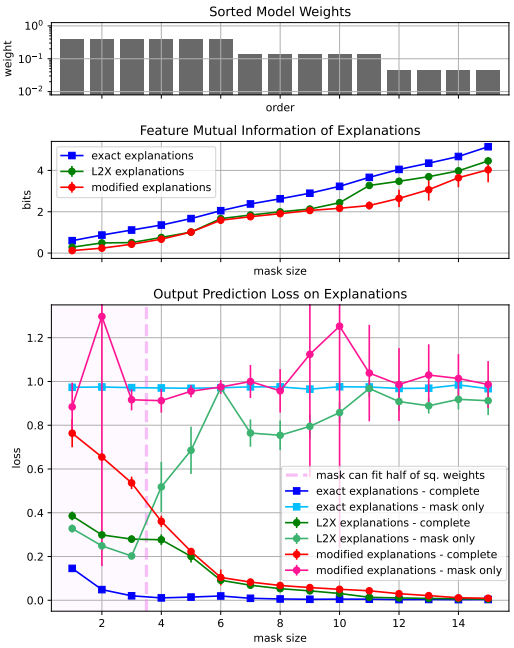
(e) Model 5

(f) Model 6

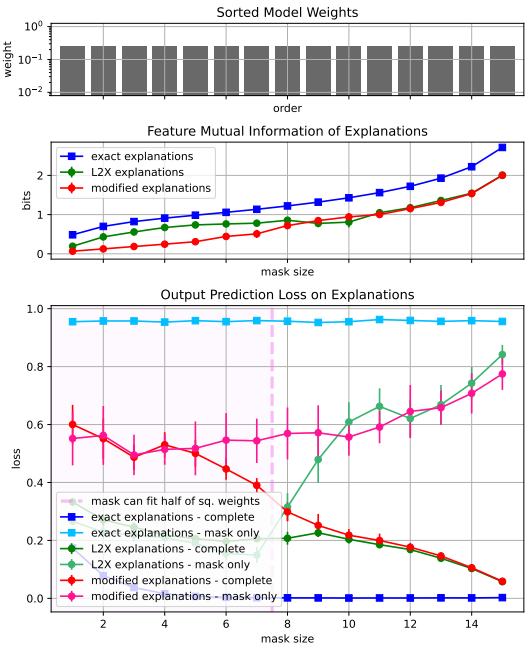
Figure C.3.: Plots showing the results of our experiments on passive mitigation of output encoding. The used models and the displayed results for exact and learned explanations are the same as in Figure 5.2. Additionally to this, in this figure also the results obtained for L2X explanations with passive mitigation of output encoding are shown labelled as *modified explanations*. In the graphs belonging to learned and modified explanations, the vertical error bars show the standard deviation of the results observed in 10 randomly initiated and trained explainer models.



