# Advances in Explainable AI for Deep Learning: Algorithms and Applications

By

GE SHI

DISSERTATION

Submitted in partial satisfaction of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

Computer Science

in the

OFFICE OF GRADUATE STUDIES

of the

UNIVERSITY OF CALIFORNIA

DAVIS

Approved:

Ian Davidson, Chair

Xin Liu

Hamed Pirsiavash

Committee in Charge

2024

# Abstract

## Advances in Explainable AI for Deep Learning: Algorithms and Applications

In the last decade, thanks to the notable progress of Deep Learning (DL), Artificial Intelligence (AI) has achieved tremendous advancement. However, the lack of explainability has limited the wide use of deep models in real-world applications due to its black-box nature. This inherent defect of black-box AI systems confronts people with but is not limited to issues such as accessibility, improveability, and accountability. The paradigm underlying these problems falls into the so-called explainable AI (XAI) field, which is widely acknowledged as a crucial feature for the practical deployment of AI models [6]. This dissertation presents a comprehensive study of the advanced progress of XAI in multiple aspects from parameter space to input feature space and provides practical algorithms. The research combines concept taxonomy, algorithm designs, and empirical studies in various deep-learning systems.

The dissertation is structured as follows. Chapter 1 highlights the need for an XAI module in deep learning systems and categorizes general XAI approaches into two types: input space and parameter space explanations. It reviews key progress in the field and introduces approaches closely related to the works presented in later chapters. Chapter 2 focuses on loss landscapes, a popular parameter space explanation, and introduces new visualization tools for model diagnosis. Chapter 3 presents two post-hoc explanation methods: Shapley Value Integrated Gradients (SIG), an extension of Integrated Gradients, and a benchmark of XAI methods in low signal-to-noise ratio environments. Chapter 4 applies post-hoc explanations to machine learning, using feature ablation for clinical prognosis with task-fMRI data and combining large language models with an exemplar selection algorithm to explain unsupervised clustering in topic modeling.

This dissertation provides an overview of the research completed at the moment of writing, which captures the essence of the research and its significance.

# Acknowledgments

I would like to express my deepest gratitude to my primary instructor, Professor Ian Davidson, for his invaluable guidance and support throughout my academic journey. I am also sincerely grateful to Professor Hamed Pirsiavash and Professor Xin Liu for serving as members of my dissertation committee.

I extend my thanks to Professor Jiawei Zhang and Professor Jason Smucny for being part of my qualification exam committee. I am also fortunate to have had the support of my lab mates, Zilong Bai, Hongjing Zhang, Michael Livano, Kaitian Xie, and Jiaqing Chen, whose collaboration and friendship enriched my research experience.

Finally, I am deeply indebted to Prof. Michael Mahoney, Prof. Ross Maciejewski, Prof. Yaoqing Yang, Dr. Gunther Weber, Dr. Talita Perciano, and Dr. Caleb Geniesse for their mentorship during my time at Lawrence Berkeley National Laboratory. Their insights and guidance have been instrumental in my development as a researcher.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

Explainable Artificial Intelligence (XAI) refers to methods and techniques in the field of artificial intelligence (AI) that make the decision-making processes of AI systems understandable to humans. Unlike traditional AI models, which often operate as "black boxes," XAI aims to provide insights into how these models derive their conclusions, allowing users to interpret and trust the results. The importance of XAI lies in its ability to enhance transparency, accountability, and fairness, especially in critical applications such as healthcare, finance, and law, where decisions significantly impact human lives. By making AI systems more interpretable, XAI helps build trust and ensures that AI technologies are used responsibly and ethically.

**Explainability**: Explainability is defined as the ability to explain or to provide meaning in understandable terms to a human. Although there is a considerably large group of literature ([40, 75, 6, 88, 30, 17, 89]) defining the scope of XAI, some include confidence, fairness, accessibility, etc., there's a lack of a clear consensus on the term Explainable Artificial Intelligence (XAI). It might be interesting to place the reference starting point at the definition of the term given by D. Gunning in [41].

> “XAI will create a suite of machine learning techniques that enables human users to understand, appropriately trust, and effectively manage the emerging generation of artificially intelligent partners”.

Despite the complications, the explicit definition of Explainable Artificial Intelligence

(XAI) refers to the ability of an AI system to provide meaningful explanations or justifications for its decisions or outputs in a manner that is understandable to humans. This involves presenting the inner workings of the AI model in a transparent and interpretable manner, allowing users to comprehend the reasoning behind the system's outputs.

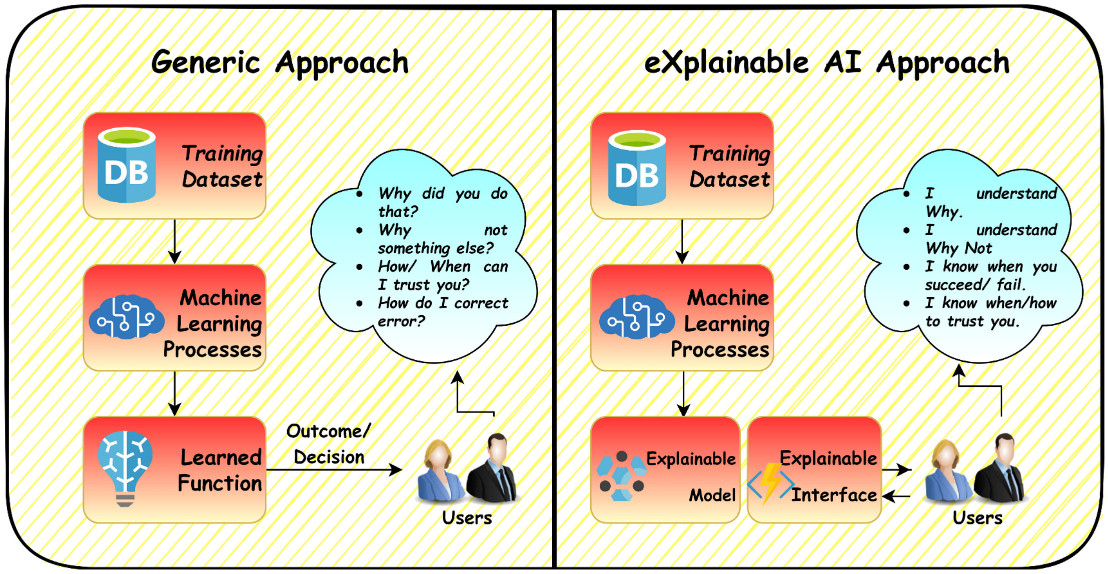


Figure 1.1: AI today and tomorrow with XAI. [17]

## 1.1 Why Do We Need Explainable Artificial Intelligence?

The concerns about the accountability of black-box machine learning systems push the governments to draft regulations. The General Data Protection Regulation (GDPR) ([84]) adopted by the European Parliament [85] includes clauses on automated decision-making and profiling. These clauses introduce, to some extent, a right of explanation for individuals to obtain "meaningful explanations of the logic involved" when automated decision-making takes place. There is still some debate among legal scholars regarding the scope of these clauses, but there is a consensus on the urgency of implementing the principle of explanation. Without technology capable of explaining the logic of black boxes, the right to an explanation would remain ineffective.

This regulation reflects the growing concern about using sophisticated machine-learning classification models trained on massive datasets without a clear understanding of their decision-making processes. It is recognized that this lack of understanding impacts ethics, accountability, safety, and industrial liability. Researchers put efforts into facilitating explainability for AI systems to deal with the regulations.

The need for XAI arises from increasingly using complex, "black box" machine learning models whose internal workings are either unknown or difficult to interpret. There are several reasons for the demand for XAI:

**Debugging and Troubleshooting**: XAI plays a crucial role in debugging machine learning systems by providing insights into model behavior, identifying sources of errors, and facilitating the resolution of issues. It's widely used to identify errors, detect data issues, evaluate performance, and improve model robustness.

**Trust and Awareness**: The availability of transparent machine-learning technologies would lead to a gain of trust and awareness. Users need to understand the reasons behind a decision or an event, especially when the consequences could be severe, such as in the case of the self-driving Uber car incident that resulted in the death of a pedestrian.

**Responsibility and Accountability**: In industries such as self-driving cars, medicine, biology, and socio-economic sciences, the use of machine-learning models requires an explanation not only for trust and acceptance of results but also for the sake of the openness of scientific discovery and the progress of research. Additionally, in cases where decisions have significant real-world impacts, it is important to be able to manage responsibilities and be accountable for the decisions made by AI systems.

**Scientific Research**: In fields like medicine, biology, and socio-economic sciences, the use of machine-learning models requires explanations not only for trust and acceptance of results but also for the sake of the openness of scientific discovery (1.2) and the progress of research.

**Legal and Ethical Considerations**: Regulatory bodies and legal frameworks are increasingly requiring explanations for decisions made by AI systems to ensure fairness, non-discrimination, and compliance with regulations.

**Decision-making**: In scenarios where crucial decisions are made based on predictions or

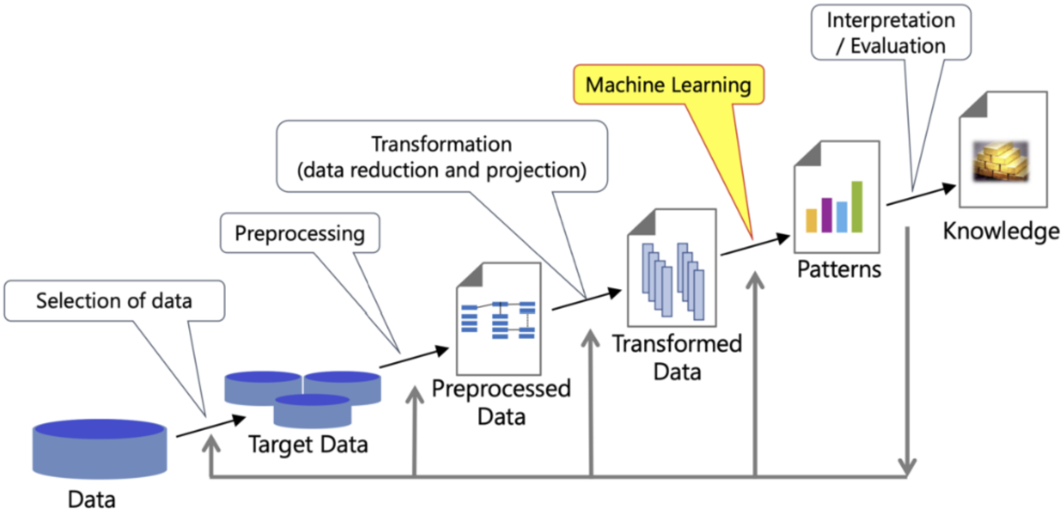


Figure 1.2: Knowledge Discovery Process. [88]

classifications provided by AI models, the decision-maker needs to understand the rationale behind the model's output. For example, in medical applications, a surgeon might need to understand the reasoning behind a model's recommendation for a particular course of action.

**Comprehensibility**: The need for an interpretable model can vary based on the consequences of the decision. If there are no crucial decisions to be made based on the model's output, or there are no consequences for unacceptable results, then an interpretable model may not be necessary.

**Facilitating Collaboration and Communication**: Explanations provide a common language for stakeholders to discuss and understand model behavior. By visualizing model decisions and predictions in an interpretable manner, XAI fosters collaboration between data scientists, domain experts, and end-users, facilitating the debugging process.

With the efforts of the entire community, the AI system tomorrow (Figure 1.1) will be much more transparent and interactive thanks to the development of XAI.

## XAI Categorization

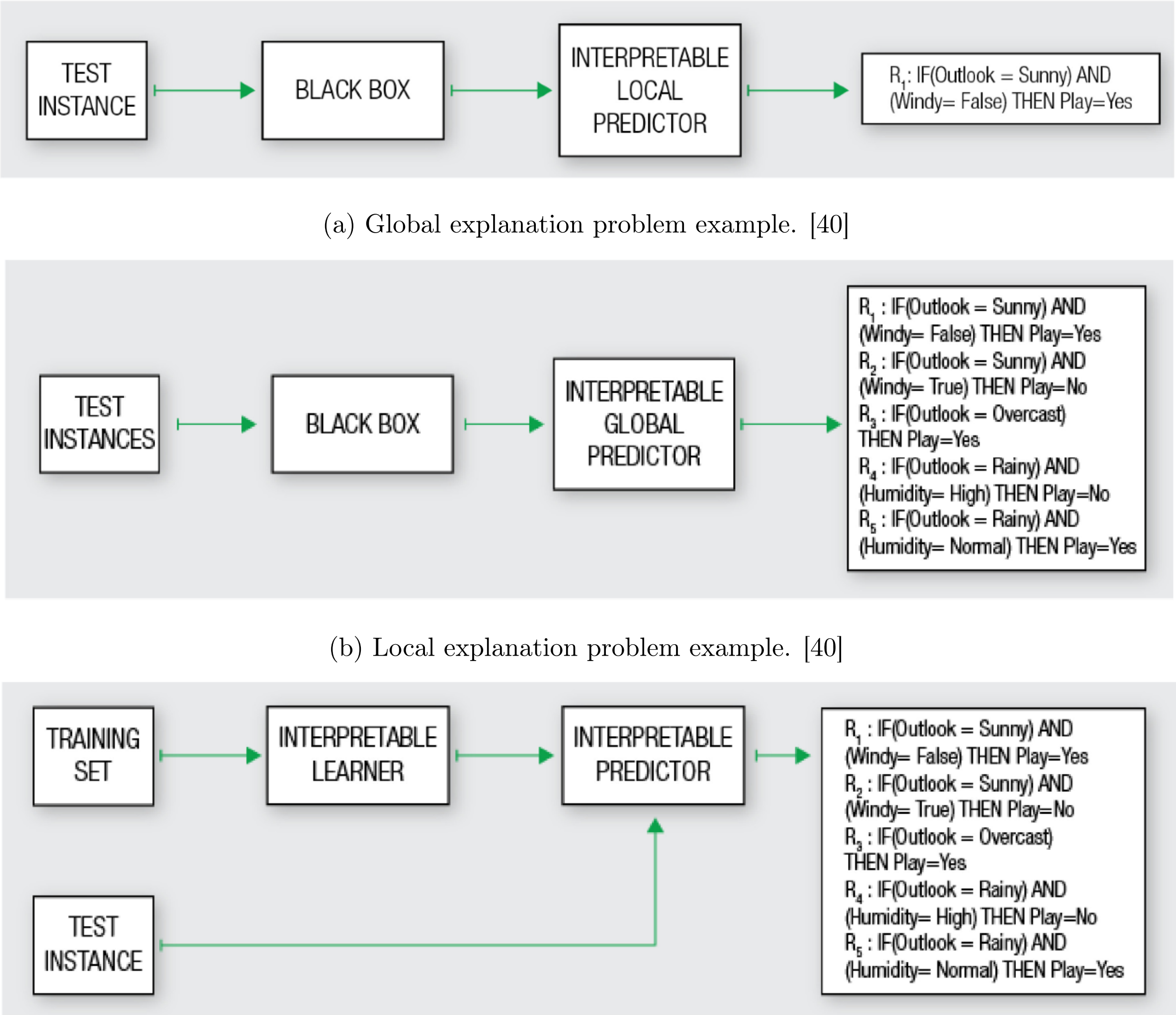
There are many ways to categorize XAI methods by different criteria.

- Based on the modality of choice, the XAI methods can be categorized into explanations of the parameter space and explanations of the feature space. The explanation of parameter space puts attention on the effects of modules and optimizations.
- Based on the domain of the data, the XAI methods can be categorized into explanations on tabular data, image data, text data, and time series data.
- Based on when the explainability takes place, the XAI methods can be categorized into intrinsic transparent models and post-hoc explanations. The transparent models achieve interpretability to some extent at the stage of design and model build. The post-hoc method explains the behaviors of the pre-trained black-box model.
- Based on the scope of explanations on data, the XAI methods can be categorized into local explanations, semi-global explanations, and global explanations. The scope of them ranges from decisions of single instances, subsets, and entire datasets.
- Based on what models the XAI method supports, the XAI methods can be categorized into model-specific methods and model-agnostic methods. Model-specific methods are only technically effective for models with certain architectures while model-agnostic method is universally capable.

These criteria are used for high-level conceptual categorizations. The combinations of them signify each specific XAI method. As for the research presented in this thesis, in Chapter 2 Shapley Value Integrated Gradients (SIG) is a local level post-hoc explanation method; Hierarchical Aggregation of Local explanation (HALE) is generated semi-global explanations in a post-hoc manner; Chapter 3 presents two applications of post-hoc explanation techniques applied to scientific data, aimed at enhancing both trust and performance.

## Challenges and Opportunities

Explainable Artificial Intelligence (XAI) presents unique challenges and opportunities across different audiences, including research scientists, machine learning engineers, and domain experts. For research scientists, the main challenge is developing XAI methods that strike a balance between model interpretability and accuracy, ensuring that explanations are both


TEST INSTANCE
BLACK BOX
INTERPRETABLE LOCAL PREDICTOR
$R_1$: IF(Outlook = Sunny) AND (Windy= False) THEN Play=Yes
TEST INSTANCES
BLACK BOX
INTERPRETABLE GLOBAL PREDICTOR
$R_1$ : IF(Outlook = Sunny) AND (Windy= False) THEN Play=Yes
$R_2$ : IF(Outlook = Sunny) AND (Windy= True) THEN Play=No
$R_3$ : IF(Outlook = Overcast) THEN Play=Yes
$R_4$ : IF(Outlook = Rainy) AND (Humidity= High) THEN Play=No
$R_5$ : IF(Outlook = Rainy) AND (Humidity= Normal) THEN Play=Yes
TRAINING SET
INTERPRETABLE LEARNER
INTERPRETABLE PREDICTOR
TEST INSTANCE
$R_1$ : IF(Outlook = Sunny) AND (Windy= False) THEN Play=Yes
$R_2$ : IF(Outlook = Sunny) AND (Windy= True) THEN Play=No
$R_3$ : IF(Outlook = Overcast) THEN Play=Yes
$R_4$ : IF(Outlook = Rainy) AND (Humidity= High) THEN Play=No
$R_5$ : IF(Outlook = Rainy) AND (Humidity= Normal) THEN Play=Yes

(a) Global explanation problem example. [40]

(b) Local explanation problem example. [40]

(c) Transparent model example. [40]

meaningful and scientifically sound. Machine learning engineers face the practical challenge of implementing these methods in real-world systems, ensuring that they are computationally efficient and seamlessly integrated into existing workflows. Domain experts, who often rely on AI systems for decision-making, must grapple with the complexity of understanding AI-generated explanations within the context of their specific fields. Despite these challenges, XAI offers significant opportunities, such as enhancing interdisciplinary collaboration, increasing transparency, and fostering trust in AI technologies. By making AI systems more interpretable, XAI enables all stakeholders to understand better and validate AI-driven decisions, ultimately leading to more responsible and informed use of AI across various domains.

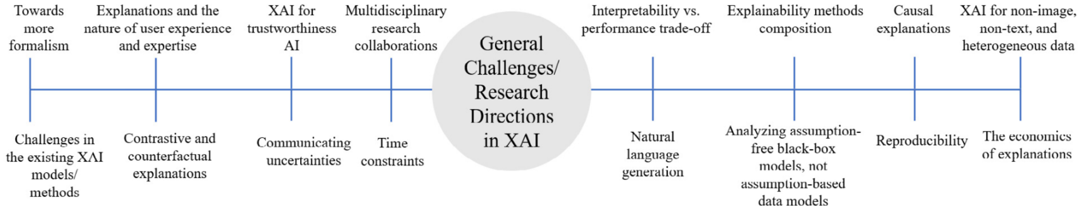

Towards more formalism
Explanations and the nature of user experience and expertise
XAI for trustworthiness AI
Multidisciplinary research collaborations
General Challenges/ Research Directions in XAI
Interpretability vs. performance trade-off
Explainability methods composition
Causal explanations
XAI for non-image, non-text, and heterogeneous data
Challenges in the existing XAI models/ methods
Contrastive and counterfactual explanations
Communicating uncertainties
Time constraints
Natural language generation
Analyzing assumption-free black-box models, not assumption-based data models
Reproducibility
The economics of explanations

Figure 1.4: General Challenges and Research Directions in XAI. [89]

XAI presents both challenges and opportunities in various domains [89] as shown in Figure 1.4. Here are some of them that are related to our work:

**There's a lack of understanding, diagnosing, and improving the training process of models.** This kind of model-level or algorithm-level explanation is essential. To overcome this challenge, loss landscapes, or visualizations of how a loss function behaves with respect to the parameters of a model have been proposed [61]. It is useful in understanding the convergence behavior, identifying local minima and saddle points, improving and choosing optimization algorithms, and diagnosing training issues. By examining the landscape, researchers and developers can see whether the training process is likely to converge towards a minimum. A smooth landscape with a clear global minimum is ideal because it suggests that optimization algorithms like gradient descent are more likely to succeed in finding the lowest possible loss, thus optimizing the model effectively. A very steep landscape might indicate a high sensitivity to changes in model parameters, which can lead to issues like exploding gradients. Conversely, a very flat landscape might be prone to vanishing gradients, where changes in loss are so minute that the model stops learning effectively.

**There's a lack of study on how XAI can guide feature selection.** There is a notable gap in research exploring how Explainable Artificial Intelligence (XAI) can be leveraged for feature selection, a critical process in building effective machine learning models. Feature selection involves identifying the most relevant inputs that contribute significantly to the predictive power of a model, thus improving performance and interpretability. While XAI techniques are typically employed to elucidate the decision-making processes of complex models, their potential to systematically identify and prioritize features remains underexplored. This underutilization of XAI in feature selection represents a missed opportunity, as these methods can provide insights into feature importance and interactions that traditional selection techniques might overlook. Addressing this gap could lead to more efficient and transparent models, as well as a deeper understanding of the underlying data patterns, thereby advancing both the fields of machine learning and explainability.

**There's a great need for XAI in the scientific machine learning field.** XAI plays a crucial role in scientific machine learning, a field that intersects artificial intelligence with scientific research across disciplines like physics, chemistry, biology, and neuroscience sciences. Scientific machine learning often aims not just to predict or classify but to uncover underlying scientific principles and mechanisms. XAI can help reveal how certain inputs (e.g., molecular structures, environmental variables) are related to outputs (e.g., chemical properties, climate forecasts), potentially leading to new scientific insights and hypotheses. By understanding the model's decision-making process, scientists can identify new patterns and relationships that might not be apparent through traditional methods. There's an opportunity to use the XAI method to improve the scientific machine-learning pipeline and justify the learned model.

Inspired by these challenges, we worked to directly solve them which are demonstrated in the next few chapters. Before diving deep into the details of each chapter, we introduce the background knowledge of them in Section 1.2 and Section 1.3.

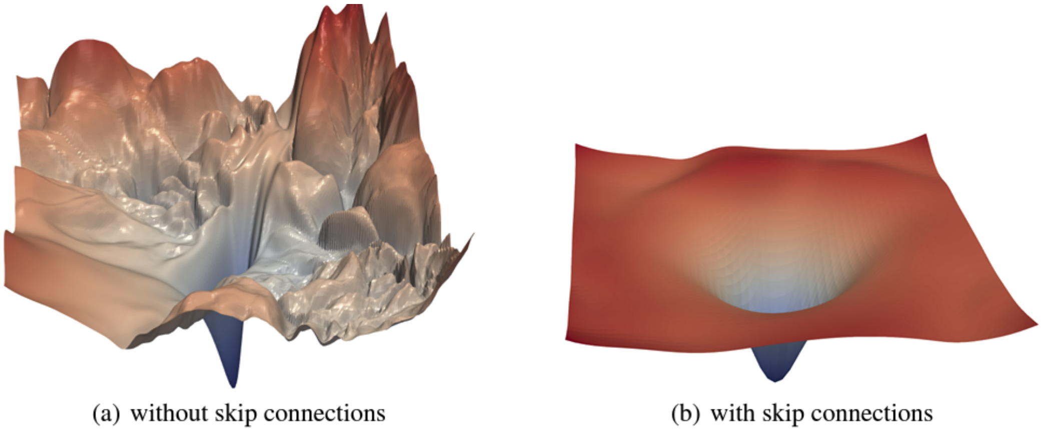
## 1.2 Taxonomizing the Loss Landscapes

Among the various approaches to understanding the behavior of neural network (NN) models, the exploration of their loss landscapes [61] has emerged as particularly insightful. Indeed, analyzing loss landscapes has provided valuable insights into the functioning of many popular techniques, including large-batch training [53, 116], adversarial training [115], residual connections [64], and BatchNorm [91]. The conceptualization of neural network models in terms of their loss landscapes has roots in the statistical mechanics approach to learning, with recent years witnessing increased attention [114] within machine learning circles.

Notably, local metrics, such as the smoothness of the loss landscape, have been demonstrated to correlate with global properties of the model, such as good generalization performance. A recent concept of interest is the notion of *sharpness* of local minima. While sharpness can be assessed using first-order sensitivity measures like the Jacobian or Lipschitz constant, it is more suitably gauged through second-order sensitivity measures, often via the Hessian spectrum [117]. It has been noted that in certain cases, neural networks generalize effectively when they converge to a relatively flat, i.e., non-sharp, local minimum [53].

Training neural networks hinges on our capacity to discover "good" local minima of highly non-convex loss functions. Certain network architecture designs, such as skip connections, yield loss functions that are easier to train, while well-chosen training parameters (batch size, learning rate, optimizer) produce local minima that generalize better. However, the reasons for these design choices and their impact on the underlying loss landscape remain poorly understood. [61] propose a perturbation approach of 2D visualization of local loss landscapes to answer crucial questions regarding the functioning of neural networks, including why we can minimize highly non-convex neural loss functions and why the resulting minima generalize. Figure 1.5 is a visualization of ResNet with/without skip connections. It is observed that skip connections promote flat local minima and prevent the transition to chaotic behavior, which helps explain why skip connections are necessary for training extremely deep networks. From the visualization we see that the sharpness of local minima correlates effectively with generalization error when employing this normalization method, facilitating direct comparisons across diverse network architectures and training methodologies. This allows for straightforward side-by-side evaluations of different local minima.

While local sharpness measures offer insight, their focus on the local geometry of the loss landscape overlooks the global structure, which statistical mechanics approaches to learning aim to quantify. However, many observations [38, 105] on the empirical correlation between local metrics like sharpness and more global properties like generalization performance suggest it may be correlative rather than causative. Motivated by these considerations, [114] propose a 4-phase categorization which is useful for identifying global structure versus local structure in loss landscapes. Here the phases mean different types of local minima. When training from random parameter initialization, taking different hyperparameters, a model falls into different phases after convergence. Note that the phases do not mean a model will go over all 4 of them during one training episode.


(a) without skip connections

(b) with skip connections

Figure 1.5: The loss surfaces of ResNet-56 with/without skip connections. The proposed filter normalization scheme is used to enable comparisons of sharpness/flatness between the two figures. [61]

- **Phase I**: Globally poorly connected and locally sharp: Training loss is high; Hessian eigenvalue and trace (idwhat are the Hessian eigenvalues etc calculated on? you never say in detail) are large, and mode connectivity is poor.

- **Phase II**: Globally well-connected and locally sharp: Training loss is high; Hessian eigenvalue and trace are large; and mode connectivity is poor because the trained weights fail to locate a reasonable minimum.

- **Phase III**: Globally poorly connected and locally flat: Training loss is small; Hessian eigenvalue and trace are small; yet mode connectivity remains poor.

- **Phase IV**: Globally well-connected and locally flat: Training loss is small; Hessian eigenvalue and trace are small; and mode connectivity is good (near-zero).

The general *loss function* of a neural network is defined as:

$$\mathcal{L}(\theta) = L(D, \theta) = \frac{1}{n} \sum_{i=1}^{n} l(x_i, y_i; \theta), \tag{1.1}$$

Given a loss function and parameter, the phases are distinguished by local and global statistics such as the Hessian matrix and Mode connectivity.

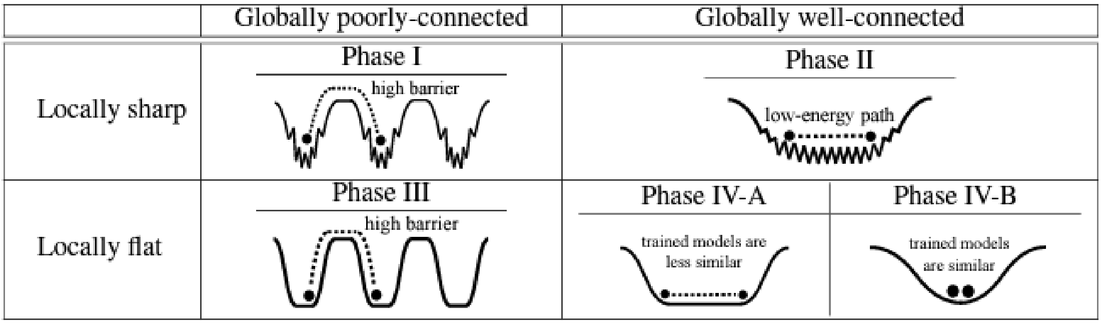


Figure 1.6: Caricature of different types of loss landscapes. Globally well-connected versus globally poorly-connected loss landscapes; and locally sharp versus locally flat loss landscapes. Globally well-connected loss landscapes can be interpreted in terms of a global "rugged convexity"; and globally well-connected and locally flat loss landscapes can be further divided into two sub-cases, based on the similarity of trained models. [114]

**Hessian** information is provided by the second-order derivative of the loss function with respect to weights $\theta$. The Hessian captures the local curvature of the loss landscape [114], and it can be used to differentiate the locally sharp versus locally flat loss landscapes. The Hessian matrix is represented as $\nabla^2_\theta \mathcal{L}(\theta)$ for weights $\theta$. In this work, we calculate the top-10 largest eigenvalues, computed as $\{\lambda_i(\nabla^2_\theta \mathcal{L}(\theta)), i \in [1, 10]\}$, using the `PyHessian` software [117].

$$H = \frac{\partial^2 \mathcal{L}}{\partial \theta^2} = \frac{\partial g_\theta}{\partial \theta} \in \mathbb{R}^{m \times m}, \tag{1.2}$$

where $m$ is the number of parameters in the neural network, and $g_\theta$ is the gradient of the loss, defined as

$$g_\theta = \frac{\partial \mathcal{L}}{\partial \theta} \in \mathbb{R}^m, \tag{1.3}$$

where $\mathcal{L}$ and $\theta$ are defined in Eq. 1.1. The idea is that by using the eigenvectors associated with the top $k$ largest eigenvalues, one can visualize the most significant local loss fluctuations for a given model. This approach is more reproducible (not relying on randomly sampled directions) and can be used to visualize the local loss landscape of an individual model more effectively and for example, to relate the local loss landscape to metrics like generalization.

**Mode Connectivity** [34] refers to the existence of curves or pathways that connect the optima (modes) of loss landscapes in a neural network. These curves can be represented by simple geometric shapes, such as polygonal chains with only one bend. Mode Connectivity provides a new understanding of the geometric properties of neural network loss surfaces, and it has significant implications for training efficiency, ensembling techniques, and Bayesian deep learning [34]. Computing the mode connectivity between two sets of weights $\theta$ and $\theta'$ involves finding a low-loss curve $\gamma(t), t \in [0,1]$, where $\gamma(0) = \theta$ and $\gamma(1) = \theta'$, such that $\int \mathcal{L}(\gamma(t))dt$ is minimized. Given the curve represented as $\gamma_\phi(t)$, we define the mode connectivity between $\theta$ and $\theta'$ to be

$$mc(\theta, \theta') = \frac{1}{2}(\mathcal{L}(\theta) + \mathcal{L}(\theta')) - \mathcal{L}(\gamma_\phi(t^*)), \tag{1.4}$$

where $t^*$ maximizes the deviation $t \mapsto |\frac{1}{2}(\mathcal{L}(\theta) + \mathcal{L}(\theta')) - \mathcal{L}(\gamma_\phi(t))|$. In general, $mc(\theta, \theta') < 0$ means the existence of a high loss "barrier" between $\theta$ and $\theta'$, indicating the mode connectivity is *poor* and the loss landscape is considered to be poorly-connected. $mc(\theta, \theta') > 0$ indicates the existence of a (relatively) lower loss path between $\theta$ and $\theta'$, which implies that the initial training failed to find reasonable optima. A *good* mode connectivity requires $mc(\theta, \theta') \approx 0$. In this case, we say the loss landscape is ideally well-connected. Globally poorly connected loss landscapes have high barriers between different local minima, which can be measured using mode connectivity.