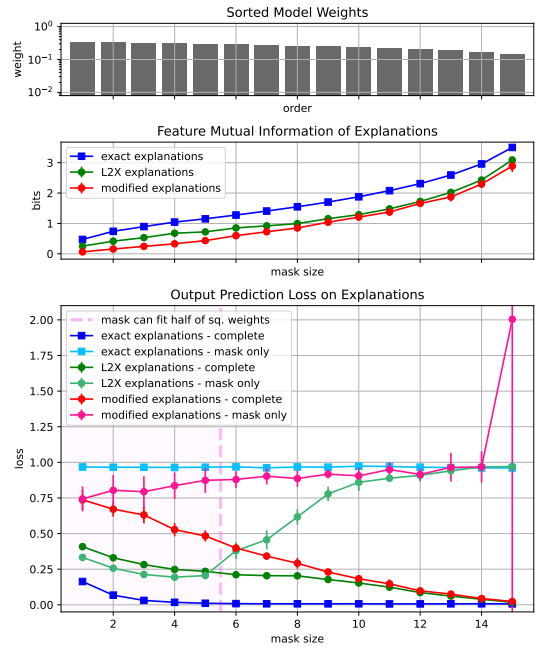
(a) Model 1



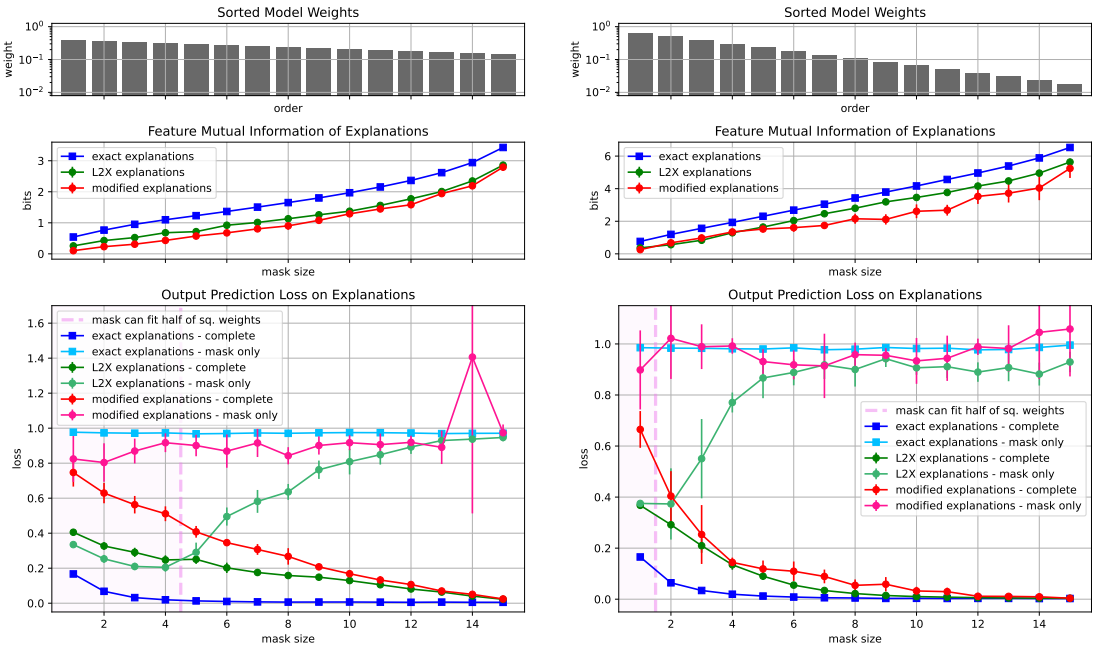
(b) Model 2



(c) Model 3



(d) Model 4



(e) Model 5 (f) Model 6

Figure C.4.: Plots showing the results of our experiments on active mitigation of output encoding. The used models and the displayed results for exact and learned explanations are the same as in Figure 5.2. Additionally to this, in this figure also the results obtained for L2X explanations with active mitigation of output encoding are shown labelled as *modified explanations*. In the graphs belonging to learned and modified explanations, the vertical error bars show the standard deviation of the results observed in 10 randomly initiated and trained explainer models.

C.3. Overview of Models and User Profiles

Tables C.1 and C.2 show the linear regression models and user profiles examined in the experiments of this thesis respectively.

chapter 5	chapter 6	squared weights
Model 1	Model A	6 weights $\frac{1-2^{-7}+2^{-17}}{6}$, and 2^{-i-1} for $i=7,\dots,16$
Model 2		6 weights $\frac{9}{60}$, 5 weights $\frac{9}{500}$, 5 weights $\frac{1}{500}$
Model 3		16 weights $\frac{1}{16}$
Model 4		$\frac{15+7i}{1080}$ for $i=0,\dots,15$
Model 5	Model B	$(2^{\beta-i/10})_{i=0}^{15}$ with $\beta\approx-1.391665$
Model 6		$(2^{\beta-11i/30})_{i=0}^{15}$ with $\beta\approx-0.663485$

Table C.1.: The linear regression models used for the experiments conducted in this thesis.

user profile	feature mask	locality	quality[4]
first	$\{1,\dots,l\}$	global	best global user profile
last	$\{d-l+1,\dots,d\}$	global	worst global user profile
random	$\sim \text{Uniform}\binom{[d]}{l}$	local	baseline local user profile
absolute	$\arg\max_{s\in\binom{[d]}{l}} \sum_{i\in s} \lvert x_i\rvert$	local	better than 'random', worse than 'greedy'
greedy	$\arg\max_{s\in\binom{[d]}{l}} \sum_{i\in s} \lvert x_i\alpha_i\rvert$	local	better than 'absolute', worse than 'exact'
exact	$\arg\max_{s\in\binom{[d]}{l}} p(y\lvert x_s)$	local	overall best user profile

Table C.2.: The user profiles used for the experiments conducted in this thesis. Note: Here we use the abbreviation $[d]=\{1,\dots,d\}$.

[4]This holds in relation to linear regression models with weights sorted by absolute value in descending order, such as the ones used in our experiments and listed in Table C.1.