**CKA Similarity** [56] is a metric for evaluating the similarity between two sets of features learned by two neural networks. CKA is widely used because the similarity is invariant to orthogonal transforms and isotropic scaling, which is a desired property to deal with rotations and scalings of feature representations. It is more robust to variations in network architectures and hyperparameters, compared to other similarity metrics such as cosine similarity or correlation coefficient [56]. Formally, for a neural network $f_\theta$ with weights $\theta$, let $F_\theta = \begin{bmatrix} f_\theta(\boldsymbol{x}_1) & \cdots & f_\theta(\boldsymbol{x}_m) \end{bmatrix}^\top \in \mathbb{R}^{m \times d}$ denote the concatenation of the (vectorized) feature maps of length-$d$ of the network over a set of $m$ randomly sampled data points. Then the (linear) CKA similarity between two sets of feature maps is given by

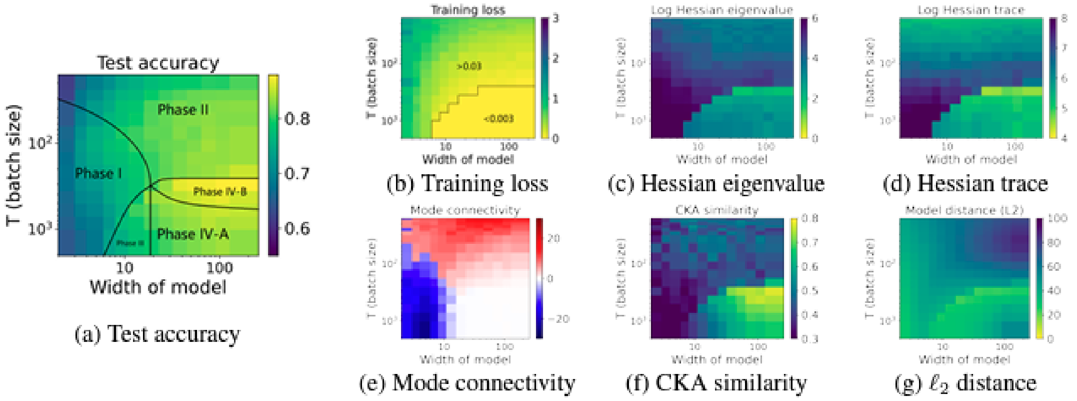
$$s(F_\theta, F_{\theta'}) = \frac{\mathrm{Cov}(F_\theta, F_{\theta'})}{\sqrt{\mathrm{Cov}(F_\theta, F_\theta)\mathrm{Cov}(F_{\theta'}, F_{\theta'})}}, \tag{1.5}$$

where we define $\mathrm{Cov}(X, Y) = (m-1)^{-2}\mathrm{tr}(XX^\top H_m YY^\top H_m)$, and $H_m = I_m - m^{-1}\mathbf{1}\mathbf{1}^\top$ is the centering matrix, for $X, Y \in \mathbb{R}^{m\times d}$. We note that the feature maps $F_\theta$ and $F_{\theta'}$ can be taken from any layer of two neural networks. They can even be taken from two layers of the same neural network. When comparing two neural networks $\theta$ and $\theta'$, we often represent the CKA similarity using a layer-wise similarity matrix, where the $(i{,}j)$-th coordinate represents the similarity between the feature maps taken from the $i$-th layer of $\theta$ and the $j$-th layer of $\theta'$. CKA similarity further divides the globally well-connected and locally flat loss landscapes into two subtypes.

*The Hessian matrix helps distinguish between locally sharp and locally flat loss landscapes.* The transition between the first and second phases is depicted in Figure 1.7c and d, marking the separation of Phase I/II from Phase III/IV. A larger Hessian eigenvalue or Hessian trace (shown in darker color) indicates a sharper local loss landscape. In Figure 1.7b, we observe that this transition coincides with a significant decrease in the training loss. Indeed, there is a reduction of more than tenfold in the training loss when transitioning from the upper side to the lower side on the right of the figure. However, comparing Figure 1.7a and c-d, categorizing loss landscapes based solely on the Hessian (or other local flatness metrics from other results) is insufficient to predict test accuracy. For instance, the test accuracy in Phase III is lower than in Phase IV-A, but the Hessian eigenvalues are almost the same.

*Mode connectivity distinguishes globally well-connected versus globally poorly-connected loss landscapes.* The second phase transition is illustrated in Figure 1.7e. The white region represents near-zero mode connectivity, indicating a flat curve in the loss landscape between freshly-trained weights. The blue region represents negative mode connectivity, implying a high barrier between weights, while the red region represents positive mode connectivity, indicating a low-loss curve between weights, although not trained to a reasonable optimum. In contrast to training loss, test accuracy only appears to show significant improvements after this transition. Particularly for well-connected loss landscapes, test accuracy can be improved with a suitable choice of temperature. This phase transition forms a curve separating Phase I from II and separates Phase III from IV.

Promising findings arise based on these definitions: optimal test accuracy is achieved when the loss landscape is globally favorable and the trained model converges to a locally



(a) Test accuracy

(b) Training loss

(c) Hessian eigenvalue

(d) Hessian trace

(e) Mode connectivity

(f) CKA similarity

(g) $\ell_2$ distance

Figure 1.7: The 2D diagram, resembling a load-temperature plot, is partitioned into different phases of learning by using batch size as the temperature and varying model width to adjust the load. Models are trained with ResNet18 on CIFAR-10, and all plots share the same set of axes. It's important to note that batch size is inversely related to temperature, with smaller values at the top of the y-axis and larger values at the bottom. [114]

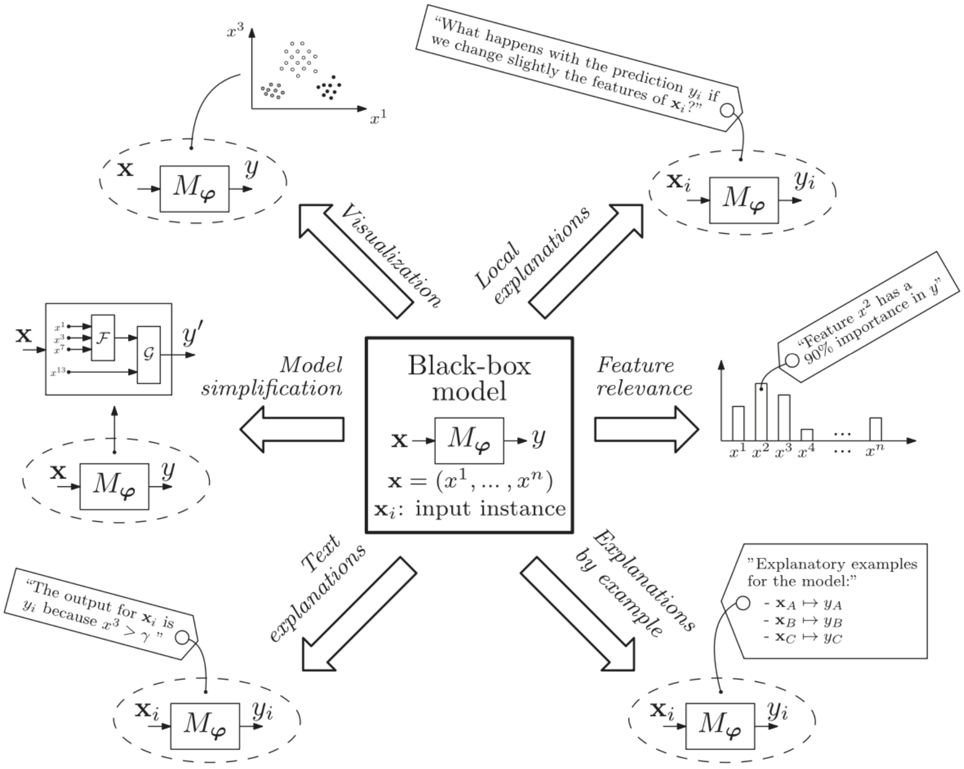
flat region. These different phases in the load-like–temperature-like phase diagram can be diagnosed using Hessian, mode connectivity, and CKA metrics. Importantly, both similarity and connectivity metrics are necessary for a globally favorable loss landscape. Phase IV-B precisely represents the region with globally favorable landscapes, exhibiting the highest test accuracies. This claim serves as the foundation of our exploration of loss landscapes.

## 1.3 Taxonomizing the post-hoc Explanations

Post-hoc explainability methods in AI are essential for the interpretability of black-box models without compromising performance. In order to enhance audience comprehension, we utilize the concepts introduced in [6] to categorize post-hoc XAI methods as outlined below.

**Text Explanations**: These methods provide explanations in textual form to make the model's decision-making process understandable. They aim to convey the rationale behind the model's predictions using natural language.

**Visualizations**: Visual explanation techniques use various visualization methods to help in the explanation of a black-box ML model. They present visual representations of the model's behavior and decision-making process, making it easier for users to comprehend



$x^3$
$x^1$
“What happens with the prediction $y_i$ if we change slightly the features of $\mathbf{x}_i$?”
$\mathbf{x}$ $M_{\boldsymbol{\varphi}}$ $y$
$\mathbf{x}_i$ $M_{\boldsymbol{\varphi}}$ $y_i$
Visualization
Local explanations
$\mathbf{x}$ $x^1$ $x^3$ $x^7$ $x^{13}$ $\mathcal{F}$ $\mathcal{G}$ $y'$
“Feature $x^2$ has a 90% importance in $y$”
Model simplification
Black-box model
$\mathbf{x}$ $M_{\boldsymbol{\varphi}}$ $y$
$\mathbf{x} = (x^1, \ldots, x^n)$
$\mathbf{x}_i$: input instance
Feature relevance
$\mathbf{x}$ $M_{\boldsymbol{\varphi}}$ $y$
$x^1$ $x^2$ $x^3$ $x^4$ $\cdots$ $x^n$
Text explanations
Explanations by example
“The output for $\mathbf{x}_i$ is $y_i$ because $x^3 > \gamma$”
”Explanatory examples for the model:”
- $\mathbf{x}_A \mapsto y_A$
- $\mathbf{x}_B \mapsto y_B$
- $\mathbf{x}_C \mapsto y_C$
$\mathbf{x}_i$ $M_{\boldsymbol{\varphi}}$ $y_i$
$\mathbf{x}_i$ $M_{\boldsymbol{\varphi}}$ $y_i$

Figure 1.8: Conceptual diagram showing the different post-hoc explainability approaches available for an ML model. [6]

the model's inner workings. For example, a Partial Dependence Plot (PDP) offers a way to explore how the model's predictions change as the inputs vary while holding other inputs constant.

**Explanations by Example**: This method involves providing explanations through examples, illustrating how the model's predictions are influenced by specific input features or patterns. It uses concrete examples to elucidate the model's decision-making process.

**Explanations by Simplification**: These methods aim to simplify complex models into more interpretable forms while maintaining a similar performance score. By simplifying the model, these methods make it easier to understand and implement.

**Feature Relevance**: Feature relevance explanation methods compute relevance scores for the model's input variables, indicating the impact of each feature on the model's output. These scores reveal the importance of different variables in the model's decision-making process.

**Activation Maximization (AM)**: AM involves observing which neurons are activated with respect to particular input records in neural networks. It helps in understanding the fundamental neurons activated and identifying input patterns that maximize the activation.

Among these categories, the most popular post-hoc local XAI method in practice is the **Saliency Mask (SM)**. This method visually highlights the determining aspects of the analyzed record, offering an efficient way to point out what causes a certain outcome, especially in images or texts. It is used to explain deep neural networks and can be viewed as a visual representation of feature importance. Figure 1.9 shows saliency maps of a neural network for semantic segmentation task.

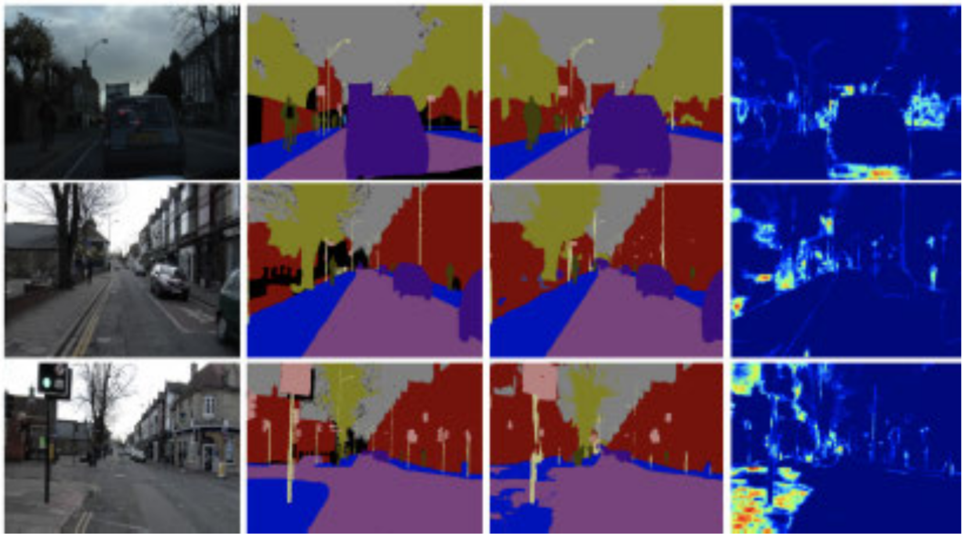


Figure 1.9: Uncertainty in computer vision for autonomous driving. [17]

**IG (Integrated Gradients)**: IG is an attribution method used in the context of deep learning. It computes the average gradient of the output with respect to each input feature while the input varies along a linear path from a chosen baseline to the input. The baseline is typically chosen to be zero. Integrated Gradients satisfy a notable property: the attributions sum up to the target output minus the target output evaluated at the baseline.

**LRP (Layer-wise Relevance Propagation)**: LRP is used to attribute the model's prediction to input features. It aims to decompose the model's output to understand the relevance of input features.

**DeepLIFT (Deep Learning Important FeaTures)**: DeepLIFT is a method for feature attribution in deep neural networks. It helps in understanding the contribution of each input feature to the model's output.

**Perturbation-based methods**: These methods involve altering input features and observing the effect on the model's output. They help in understanding the sensitivity of the model's predictions to changes in input features.

These local XAI methods focus on explaining individual predictions or instances, providing insights into why a particular prediction was made. They are particularly useful for understanding the model's behavior at a granular level.

Given all these techniques in XAI, we also dive into their application of real-world deep learning systems. In Chapter 3 we present the applications on neural imaging data and large language model-based topic modeling. These works shed light on the broad use cases of XAI to increase interpretability to both domain experts and general audiences.

## 1.4 Summary of Contributions

Motivated by the opportunities and techniques to explore the explainability of black-box machine learning models, we make the following contributions which serve as the main contents of our dissertation. The approaches presented in this dissertation are unified in exploring novel methods and new use cases for explaining deep learning models. This ranges from (Chapter 2) Feature level explanation on input space to mine important features from local with comprehensive experiments, to (Chapter 3) novelly using post-hoc explanations to justify machine learning system design and provide high-level explanations to topic modeling. Table 1.1 presents a high-level summary of the main contents of the following Chapters and my contribution to each specific project.

Table 1.1: The summary of my Contribution.

| **Chapters** | **Novelty** | **My Contribution** |
|---|---|---|
| Chapter 2<br>Feature-Level XAI on the basis of Chapter 1.3 | Chapter 2.1: Create a local attribution method that improves the Integrated Gradient method by approximating Shapley Values | • Prove that IG with multi-baselines is equivalence to SV<br>• Salient feature dropping experiment |
| | Chapter 2.2: Provide a dataset and benchmark in low signal-to-noise rate environment | • Benchmark *Model* × *Attribution* × *Noise Conditions*<br>• Leverage feature-attribution methods for feature selection |
| Chapter 3<br>XAI Applications | Chapter 3.1: Use feature ablation method to validate deep CNN on task-fMRI prognosis problem. | • Propose the multi-view multi-instance learning setting<br>• Feature ablation on frames of the scan to justify design |
| | Chapter 3.2: Combine exemplar-based XAI with LLM to generate mind maps for diverse topic modeling. | • Apply XAI to topic models to extract diverse exemplars<br>• LLM-based hierarchical mind map generation for each topic |

# Chapter 2

# Explanations on the Feature Space

As artificial intelligence (AI) becomes more sophisticated and indispensable in our daily lives, the necessity of transparency and accountability in AI decision-making has grown [1, 42] significantly. Explainable Artificial Intelligence (XAI) was proposed and has been increasingly studied within the community that focuses on developing machine learning models and systems that can offer understandable explanations for their decisions and predictions [6, 25]. Research on XAI boomed exceptionally after the paramount success of deep neural networks (DNNs), as they are considered by many as complex yet intriguing "black-box" models [5, 82].

Among the numerous methods in Explainable AI (XAI), attributing a deep network's prediction to its input features has gained popularity for its intuitive explanations and broad applicability [98, 4, 46]. Attribution is an approach to explain a single prediction of a black-box model in a post-hoc manner. Attribution methods attribute a deep network's prediction to its input features. In other words, for a particular instance, they assign a scalar value to each feature to denote its influence on the prediction through a deep network. Attribution methods have been used to discover influential features and decipher what the neural networks have learned. The definition of this category is:

**Definition 1.** *Suppose we have a function $F : R_m \rightarrow [0, 1]$ that represents a deep network, and an input $x = (x_1, \ldots, x_m) \in R_m$. Attribution of the prediction at input $x$ relative to a baseline input $\bar{x}$ is a vector $A_F(x, \bar{x}) = (a_1, \ldots, a_m)$, where $a_i$ is the contribution of $x_i$ to the prediction $F(x)$.*

We provide a few popular attribution methods [4] as examples, such as Saliency [98], Integrated Gradient [103], DeepLift [97], and Feature Ablation [118]. Besides, there remains a wealth of other significant techniques, such as SHAP [67], CAM [121], LIME [86], MAPLE [80], and LRPs [10], which we will not go into their details.

**Saliency** (SA) is the pure gradient of the function $f$ with respect to the input features.

$$a_i = \frac{\partial F(x)}{\partial x_i} \tag{2.1}$$

**Integrated Gradient** (IG) computes the integral of gradients while the input varies along a linear path from a baseline $\bar{x}$ to $x$ (normally zeros). In the implementation, $\alpha$ is discretized into 10 steps.

$$a_i = (x_i - \bar{x}_i) \cdot \int_{\alpha=0}^{1} \frac{\partial F(\tilde{x})}{\partial \tilde{x}_i}\bigg|_{\tilde{x}=\bar{x}+\alpha(x-\bar{x})} d\alpha \tag{2.2}$$

**DeepLift** (DL) decomposes the output prediction of $f$ by backpropagating the contributions of all neurons in the network to every feature of the input with the rescale rule. In practice, it provides attribution quality comparable with IG but faster.

$$r_i^{(l)} = \sum_j \frac{z_{ji} - \bar{z}_{ji}}{\sum_{i'} z_{ji} - \sum_{i'} \bar{z}_{ji}} r_j^{l+1}, \; z_{ji} = w_{ji}^{(l+1,l)} \bar{x}_i^{(l)} \tag{2.3}$$

$$a_i = r_i^{(0)}, \; r^{(L)} = F(x) - F(\bar{x}) \tag{2.4}$$

**Feature Ablation** (FA) is a perturbation-based approach to computing attribution which computes the difference in $f(x)$ by replacing each feature $x_i$ with a baseline $\bar{x}_i$ (normally zero).

$$a_i = F(x) - F(x_{[x_i=\bar{x}_i]}) \tag{2.5}$$

In this chapter, we present two works that extended the landscapes of post-hoc attribution methods in XAI. The first one in Chapter 2.1 describes an algorithm to improve the well-known integrated gradient (IG) method. The second one in Chapter 2.2 provides a curated dataset to benchmark various attribution methods.

## 2.1 SIG: Rethinking Baseline of Integrated Gradients from the Perspective of Shapley Value

Numerous studies have focused on interpreting deep neural networks (DNNs) by linking DNN predictions to their input features. Integrated Gradients (IG) is a well-established method in this domain. A critical aspect of IG is the selection of baselines, which significantly affects the quality and bias of explanations in various scenarios. Common practice involves using a single baseline, which often falls short of providing comprehensive explanations, hence the need for multiple baselines. The link between IG and the Aumann-Shapley Value offers a fresh perspective on the baseline design. Under specific assumptions, we theoretically demonstrate that a set of baselines corresponds to coalitions in Shapley Value calculations. We introduce a new baseline construction method named Shapley Integrated Gradients (SIG) [65], which employs proportional sampling to mimic the computation process of Shapley Values. Our simulation results in GridWorld tasks indicate that SIG can effectively approximate the proportions of Shapley Value. Additionally, our empirical results on the image classification task of ImageNet demonstrate that SIG provides better explanations with negligible computational overhead.

### Introduction

Integrated Gradients (IG) [103] integrates the gradients of the model's output with respect to the input along a linear path from baseline to input. IG's mechanism and ease of implementation make it highly useful for discerning how input features affect the model's output. Since its inception, IG has seen continuous developments [36, 50, 113, 81], including applications in language model explanations [31] and electronic health records [29]. For IG, and path attribution methods in general, selecting a hyperparameter known as the baseline input is essential. This choice is vital in generating meaningful feature attributions and explanations [33].

Several baseline inputs are commonly adopted in XAI including random [50], zero [3], and mean [23], all of which are single baseline values. However, using a single baseline often falls short due to several reasons. Firstly, a baseline perceived as neutral and unrelated to specific tasks might not align with the recommended "near-zero" score for effective explanations [103]. Secondly, a single baseline meticulously crafted for a specific model is not generic to all instances [19]. Intuitively, the challenges associated with a single baseline motivate us to exploit a set of baselines for IG.

Intuitively, the challenges associated with a single baseline motivate us to exploit a set of baselines for IG. We therefore delve deep into the IG method, rethinking how to construct a set of well-interpretable and consistent baselines. We trace back to the underlying scheme of IG. Although applied in distinct fields, as an attribution method, IG corresponds to a "shortcut" approximation of a cost-sharing method in economics called Aumann-Shapley Value [8] where the model, base features, and attributions of IG are analogous to the cost function, players, and cost-shares of Aumann-Shapley Value respectively [102]. Therefore, under the assumption that the baselines correspond to the coalitions in the Shapley Values, their marginal contributions can be treated as explanations for the example. Consequently, We're inspired to select a set of baselines to approximate the computation of the Shapley Values in a concise and efficient manner, echoing our objective of utilizing a set of baselines.

Given the context, we propose a novel baseline construction method called ***S**hapley **I**ntegrated **G**radients* (**SIG**). In summary, the SIG method considers coalitions in Shapley Value as baselines and employs proportional sampling (aligned with the actual distribution of coalitions) to approximate the computation pathway of Shapley Value.

To assess the effectiveness of SIG, we conduct experiments in two distinct types of environments. (1) GridWorld environment [104], in which we simulate Shapley Value using maze task properties, to further demonstrate the ability of SIG to approximate Shapley Values; and (2) Image classification tasks [43], showcases SIG's capability in providing enhanced and more consistent explanations in classical visual input tasks.

To summarize, our contributions are three-folds: (i) we analyze the drawbacks of using a single baseline for IG and hence a set of baselines is desirable; we then rethink baseline design for IG from the perspective of Shapley Value, and theoretically show that under certain hypothesis, the coalitions in Shapley Values can be regarded as a set of baselines for IG; (ii) we propose a novel variant of Integrated Gradient by constructing a set of baselines named SIG that partially approximates the computational path of the Shapley Value by finding the set of baselines; (iii) we conduct experiments on Gridworld (maze task) and ImageNet (image classification task). The experimental results demonstrate that the proposed SIG provides a new perspective on IG, showing better and more consistent explanations than existing methods.

## Related Work

**Integrated Gradients and Baseline Selection** Integrated Gradients (IG), pioneered by [103], merges gradient implementation invariance with sensitivity analysis, emphasizing the need for an appropriate baseline. [77] highlight that IG's computational path forms a direct trajectory from the selected baseline to the input.

Recent empirical studies [23, 72, 60, 12, 97, 33] offer practical baseline selection guidelines but often lack theoretical depth. For example, [23] uses the average of random dataset samples as the baseline; [33] bases a pixel's baseline on nearby pixels; and [18] treats the baseline as reflective of the background distribution, aligning with our view. Additionally, [32] observes a deviation in IG's computational path from the Shapley Value, while [19] suggests that multiple baselines can mitigate bias inherent in single-baseline attributions.

**Shapley Value** The Shapley Value, originally proposed by [94], offers a fair contribution attribution method in cooperative settings. Distinguished by its alignment with linearity, nullity, symmetry, and efficiency axioms, it remains the only distributive method fulfilling these principles. [9] later expanded this concept to cover infinite games. Its application in model explanation is evident in approaches like Shapley Additive exPlanations (SHAP) by [67], where model features are treated as players in a game, the model as a utility function, and the chain rule to simplify computations.

To address computational challenges, various model-specific Shapley Value approximation methods have been developed, such as DeepSHAP [67], TreeShap [68], and SurvSHAP [59]. Conversely, model-agnostic solutions like FastShap [49] and Monte Carlo Sampling [37, 74] have emerged, targeting efficient estimation across different models.

## Preliminaries

**Notation** Table 2.1 summarizes the detailed notation that will be used in the “Method" section for better understanding.

Table 2.1: Notations involved in SIG

| Symbol | Meaning |
|---|---|
| $x$ | The sample to explain |
| $x'$ | The baseline sample for IG |
| $x_i$ | The ith feature/player of $x$ |
| $r_1$, $r_2$ | Players of $x'$ |
| $S_1$, $S_2$ | Players of $x$ |
| $v()$ | The utility function of SV |
| $F()$ | The model/function to explain and evaluate |
| $t$ | A scalar value indicates a proportion |
| $I$ | A general complete set |
| $N$ | The set of all players in a game |
| $S$ | A coalition in a game |
| $i$ | The indicator of the player of interest |
| $V_i(S)$ | The marginal contribution of $x_i$ to $S$ |
| $w_i(S)$ | The weight of $V_i(S)$ |
| $k$ | A scalar value of the size of $S$ |
| $\hat{w}(k)$ | The normalized $w(k)$ |
| $D$ | The randomly sampled baseline set of SIG |
| $Q$ | Sampling rate of coalitions |
| $B$ | The number of all sampled examples for SIG |
| $\mathcal{A}$ | A tensor of a mini-batch of IG results |
| $\mathcal{V}$ | A tensor of the approximated SV result |

**Integrated Gradients** Integrated Gradients (IG) is a method that was developed to attribute the prediction of a DNN to its input features. It integrates the gradient of the prediction concerning the input features over a straight-line path between the input $x$ and a baseline $x'$. The IG for a model $F$ is expressed as follows:

$$f_{IG}(x_i) = (x_i - x'_i) \times \int_0^1 \frac{\partial F(x' + \alpha(x - x'))}{\partial x_i} d\alpha \tag{2.6}$$

**Shapley Value** Consider a game with $n$ players and a utility function $v$, where $v : 2^N \to \mathbb{R}$ maps subsets of players to real numbers. Any set of players in $N$ is called a coalition $S$. For a given coalition $S$ and a player $x_i$ such that $x_i \notin S$, the marginal contribution of player $x_i$ to coalition $S$ is defined as $v(S \cup x_i) - v(S)$. The Shapley Value of player $x_i$, denoted by $f_{SV}(x_i)$, is then computed as the sum over all coalitions $S \in N/\{i\}$, weighted by the probability of selecting each coalition. Specifically,

$$f_{SV}(x_i) = \sum_{S \in N/\{i\}} \frac{|S|!(|N| - |S| - 1)!}{|N|!} (v(S \cup x_i) - v(S)) \tag{2.7}$$

The Shapley value is calculated by averaging the marginal contributions across all potential coalitions. As the number of features grows, the computation time increases exponentially. To manage this effectively, one approach is to calculate contributions for just a select number of coalition samples. Thus, we are inspired to devise an approximation approach with the assistance of Integrated Gradient.

**Aumann-Shapley Value** With the extension to an infinite game, Aumann-Shapley Value was proposed with the definition that $ds$ represents the infinitely small player in the game, $I$ represents the complete set of players and $tI$ is a perfect sampling, representing a proportion $t$ of all the players. Aumann-Shapley Value can be written as follows:

$$f_{SV}(ds) = \int_0^1 v(tI + ds) - v(tI) dt \tag{2.8}$$

## Method

In this section, we will first present Integrated Gradients (IG) from the perspective of Shapley Value. Through this view, we discover and analyze the shortcomings of IG: the calculation path of IG takes a straight-line shortcut compared to that of Shapley Value. Building upon this observation, we propose our baseline construction method, Shapley Integrated Gradients (SIG).

### Integrated Gradients From the Perspective of Shapley Value

Given an explained sample $x = [x_1, \ldots, x_n]$, a baseline example $x' = [x'_1, \ldots, x'_n]$, and a function $F : R^n \to [0, 1]$ that represents a deep network, we prove that IG is associated with

Aumann-Shapley Value as follows:

**Theorem 1.** *Suppose $x'$ represents the empty set $\emptyset$ with the absence of all features, $x$ represents the complete set $I$ with the presence of all features, and utility function of any sample $x$ is evaluated as $v(x) = F(x) - F(x')$, the integral of Integrated Gradients of all features is a simulation of the integral of Aumann-Shapley Values for all players when the model is evaluated along the linearly interpolated path between baseline and explained sample.*

*Proof.* We treat each feature $x_i$ of sample $x$ as a player in game theory and the mix-up of $x$ and $x'$ in a feature-wise binary on-off manner as a coalition. Thus, the worth of perfect sample $tI$ including coalitions with $t$ proportion of $x$ and $1-t$ proportion of $x'$ is represented as $v(tI)$. The contribution of $ds$ to the coalition is $v(tI+ds)-v(tI)$. Aumann-Shapley Value computes the contribution of an infinitesimal player $ds$ by integrating the functional gain of adding the player to the perfect sample $tI$ of the all-player $I$ at all proportions $t \in [0,1]$. When $t=0$ and $t=1$, we get an $\emptyset$ and a complete set $I$ respectively. Therefore, the worth of the complete set $I$ that represents the integral contribution of all players can be written as follows:

$$\begin{aligned} v(I) &= \int_{\emptyset}^{I} f_{SV}(ds) \\ &= \int_{\emptyset}^{I} \int_{0}^{1} v(tI+ds) - v(tI)dt \\ &= \int_{\emptyset}^{I} \int_{0}^{1} \frac{v(tI+ds) - v(tI)}{ds} dtds \\ &= \int_{\emptyset}^{I} \int_{0}^{1} \frac{dv(tI)}{ds} dtds \end{aligned}$$

Meanwhile, IG accumulates the contribution of each feature by integrating the partial derivatives of model $F$ with respect to the feature at points along a straight-line path $[x', \ldots, x' + \alpha(x-x'), \ldots x]$ from the baseline to the explained sample. As such, the integral of IG overall features from $\emptyset$ to $I$ can be written as follows:

$$\begin{aligned}
\int_{\emptyset}^{I} f_{IG}(dx) &= \int_{\emptyset}^{I} (x_i - x_i') \int_0^1 \frac{\partial F(x' + \alpha(x - x'))}{\partial x_i} d\alpha dx \\
&= \int_0^1 \int_{\emptyset}^{I} (x_i - x_i') \frac{\partial F(x' + \alpha(x - x'))}{\partial x_i} dx d\alpha \\
&= \int_0^1 \nabla F(x' + \alpha(x - x')) d\alpha \\
&= F(x) - F(x') = v(I)
\end{aligned}$$

□

Interestingly, indicated by the proof above, the integral of IG approaches the integral of the Shapley value, but through a different computation path. Each feature in IG plays essentially the same role as a player in Shapley Value. Upon this proof, we use player and feature interchangeably in the following paragraphs.

### Limitations of Integrated Gradients

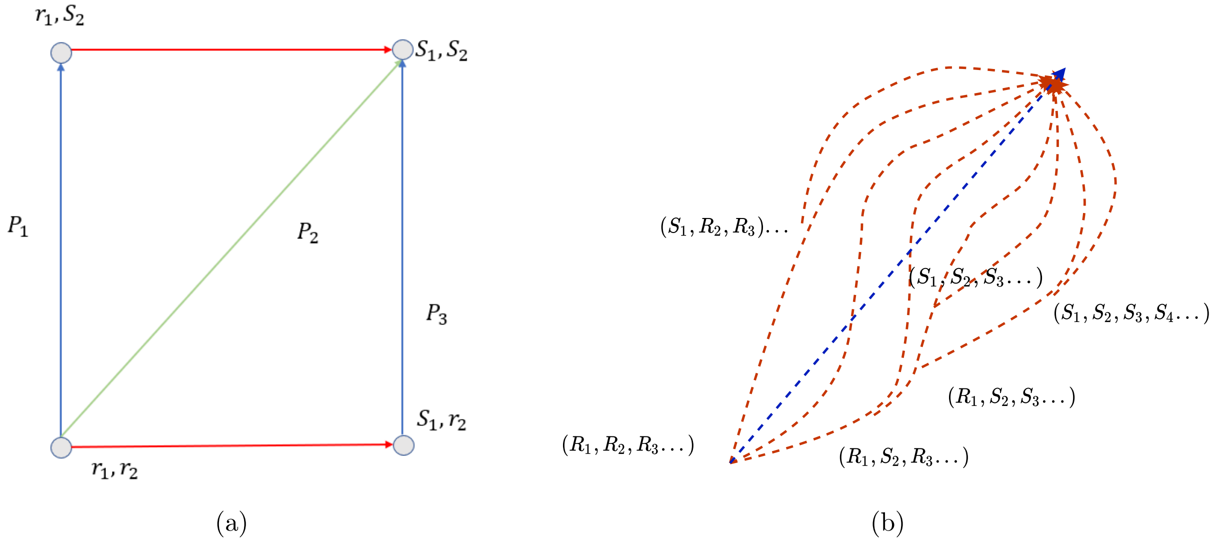


Figure 2.1: (a) For a two-feature input, red lines represent the calculation of Shapley Value for feature $S_1$ and blue lines represent that for feature $S_2$. While Path $P_2$ is the calculation path of IG. (b) Red paths are the calculation path of Shapley Value while the blue path is the calculation path of IG for a $n$-features input.

From the aforementioned Theorem 1, we derive that IG approaches the Aumann-Shapley Value in computing the contribution of the entire feature set. However, the goal of attribution methods is to compute the contribution of each individual feature. We illustrate their difference with a simple example.

As is shown in Fig. 2.1a, we assume that there is a two-feature input of the deep model, denoted as $x = (S_1, S_2)$ with its corresponding baseline as $x' = (r_1, r_2)$. Then the set of coalitions encompasses $(r_1, r_2)$, $(S_1, r_2)$, $(r_1, S_2)$, and $(S_1, S_2)$. The Shapley Value of features $S_1$ and $S_2$ are:

$$\begin{aligned} f\left(S_1\right) &= \frac{\left(v\left(S_1, r_2\right) - v\left(r_1, r_2\right)\right)}{2} + \frac{\left(v\left(S_1, S_2\right) - v\left(r_1, S_2\right)\right)}{2} \\ f\left(S_2\right) &= \frac{\left(v\left(r_1, S_2\right) - v\left(r_1, r_2\right)\right)}{2} + \frac{\left(v\left(S_1, S_2\right) - v\left(S_1, r_2\right)\right)}{2} \end{aligned}$$

From the computation of Shapley Value for $S_1$ and $S_2$, we observe that calculation paths traverse through all four points of coalitions, represented by the blue and red lines in Fig. 2.1. The calculation paths of Shapley Value are as $P_1$ and $P_3$. On the other hand, as the proof shows, we ascertain that IG actually computes contribution along the straight line $P_2$. It becomes apparent that the calculation path of IG expressed as $P_2$, takes a shortcut compared to the path of Shapley Value, shown as $P_1$ and $P_3$.

Extending to the n-features setting, as shown in Fig. 2.1b, the computation of Shapley Value takes many polyline paths, while the calculation of the IG approximates it by taking a straight line shortcut between the baseline and the explained sample. Consequently, this discrepancy may result in inaccurate attributions of individual features.

### Shapley Integrated Gradients

In the above subsection, we observe that IG may lead to inaccuracy in estimating the contributions for features caused by the shortcut of the calculation path compared to the Shapley Value. However, because of the single-path computation, its efficiency is obvious. On the contrary, the Shapley Value is appreciated for its accuracy in determining individual contributions, and its computational burden of leveraging all coalitions, subject to $\mathcal{O}(2^n)$ where $n$ is the number of features, restricts its application. To take some benefits of both, we combine Shapley Value with IG and propose ***Shapley Integrated Gradients*** (SIG), a novel baseline construction method for improving IG from the perspective of Shapley Value.

The intuition of the algorithm comes from Equation 2.6. As shown, the Shapley Value of feature $i$ is the weighted average of marginal contributions of a player across all possible coalitions $\{S; S \in N/\{i\}\}$, where the weight $w_i(S)$ and marginal contribution $V_i(S)$ are defined as:

$$w_i(S) = \frac{|S|!(|N| - |S| - 1)!}{|N|!} \tag{2.9}$$

$$V_i(S) = v(S \cup x_i) - v(S) \tag{2.10}$$

By inspecting the above two factors, here are some interesting observations:

**Observation 1.** *Given a game $N$, $w_i(S)$ is only dependent on the number of players $k = |S|$ in the coalition $S$. Therefore, we can merge $\{w_i(S); \forall S \text{ s.t. } |S| = k \wedge \forall i, i \in N\}$ as a universal weight value $w(k)$. This value can be precomputed as long as we decide $k$.*

**Observation 2.** *The sum of $\{w(k); \forall S \text{ s.t. } |S| = k \wedge \forall k, k \in [0, |N| - 1]\}$ is $1$ since $w(k)$ represents the inverse value of the number of combinations $\binom{|N|-1}{|S|-1}$ times the number of different values of $k$ we can choose from the game. $k$ acts like the proportion $t$ in Equation 2.7 to measure the magnitude of coalitions. Thus, all coalitions are naturally categorized into $|N|$ groups subjected by the size $k$ and $\sum_S w(k) = 1/|N|$.*

$$\begin{aligned}\frac{1}{w(k)} &= \frac{|N|!}{|S|!(|N| - |S| - 1)!} \\ &= |N| \times \frac{|N-1|!}{k!(|N| - 1 - k)!} \\ &= |N| \times \binom{|N| - 1}{k}\end{aligned}$$

**Observation 3.** *As a direct conclusion of Theorem 1, the marginal contribution of adding $i; \forall i \in N$ to $S$ can be approximated by using IG from baseline $S$ to complete set $I$.*

In the definition of Shapley Value in Equation 2.6, the marginal contributions of all the coalitions $S$ are computed and then weighted by their corresponding $w_i(S)$ whose quantity is exponentially large. However, thanks to the observations above, we leverage a “trick” called **proportional sampling**, as is shown in Fig. 2.2a, to circumvent it.

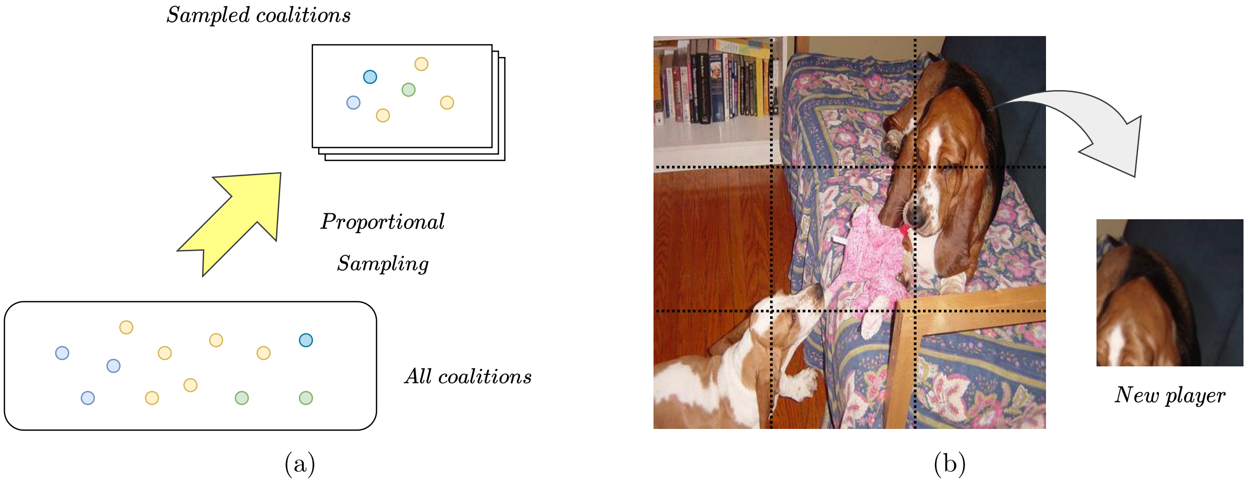


Figure 2.2: (a) Proportional sampling in our SIG. Different colored nodes represent different weights defined by Equation 2.9. (b) Construction of new players. A patch of pixels is considered a new player/feature on which we search the baseline set.

**Proportional Sampling.** Instead of sampling coalitions from a binomial distribution and weighting them by their corresponding $w_i(S)$, an alternative approach is to sample coalitions with a probability that is proportional to the pre-computed $w_i(k)$ and weight them uniformly by 1. In short, the expectation of *proportional sampling* × *uniform weights* is equal to the expectation of *proportional weights* × *uniform sampling*. The reason behind it is sampling from all coalitions for a game-playing task is time-consuming and sometimes appears infeasible in practice. Therefore, we initially randomly sample a proportion $Q$ of coalitions. From this subset, we then sample $N$ baselines. We provide a proof of it as follows.

**Proof.** We prove that our proposed **proportional sampling** is an unbiased estimator of the true Shapley Value. Let $f_{SV}^{PS}(x_i)$ denote the estimated Shapley Value of $x_i$ computed leveraging proportional sampling. $p_i(S)$ is the probability of $S$ being sampled. $V_i(S)$ is the marginal contribution. $C_k$ is the value of $\binom{|N|-1}{k}$. Then $p_i(S) = \frac{1}{|N|}p_i(k) = \frac{1}{N \times C_k}$. Suppose we sampled for $M$ times $S$ based on probability function $p_i(S)$,

$$
\begin{aligned}
\mathbb{E}(f_{SV}^{PS}(x_i)) &= \frac{1}{M}\sum p_i(S)V_i(S) \\
&= \frac{1}{M}\sum \frac{1}{|N|}\sum_{k=0}^{N-1} p_i(k)V_i(S;|S|=k) \\
&= \frac{1}{M}\sum \frac{1}{|N|}\sum_{k=0}^{N-1} \frac{1}{C_k}\sum_{j=0}^{C_k} V_i(S_j) \\
&= \frac{1}{M}\sum \sum_{S\in N/\{i\}} \frac{1}{|N|\times C_k}V_i(S) \\
&= \frac{1}{M}\sum 1\cdot \sum_{S\in N/\{i\}} w_i(S)V_i(S) \\
&= \mathbb{E}(\sum_{S\in N/\{i\}} w_i(S)V_i(S)) = \mathbb{E}(f_{SV}(x_i))
\end{aligned}
$$

We justify the “trick” to create an unbiased estimator of Shapley Value briefly. Due to Observation 1, the weight of coalition $S$ can be precomputed once we select a $k$, and it is a constant no matter which specific coalition is drawn as long as its size is $k$. Due to Observation 2, the expected marginal contribution of different sizes is equally treated for different $k$ values. Therefore, the prior importance of each coalition is universally 1. Suppose we draw an infinite large number of samples, the expectation of $V_i(S)$ drawn proportional to weight $w(k)$ is an infinite approach to that of $V_i(S)$ weighted by $w(k)$.

In addition, Observation 3 enables reducing the computation burden of computing the marginal contributions of all players $i$ one by one but completing them at once through IG. Since IG is a gradient-based approach which is supported by GPUs, leveraging it gives SIG more potential to speed up.

To sum up, the three observations inspire our proposed SIG algorithm that constructs a set of sampled coalitions with different magnitudes $k$ as baselines of IG to approximate Shapley Value in an efficient way.

### SIG Algorithm

We posit that a coalition in Shapley Value corresponds to a baseline in IG. We aim to obtain relative Shapley Value since people typically care more about relative contributions of features rather than absolute contributions. The overall algorithm, termed as *Shapley Integrated Gradients* (SIG), is presented in Algorithm 1.

**Algorithm 1** Shapley Integrated Gradients (SIG)

```
INPUT: explained sample $x$, default sample $x'$, model $F$
PARAMETER: player set $N$, sample rate $Q\%$, sample size $B$
OUTPUT: approximated Shapley Values $\mathcal{V}$
Initialize baseline set $D = \emptyset$
for $k \leftarrow 0$ to $N-1$ do
    Compute weight $w(k)$ as Equation 2.9
end for
Normalize all weights $\hat{w}(k) = w(k)/\sum_{k=0}^{N-1} w(k)$
Sample $Q\%$ of all coalitions according to $\hat{w}(k)$ as set $S(N)$
while $|D| < B$ do
    Randomly sample a coalition $S$ from $S(N)$
    Construct a baseline $d$ from $S$ based on $x$ and $x'$
    Add $d$ to $D$
end while
Compute mini-batch IG $\mathcal{A} = f_{IG}(x, D, F)$ as Equation 2.8
Average through mini-batch $\mathcal{V} = \frac{1}{B}\sum_{i=1}^{B} \mathcal{A}_i$
return $\mathcal{V}$
```

There are two key points to note in the algorithm. (i) There doesn't have to be a one-to-one mapping from a player to a feature in explained sample $x$. As is shown in Fig. 2.2b, we treat a patch of pixels as a single player to construct baseline sets. (ii) The construction of a baseline $d$ of coalition $S$ is by replacing the default values of $x'$ in $S$ with corresponding values of $x$. The default value is dependent on the task which we will specify in the experiments.

There are two sources of discrepancies between our SIG and Shapley Value: (i) We only collect a subset of coalitions, with the omission of many other coalitions; (ii) The marginal contribution approximated by IG is a deviated from the accurate definition of Shapley Value.

Finally, compared with accurately computing Shapley Values, the benefits of using SIG are as follows:

- SIG is more suitable in a mini-batch computation setting by leveraging powerful parallel computation devices such as GPUs which is prevalent in current large-scale applications.

- SIG has the flexibility for users to define a smaller sample size instead of collecting all coalitions.
- SIG supports user-defined players (grouping features) or feature-wise dense estimation of Shapley Values.

## Experiments

In this section, our objective is to further validate the soundness of the SIG by exploring the answers to the following questions.

- **Question 1:** Does SIG provide a better approximation of the Shapley Value than other IG methods and Shapley Value approximations?
- **Question 2:** Does SIG provide more insightful explanation capabilities than other baseline methods in complex tasks with visual inputs?

### Simulation of Shapley Value – GridWorld

- **Simulation of Shapley Value** involves a simple GridWorld environment [104] comprising 2 distinct sub-tasks to evaluate our SIG's ability to simulate Shapley Value for Question 1;
- **Performance of Explanation** focuses on the classical image input-based tasks: image classification task employing ResNet model [43] on ImageNet Dataset [27] for Question 2.

As shown in Fig. 2.3, we construct 2 × 2 GridWorld and 2 × 3 GridWorld to simulate Shapley Value. In both tasks, we first utilize the *reward function* as the utility function in the definition of Aumann-Shapley Value. The *reward function* operates as follows:

- For each step the agent takes before reaching the terminal goal for the first time, it receives a reward of -1.
- Upon reaching the goal at any instance, the agent is awarded a reward of $+1$.

In GridWorld, the agent performs valid actions (that do not result in out-of-bounds) based on its state to maximize the cumulative rewards. The simplicity of GridWorld facilitates

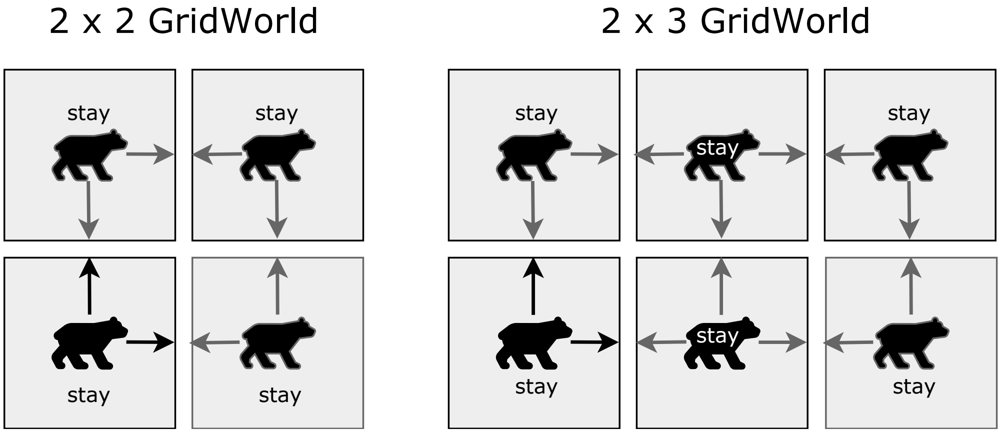
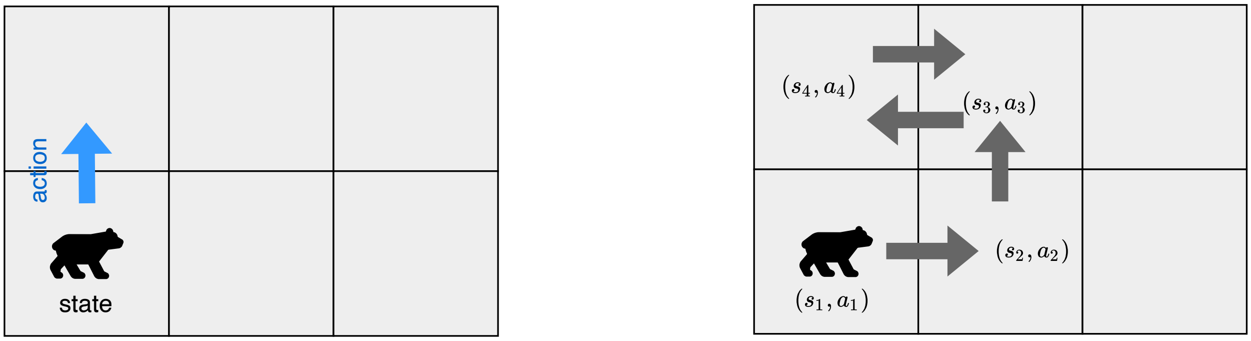


Figure 2.3: 2×2 and 2×3 GridWorld tasks. The agent needs to find a path from the start that leads to the goal.



(a) Player

(b) Coalition

Figure 2.4: Visualization of players and coalitions in GridWorld.

the computation of Shapley Values, making it an ideal simulation environment to answer **Question 1**.

In this setting, we define players and coalitions based on GridWorld's maze task properties. Specifically, we define the state-action pair $(s, a)$, i.e., the agent at state $s$ executes action $a$ as a player, as shown in Fig. 2.4a. We further define a coalition as a combination of time steps. For instance, consider a coalition $S$ comprising 4 time steps: $\{(s_1, a_1), (s_2, a_2), (s_3, a_3), (s_4, a_4)\}$, as shown in Fig. 2.4b. From coalition $S$, we can generate $4 \times 3 \times 2 \times 1 = 24$ permutations, $i \in \{1, 2, \cdots, 24\}$, such as $T_1 = \{(s_4, a_4), (s_2, a_2), (s_3, a_3), (s_1, a_1)\}$ and $T_2 = \{(s_3, a_3), (s_2, a_2), (s_4, a_4), (s_1, a_1)\}$, among others, where $T_i$ denotes a trajectory.

Essentially, our reward function focuses on these permutations or trajectories. It assesses each permutation or trajectory and assigns rewards accordingly. For example, if trajectory $T_1$ encounters a scenario where the agent cannot transition from state $s_4$ to state $s_2$ using action $a_4$, the agent is assumed to remain at state $s_1$ without taking any action until the game concludes. The utility function for coalition $S$ is defined as the maximum cumulative rewards achievable from any trajectory derived from coalition $S$.

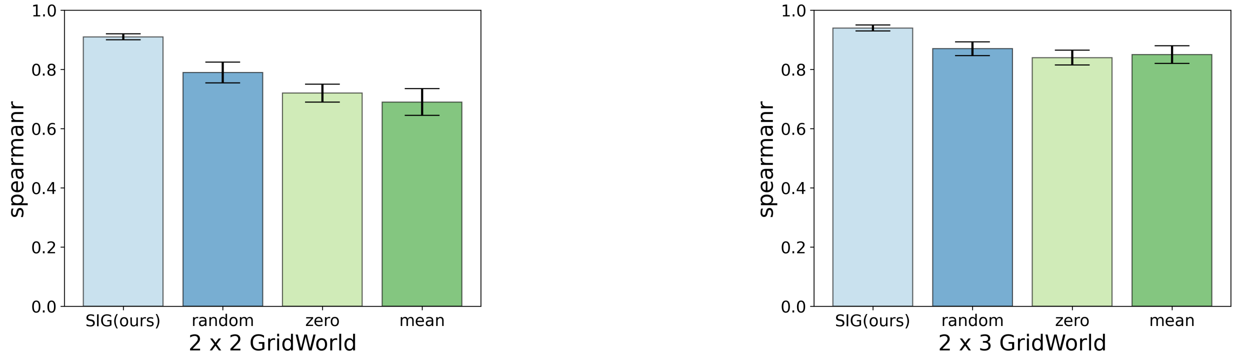
Based on the above definition, we define 12 players, i.e., 12-time steps, in the 2 × 2 GridWorld (4 states, 3 valid actions per state, as shown on the left in Fig. 2.3), and 20 players, i.e., 20-time steps, in the 2 × 3 GridWorld (6 states, 4 valid actions in the middle two states, and 3 valid actions in the other states as shown on the right in Fig. 2.3), to serve as the two Shapley Value conditions, respectively.

**SIG Performance** Further, in order to objectively assess the interpretability of SIG in GridWorld tasks, we select three general baseline methods and five Shapley Value approximation techniques for performance comparisons.

**(1) Baseline Methods:** (i) **random baseline** selects baseline samples randomly from the set being explained; (ii) **zero baseline** sets each feature's baseline value to zero; (iii) **mean baseline** calculates each feature's baseline as the average across the sample set.

**(2) Shapley Value Approximation Methods:** (i) **Deep Shap (DS)**: a unified model prediction interpretation method [67]; (ii) **Owen Shap (Owen)**: a multilinear sampling algorithm for Shapley Value estimation [78]; (iii) **Shapley Value Sampling (AS)**: assessing feature importance using Shapley Values and averaged outputs from feature permutations [55]; (iv) **Fast Shap (FS)**: a real-time Shapley Value approximation approach [49].

To reasonably evaluate the interpretability of each method, we choose the *Spearman's rank correlation coefficient* [35] as the quantitative metric to measure the order of players' contributions. This metric evaluates the similarity between the computed Shapley Value order and the actual Shapley Value order. In investigating the consistency of SIG under varying Shapley Values, we conduct 10 independent experiments to minimize the chance of drawing false conclusions from unlikely events.



(a)

(b)

Figure 2.5: Comparative performance of SIG and baseline methods when $\mathbf{Q}$ = 40% and $\mathbf{B}$ = 200. (a) The average *Spearmanr metric* for 2 × 2 GridWorld; (b) The average *Spearmanr metric* 2 × 3 GridWorld. The black line indicates the variance of the *Spearmanr metric*.

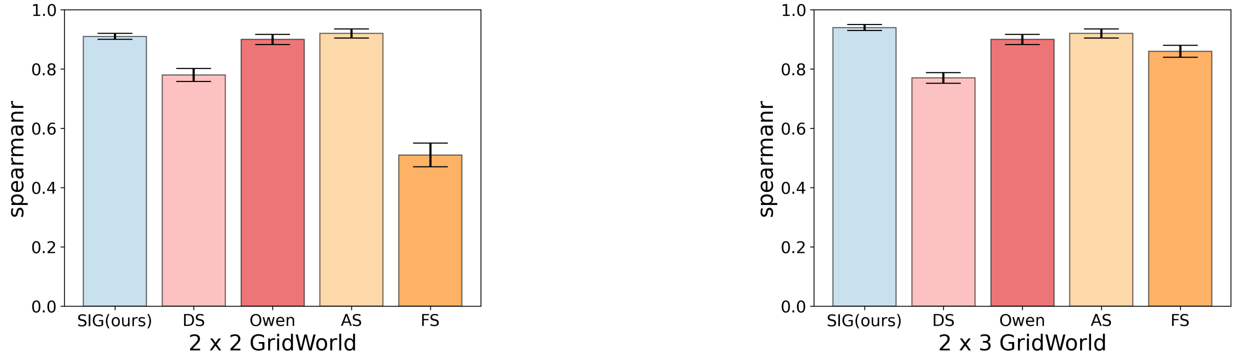


Figure 2.6: Comparative performance of SIG and Shapley Value approximation methods when $Q$ = 40% and $B$ = 1500. (a) The average *Spearmanr metric* for 2 × 2 GridWorld; (b) The average *Spearmanr metric* 2 × 3 GridWorld. The black line indicates the variance of the *Spearmanr metric*.

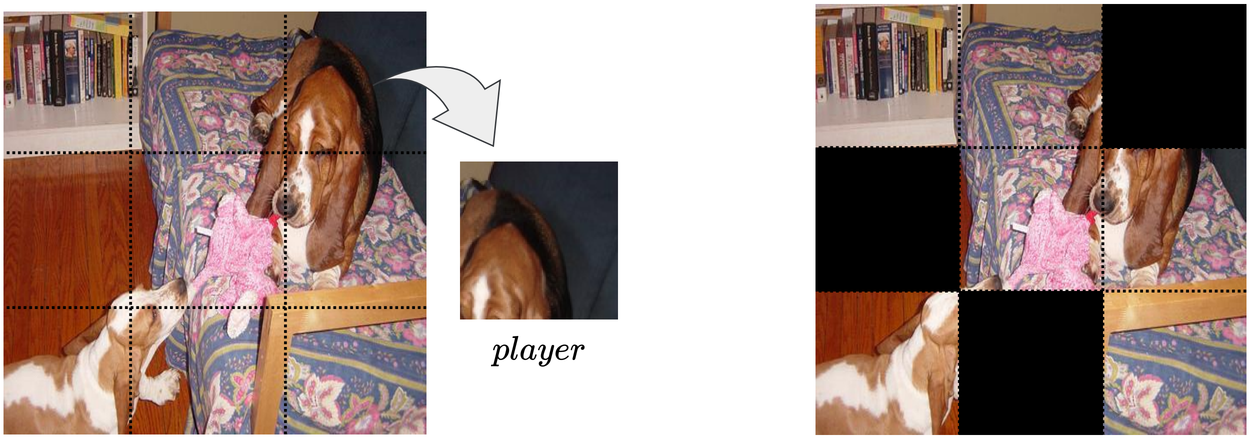
The comparative performance of SIG and the baseline methods is displayed in Fig. 2.5. As shown in Fig. 2.5a and Fig. 2.5b, SIG exhibits a higher average *Spearmanr metric* compared to the other three baseline methods across different Shapley Value conditions. This indicates that the order computed by SIG is more closely aligned with the actual Shapley Value order of players.

Fig. 2.6 presents the comparative performance of SIG and other Shapley Value approximation methods. Similar to the conclusions drawn from the comparison experiments with the baseline method, our SIG also ranks among the top tier when compared to the Shapley Value approximation methods.

Moreover, the performance in Fig. 2.5 and Fig. 2.6 demonstrate that the variance of SIG is lower than that of the other three baseline methods and the five Shapley Value approximation methods. This is further evidence of the consistency of SIG.

### Performance of Explanation – ImageNet

We further conduct experiments in the image classification task. In this experiment, we aim to find an answer to **Question 2** by employing a quantitative metric to evaluate the algorithmic performance in the classical image classification task.



(a) Player

(b) Coalition

Figure 2.7: Visualization of players and coalitions in ImageNet.

Specifically, we regard the pre-trained ResNet-18 [43] model with Pytorch on the ImageNet dataset as the utility function and whether the model classifies the image correctly as a utility. To simplify the process, we follow [22] to consider a patch of pixels as a single player, thereby streamlining the computation of contributions.

Concretely, we divide an image into 9 areas with 3 rows and 3 columns, taking one area as a player and multi areas as a coalition as shown in Fig. 2.7.

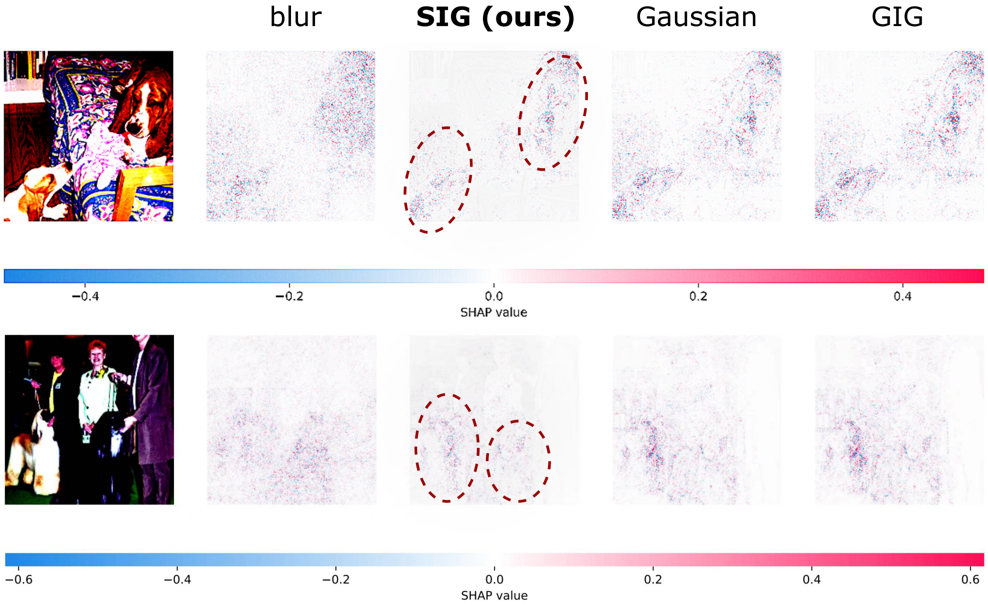


Figure 2.8: Saliency maps of four baseline methods in the image classification task for dogs. Our SIG method predominantly focuses on dogs, whereas other methods disperse their attention to areas outside the dogs.

Table 2.2: The iAccuracy of removing the top 10 most important patches of sampled images from the ImageNet dataset.

| Methods | **SIG(ours)** | Gaussian | blur | GIG |
|---|---|---|---|---|
| iAcc | **0.926** | 0.948 | 0.931 | 0.929 |

We adopt three baseline methods for IG as recommended by [100]: (i) **Blurred Baseline**: using a blurred image version to signify missing information; (ii) **Gaussian Baseline**:

Table 2.3: Running time of baseline methods in image classification task. The unit of running time is second (s). The construction of the baseline set and the execution of batch operations within the model take the most time.

| | **SIG(ours)** | blur | GIG | Gaussian |
|---|---|---|---|---|
| Environment | *Running Time (s)* | | | |
| **ImageNet** | **1.23 ± 1.88** | 0.80 ± 0.06 | 5.24 ± 2.5 | 0.77 ± 0.13 |

employing a gaussian distribution centered on the current image; (iii) **Guided Integrated Gradients (GIG)** [52]: adjusting the attribution path as the baseline.

In our experimental setup, each baseline method computes feature contributions for each output neuron (i.e., class) of the model. To minimize computational cost, we focus on the classes with the highest probability.

Fig. 2.8 illustrates the saliency map of the four methods in the dog classification task. It can be seen that SIG focuses mainly on the dog even when there are multiple other objects in the image such as mats, beds, people, etc., whereas the other baseline methods either do not focus well on the dog's features or shift their attention to background regions other than dogs.

We consider *iAccuracy metric* adapted from [62] as a quantitative metric to highlight the impact of features on model predictions. We sort the contribution scores from high to low and gradually remove the most important patches based on the sorted results. Then, we feed the images to the ResNet model and observe if the prediction remained the same as unchanged images. When more patches are removed, the prediction of the model deviates more from the unchanged prediction and we draw a curve of prediction change with respect to the number of removed patches. The iAccuracy metric specifically means that the area under the curve. More formally, the iAccuracy metric is defined as $iAcc(L) = \frac{1}{L+1}(\sum_{k=0}^{L} \mathbf{1}_{F(x^0)=F(x^k)})$, where $x^k$ is noted as the image $x^0$ removing the top $k$ patches attributed by the baseline methods. To strike a balance between the workload and reliability, we have carried out experiments on 1000 randomly selected images. As shown in Table 2.2, the iAccuracy score is smaller than the other three baseline methods. A smaller score means that when the same amount of key patches are removed from the images, the model's prediction deviates more from the original image. In other words, SIG finds out the more important patches than the other baseline methods.

Additionally, we assess the execution times of various baseline methods in image classification tasks. As indicated in Table 2.3, the runtime for our Shapley Integrated Gradients (SIG) method shows a modest increase compared to single baseline approaches like the blur baseline. However, it remains significantly more efficient than Gradient-based Input Gradients (GIG), which require considerable time to generate baselines. Notably, the runtime of SIG is substantially lower than that of single baseline methods within image classification tasks, primarily due to the batch processing capabilities of ResNet. At this point, combining the visualization of the saliency map shown in Fig. 2.8 with the quantitative analysis shown in Table 2.2, we can provide a **positive response** to **Question 2**.

## Summary

**Discussion** In summary, by exploring the answers to questions 1 and 2, we experimentally validate the capabilities of the SIG: the ability to correctly approximate Shapley values, improved interpretability, and the ability to provide consistent explanations for visual input tasks. Nevertheless, there remain intriguing challenges that deserve further research. For example, our current creation of players using the patches of pixels method is slightly coarse; the method for fitting Shapley Value proportions exhibits considerable randomness; timing performance can be further enhanced. Addressing these issues is an important goal for our ongoing and future research.

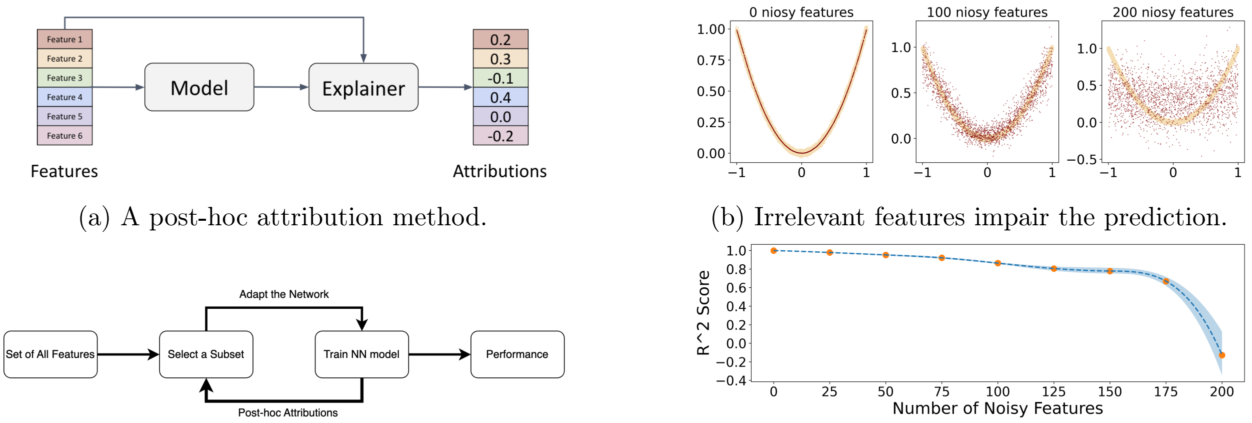
**Conclusion** In this work, we rethink the baseline of Integrated Gradients (IG) from the perspective of Shapley Value. We observe and theoretically analyze that IG can be viewed as Auman-Shapley Value under certain assumptions. Specifically, a set of baseline aligns with the coalitions in the Shapley Value, thus tackling the challenges of exploiting single baselines. Therefore, we propose a novel baseline construction method SIG, which is a hybrid of the IG and Shapley Value that provides an improved and consistent explanation compared to existing baseline methods for IG. Experimental results in Gridworld tasks and the image classification tasks validate SIG's ability to approximate Shapley Value, better interpretability, and consistent explanations.

## 2.2 A Benchmark to Evaluate Post-Hoc Local Attribution Methods in Low SNR Environments

This study [96] examines the efficacy of post-hoc local attribution methods in identifying features with predictive power from irrelevant ones in low signal-to-noise ratio (SNR) environments, common in real-world machine learning applications. We developed synthetic datasets encompassing symbolic functional, image, and audio data, incorporating a benchmark on the *(Model* × *Attribution* × *Noise Condition)* triplet. By rigorously testing various classic models trained from scratch, we gained valuable insights into the performance of these attribution methods under multiple conditions. Based on these findings, we introduce a novel extension to the recursive feature elimination (RFE) algorithm, enhancing its applicability for neural networks. Our experiments highlight its strengths in prediction and feature selection, alongside limitations in scalability. Further details and additional findings are included in the appendix, with extensive discussions. The codes and resources are available at URL.

### Introduction

Machine learning success typically hinges on two complementary strategies: (I) identifying the most predictive features for learning, referred to as the data-centric approach [119], and (II) training the model to approximate optimal weights, known as the model-centric approach [79]. Both strategies are crucial for reducing generalization errors in predictive tasks, with feature engineering playing an essential role [44]. Noisy or irrelevant features are prevalent in real-world applications [13]. Due to their robustness against noise, neural networks have become a common choice for analyzing low signal-to-noise ratio (SNR) data across various domains, including finance [92], clinical settings [47], and scientific research [21]. While black-box models often suffice for multimedia data such as online images, videos, and text posts, low SNR domains demand high levels of explainability [87], underscoring the critical need for transparent methodologies. Post-hoc local attribution methods, which assign importance scores to individual features [4], are widely utilized to elucidate neural network preferences regarding input features. Despite their popularity, there is a paucity of *rigorous quantitative* empirical research examining the ability of these methods to effectively differentiate between features with strong predictive capabilities and those that are irrelevant. This gap in the literature motivates our study.



(a) A post-hoc attribution method.

(b) Irrelevant features impair the prediction.

(c) The adapted RFE method.

(d) The robustness to noisy features of NN.

Figure 2.9: A teaser figure of the approach (on the left) and challenge (on the right) of this work. In (a) and (c), the attributions are scalar weights assigned to features via a one-to-one mapping in a post-hoc manner. In (b) and (d), only one feature is predictive as defined by $y = x^2$.

In this research, we conduct an empirical analysis to assess the effectiveness of using post-hoc attribution methods to differentiate between predictive and irrelevant features. Our study yields several noteworthy findings: (I) Gradient-based saliency alone is sufficient for feature selection, offering high precision, convergence, and low cost; (II) A significant positive correlation exists between the efficacy of post-hoc attribution methods and the generalization capabilities of the predictive model; (III) Neural networks are less susceptible to structural noise compared to random noise; (IV) Neural networks more effectively identify predictive features at fixed positions than those randomly distributed. Building on these insights, we further explore the inherent robustness of neural networks and the discriminative capacity of post-hoc attribution methods to enhance the recursive feature elimination (RFE) technique [20]. Our contributions are three-fold:

- We created synthetic datasets for symbolic functional, image, and audio analysis, systematically blending predictive and irrelevant features. These datasets serve as accessible resources for researchers exploring this domain, facilitating downstream empirical studies.

- We evaluated the effectiveness of several well-known post-hoc attribution methods across various (Model × Attribution × Noise Condition) combinations within the curated datasets, uncovering several important but previously unnoticed insights.

- We adapted the Recursive Feature Elimination (RFE) strategy, traditionally applied to transparent models such as linear models, SVMs, and decision trees, for use with neural networks. Our empirical results highlight both the strengths and limitations of this approach.

## Benchmark Procedure

The general procedure of the benchmark includes data generation, ground truth annotation, metrics definition, model training, and post-hoc attribution methods evaluation. This section focuses on data generation and metrics defining.

### Data Generation

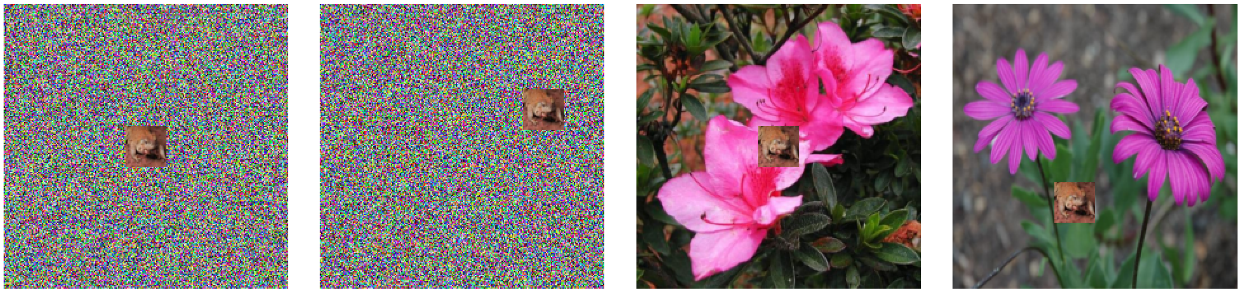
We generated symbolic functional, vision, and audio data for downstream empirical studies to benchmark post-hoc attribution methods in various conditions. One novel and intriguing property of our synthetic dataset is the design of the *(Model × Attribution × Noise Condition)* triplet. Beyond the *(Model × Attribution)* paradigm adopted by other benchmarks, a noise condition factor is introduced. We designed the data generation and empirical study to address the following questions: Among the three factors, how does each affect the predictive feature identification ability of a post-hoc attribution method?

To avoid misunderstanding about the triplet, we elucidate the concepts in this context separately:

**Model** : A model embodies a trained checkpoint affected by the architecture (e.g., CNN-based, Transformer-based), the configuration (e.g., widths and depths of a model), and the hyper-parameters of training (e.g., learning rate, dropout rate).

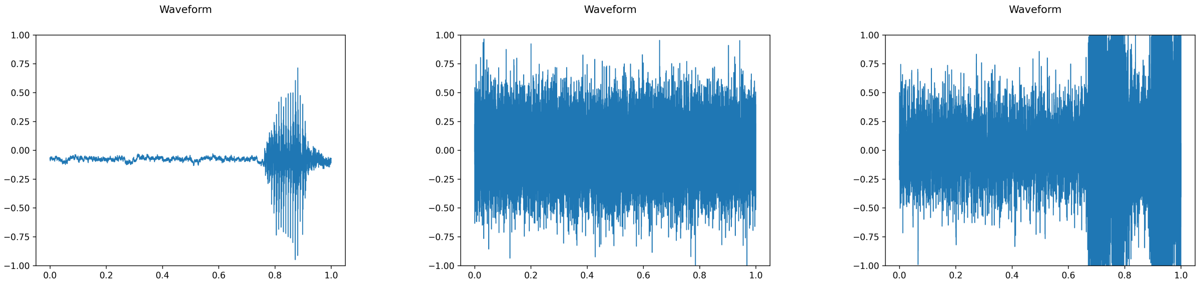
**Attribution** : Among the various feature attribution methods, we specifically study Saliency (SA), DeepLift (DL), Integrated Gradient (IG), and Feature Ablation (FA), which are model-agnostic to all NN-based models for fair comparison across models. SA is the pure gradient. DL is a backpropagation-based approach. IG is a gradient-based approach referring to baseline data. FA is a perturbation-based approach. Detailed definitions are provided in the appendix.

**Noise Condition** : Noise conditions include but are not limited to the type of noise, the signal-to-noise ratio, the magnitude of label noise, and the way that features are aligned across instances.



(a) RBFP (b) RBRP (c) SBFP (d) SBRP

Figure 2.10: The examples of synthetic vision data and saliency maps of attribution methods. The foreground images can be placed at a fixed position (center) across instances or randomly. The background images can be generated by Gaussian noise or images of flowers.



(a) Speech command (b) Gaussian noise (c) Rainforest connection species

Figure 2.11: The examples of sources to construct synthetic audio data. Figure (a) is the foreground predictive feature while (b) and (c) are background features that are irrelevant to the classification task.

We enriched the context by generating three different types of synthetic data for broader interest:

- Symbolic Functional Data: Based on human-designed symbolic functions with ground truth annotations derived from math formulas, used to study the general behaviors of multilayer perceptron (MLP) networks on regression tasks.
- Vision Data: Used to study popular architectures for visual scene classification tasks, with noisy input samples in the form of 224×224 images.
- Audio Data: Used to study popular architectures for sequential data classification tasks, with noisy input samples being 10-channel waveform audio sequences.

In addition to traditional metrics like accuracy and mean absolute error (MAE), we introduce two additional metrics for a more comprehensive evaluation.

**Uniform Score (UScore)** is a modified version of the Mean Absolute Error (MAE) that normalizes it into the range $(0, 1)$. We employ this metric to assess the proximity of predictions to the true symbolic values. We prefer the UScore over MAE in our multiple regression tasks because using MAE could result in excessively varied scales across different tasks. To ensure a fair summary and consistent comparison, we propose the Uniform Score, which is defined as follows:

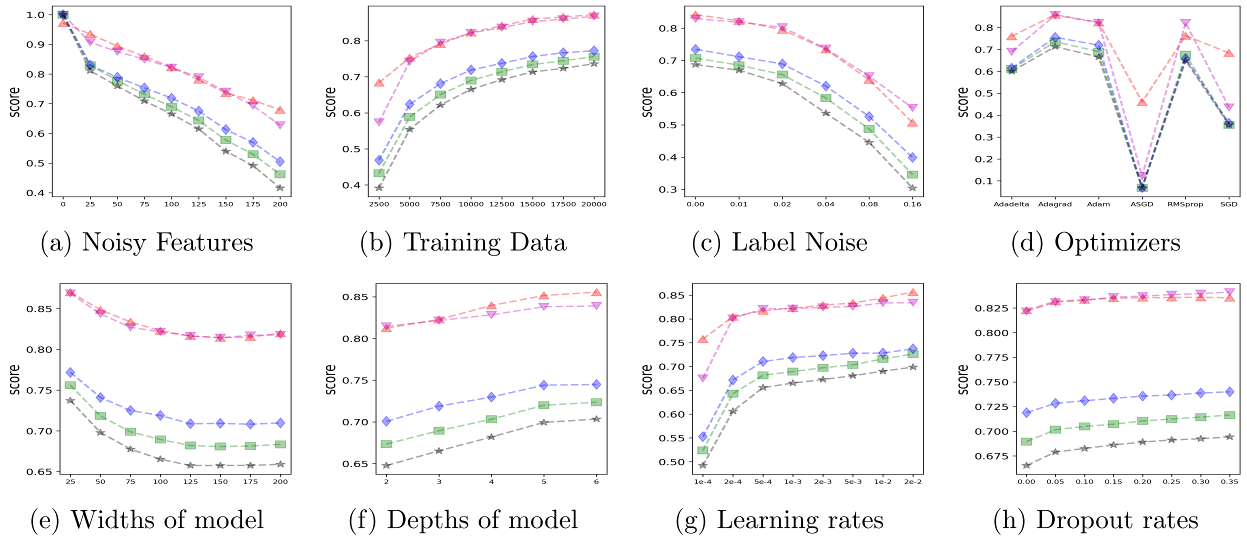
$$UScore = \frac{1}{N}\sum_{i=1}^{N}(1 - \frac{|\hat{y}_i - y_i|}{|\hat{y}_i| + |y_i| + \epsilon}), \tag{2.11}$$

where $\hat{y}_i$ represents the predicted value and $y_i$ denotes the ground truth target value for the $i$-th instance in a dataset consisting of $N$ samples.

**Functional Precision (FPrec)** quantifies the overlap between the $k$ predictive features given by annotation and those deemed top-$k$ important by a model, as ranked by the post-hoc attribution method. This approach is akin to the feature agreement measure introduced by [57], effectively integrating both precision and recall aspects into a single metric.

$$FPrec = \frac{|\{\text{top-k features of model}\} \cap \{\text{k predictive features}\}|}{k} \tag{2.12}$$

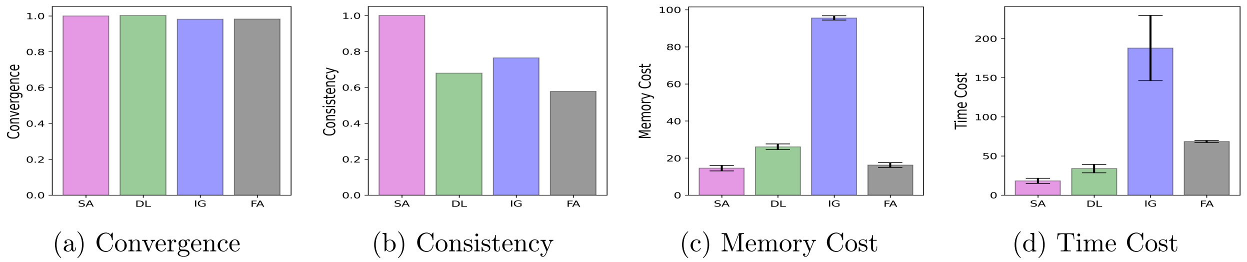
## Experiments and Insights



(a) Noisy Features (b) Training Data (c) Label Noise (d) Optimizers

(e) Widths of model (f) Depths of model (g) Learning rates (h) Dropout rates

Figure 2.12: Experimental results on symbolic functional data using MLP regressors differentiated by varying factors. For each subplot, we only change one factor from the default configuration. ▲ denotes the *UScore* of the predictions. ▼, ■, ♦, and ★ denote the *FPrec* of SA, DL, IG, and FA methods respectively.



(a) Convergence (b) Consistency (c) Memory Cost (d) Time Cost

Figure 2.13: The results of our simulation experiments. Convergence is quantified by the area under the FPrec curve across 300 training epochs, while consistency is assessed through the average agreement between the top-k most important features of a sample and the average importance of the entire dataset. Both convergence and consistency scores are normalized so that the value for SA is set to 1. Additionally, we report the average and standard deviation of the memory and time costs incurred at the test stage.

Building on the benchmark pipeline, we conducted a series of evaluation experiments. This section discusses experimental results and observations, addressing several pertinent research questions and providing insights for future studies. We present experiments for each modality separately. Experiments are repeated five times with random seeds.

### Symbolic Functional Data Experiment

We conducted experiments to assess the performance of neural network models and attribution methods under various configurations. Each experiment altered only one aspect of our standard setup to isolate the impact of individual factors. Our default configuration included a dataset size of 10,000, a 4:1 train-test split, 100 noisy features, label noise set at 0.01, model dimensions with widths of 100 and depths of 3, an Adam optimizer, a learning rate of 0.001, no dropout, and a training duration of 1000 epochs targeting mean squared error loss. We reported the UScore of the model and FPrec of the attribution methods.

From the analysis of plots, we observed consistent trends across most figures, with a few exceptions like FA. Two key insights emerged: (I) SA consistently outperforms other attribution methods in low SNR environments, as measured by FPrec. (II) The effectiveness of all XAI methods is closely tied to the model's predictive capabilities. Additionally, enhancements in regression model performance were noted with reductions in noisy features and label noise and increases in dataset size, model depth, learning rate, and dropout rate. However, wider models, despite having greater capacity, showed diminished predictive accuracy, possibly due to more neurons in a layer learning to memorize noise features and their nuanced internal correlation rather than the real underlying patterns. In tests with default Pytorch optimizers, ASGD was notably ineffective at learning weights due to gradient oscillation.

We also aimed to determine which attribution methods converge more rapidly across epochs, maintain greater consistency across samples, and utilize fewer computational resources. All four methods exhibited similar convergence rates corresponding to model training progression. SA demonstrated significantly better consistency compared to the other three methods. In terms of computational efficiency, IG consumed considerably more memory and time, while SA proved the most resource-efficient. Thus, we concluded that the naive SA

Table 2.4: Experimental results (Top-1 classification accuracy and attribution IoU) on synthetic vision data with random background noise.

| Architecture | GFLOPs | RBFP | | | | | RBRP | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | **Pred**(ACC%) | **SA**(IOU) | **DL**(IOU) | **IG**(IOU) | **FA**(IOU) | **Pred**(ACC%) | **SA**(IOU) | **DL**(IOU) | **IG**(IOU) | **FA**(IOU) |
| AlexNet [58] | 0.71 | $10.00_{\pm 0.000}$ | $0.021_{\pm 0.000}$ | $0.011_{\pm 0.101}$ | $0.012_{\pm 0.086}$ | $0.021_{\pm 0.000}$ | $10.00_{\pm 0.000}$ | $0.021_{\pm 0.000}$ | $0.018_{\pm 0.002}$ | $0.023_{\pm 0.002}$ | $0.021_{\pm 0.000}$ |
| DenseNet121 [48] | 2.83 | $83.38_{\pm 0.175}$ | $0.738_{\pm 0.027}$ | $0.566_{\pm 0.090}$ | $0.601_{\pm 0.036}$ | $0.748_{\pm 0.003}$ | $82.07_{\pm 0.429}$ | $0.699_{\pm 0.003}$ | $0.367_{\pm 0.091}$ | $0.372_{\pm 0.119}$ | $0.362_{\pm 0.001}$ |
| GoogleNet [106] | 1.5 | $79.03_{\pm 0.129}$ | $0.703_{\pm 0.012}$ | $0.287_{\pm 0.176}$ | $0.302_{\pm 0.207}$ | $0.699_{\pm 0.018}$ | $77.88_{\pm 0.484}$ | $0.699_{\pm 0.008}$ | $0.275_{\pm 0.0076}$ | $0.279_{\pm 0.186}$ | $0.369_{\pm 0.006}$ |
| ResNet18 [43] | 1.81 | $77.44_{\pm 0.164}$ | $0.678_{\pm 0.017}$ | $0.530_{\pm 0.033}$ | $0.551_{\pm 0.023}$ | $0.677_{\pm 0.022}$ | $76.26_{\pm 0.223}$ | $0.659_{\pm 0.008}$ | $0.393_{\pm 0.105}$ | $0.287_{\pm 0.029}$ | $0.321_{\pm 0.005}$ |
| ResNet50 [43] | 4.09 | $83.23_{\pm 0.015}$ | $0.656_{\pm 0.034}$ | $0.326_{\pm 0.014}$ | $0.457_{\pm 0.153}$ | $0.682_{\pm 0.043}$ | $82.06_{\pm 0.390}$ | $0.681_{\pm 0.007}$ | $0.289_{\pm 0.078}$ | $0.207_{\pm 0.065}$ | $0.334_{\pm 0.003}$ |
| Vgg13 [99] | 11.31 | $10.00_{\pm 0.000}$ | $0.021_{\pm 0.000}$ | $0.011_{\pm 0.009}$ | $0.021_{\pm 0.000}$ | $0.010_{\pm 0.005}$ | $10.00_{\pm 0.000}$ | $0.021_{\pm 0.000}$ | $0.015_{\pm 0.005}$ | $0.006_{\pm 0.000}$ | $0.016_{\pm 0.003}$ |
| Vit_b_16 [28] | 17.56 | $48.05_{\pm 1.544}$ | $0.085_{\pm 0.008}$ | $0.148_{\pm 0.034}$ | $0.117_{\pm 0.016}$ | $0.369_{\pm 0.038}$ | $24.53_{\pm 0.039}$ | $0.049_{\pm 0.004}$ | $0.071_{\pm 0.008}$ | $0.065_{\pm 0.006}$ | $0.054_{\pm 0.010}$ |
| Vit_l_32 [28] | 61.55 | $53.27_{\pm 1.376}$ | $0.201_{\pm 0.036}$ | $0.273_{\pm 0.093}$ | $0.269_{\pm 0.028}$ | $0.709_{\pm 0.088}$ | $17.10_{\pm 0.111}$ | $0.043_{\pm 0.006}$ | $0.072_{\pm 0.010}$ | $0.071_{\pm 0.010}$ | $0.061_{\pm 0.013}$ |

Table 2.5: Experimental results (Top-1 classification accuracy and attribution IoU) on vision data with structural background noise.

| Architecture | GFLOPs | SBFP | | | | | SBRP | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | **Pred**(ACC%) | **SA**(IOU) | **DL**(IOU) | **IG**(IOU) | **FA**(IOU) | **Pred**(ACC%) | **SA**(IOU) | **DL**(IOU) | **IG**(IOU) | **FA**(IOU) |
| AlexNet [58] | 0.71 | $10.00_{\pm 0.000}$ | $0.021_{\pm 0.000}$ | $0.021_{\pm 0.000}$ | $0.017_{\pm 0.055}$ | $0.021_{\pm 0.000}$ | $10.00_{\pm 0.000}$ | $0.021_{\pm 0.000}$ | $0.021_{\pm 0.000}$ | $0.021_{\pm 0.000}$ | $0.021_{\pm 0.000}$ |
| DenseNet121 [48] | 2.83 | $85.60_{\pm 0.240}$ | $0.767_{\pm 0.002}$ | $0.645_{\pm 0.008}$ | $0.658_{\pm 0.019}$ | $0.857_{\pm 0.003}$ | $84.23_{\pm 0.174}$ | $0.681_{\pm 0.024}$ | $0.599_{\pm 0.013}$ | $0.608_{\pm 0.040}$ | $0.400_{\pm 0.013}$ |
| GoogleNet [106] | 1.5 | $81.19_{\pm 0.397}$ | $0.780_{\pm 0.003}$ | $0.570_{\pm 0.035}$ | $0.572_{\pm 0.032}$ | $0.819_{\pm 0.004}$ | $79.42_{\pm 0.355}$ | $0.731_{\pm 0.005}$ | $0.519_{\pm 0.041}$ | $0.515_{\pm 0.043}$ | $0.402_{\pm 0.003}$ |
| ResNet18 [43] | 1.81 | $80.41_{\pm 0.175}$ | $0.745_{\pm 0.002}$ | $0.610_{\pm 0.023}$ | $0.618_{\pm 0.018}$ | $0.825_{\pm 0.006}$ | $77.56_{\pm 0.392}$ | $0.637_{\pm 0.011}$ | $0.509_{\pm 0.026}$ | $0.514_{\pm 0.045}$ | $0.381_{\pm 0.008}$ |
| ResNet50 [43] | 4.09 | $87.71_{\pm 0.055}$ | $0.762_{\pm 0.002}$ | $0.643_{\pm 0.019}$ | $0.680_{\pm 0.034}$ | $0.863_{\pm 0.011}$ | $85.09_{\pm 0.086}$ | $0.675_{\pm 0.011}$ | $0.565_{\pm 0.038}$ | $0.613_{\pm 0.023}$ | $0.417_{\pm 0.001}$ |
| Vgg13 [99] | 11.31 | $10.00_{\pm 0.000}$ | $0.021_{\pm 0.000}$ | $0.019_{\pm 0.001}$ | $0.014_{\pm 0.062}$ | $0.016_{\pm 0.005}$ | $10.00_{\pm 0.000}$ | $0.021_{\pm 0.000}$ | $0.015_{\pm 0.006}$ | $0.016_{\pm 0.005}$ | $0.015_{\pm 0.004}$ |
| Vit_b_16 [28] | 17.56 | $52.50_{\pm 1.420}$ | $0.671_{\pm 0.012}$ | $0.364_{\pm 0.056}$ | $0.485_{\pm 0.064}$ | $0.752_{\pm 0.005}$ | $22.55_{\pm 0.609}$ | $0.178_{\pm 0.006}$ | $0.090_{\pm 0.026}$ | $0.085_{\pm 0.011}$ | $0.097_{\pm 0.010}$ |
| Vit_l_32 [28] | 61.55 | $53.78_{\pm 0.695}$ | $0.723_{\pm 0.061}$ | $0.273_{\pm 0.056}$ | $0.375_{\pm 0.023}$ | $0.542_{\pm 0.104}$ | $13.92_{\pm 1.754}$ | $0.101_{\pm 0.031}$ | $0.058_{\pm 0.015}$ | $0.054_{\pm 0.020}$ | $0.057_{\pm 0.015}$ |

method is the best considering all factors.

### Vision Data Experiment

For the vision task, we evaluated eight different architectures under four noise conditions, as detailed in Tables 2.4 and 2.5. All models underwent 30 epochs of training using the AdamW optimizer and a cosine annealing with warm restarts scheduler, with learning rates set to 0.001. The models were tested to identify the top-k important features, where k corresponds to the number of foreground image pixels forming a rectangle. Using these features, we determined the minimum bounding rectangle and calculated the intersection over union (IOU) score with the ground truth to assess the performance of attribution methods. Notably, AlexNet and Vgg13 failed to converge in all experiments, merely producing random guesses, likely due to the absence of skip connections. We observed that only SA consistently reported an even distribution of attributions, indicating no intrinsic inductive bias, unlike other methods. Moreover, SA generally outperformed other attribution methods, aligning with results from the symbolic functional data experiments.

**Impact of Structural vs. Random Background Noise** : Neural networks (NNs) generally perform better at filtering out structural noise compared to random noise. Similarly, all attribution methods demonstrated enhanced performance with structural noise. However, the performance improvement varied significantly among different attribution methods. SA showed only modest gains, whereas other methods improved substantially. Notably, the FA method outperformed SA on structural backgrounds (SBFP), possibly because patch-ablation-based FA, which interprets patches of pixels rather than individual pixels, is more effective at handling structural noises due to their semantic coherence.

**Impact of Predictive Features at Random vs. Fixed Positions** : Both neural networks and attribution methods show improved performance for predictive features at fixed positions, a trend that is particularly pronounced in Vision Transformers (ViTs). This could be due to two main factors: First, position encoding in ViTs may be less effective at integrating positional information into the input. Second, the pixel patches ViTs analyze often include a mix of irrelevant and predictive features. This effect is also observed in attribution methods, where performance in the SBFP condition is significantly better than in others. Specifically, ViT_l_32 outperforms ViT_b_16 in fixed position scenarios but is less effective with random positions, likely because smaller patches include fewer patch-level noises when the foreground moves. Interestingly, even though CNN-based models are theoretically invariant to translation, they too perform better in fixed position conditions.

In summary, two broad observations emerge from our analysis: (I) Among various attribution methods, SA almost always outperforms the other three methods. (II) Neural networks demonstrate superior performance when irrelevant features are structural and positioned fixedly.

### Audio Data Experiment

Similar to our vision data experiments, we also explored the effects of random versus structural noise on multi-channel time-series data, training each model with the AdamW optimizer and cosine annealing with warm restarts scheduler. The learning rate for the transformer is 0.0001 while the others are 0.001. The results are detailed in Table 2.6. We applied attribution

Table 2.6: Experimental results (classification Top-1 accuracy and FPrec) on synthetic audio data with the foreground signal at a fixed position.

| **Architecture** | **GFLOPs** | **RBFP** | | | | | **SBFP** | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | **Pred**(ACC%) | **SA**(FPrec) | **DL**(FPrec) | **IG**(FPrec) | **FA**(FPrec) | **Pred**(ACC%) | **SA**(FPrec) | **DL**(FPrec) | **IG**(FPrec) | **FA**(FPrec) |
| RNN [71] | 6.86 | $67.00_{\pm 0.135}$ | $0.653_{\pm 0.129}$ | $0.585_{\pm 0.091}$ | $0.275_{\pm 0.170}$ | $0.447_{\pm 0.009}$ | $63.16_{\pm 0.105}$ | $0.644_{\pm 0.113}$ | $0.541_{\pm 0.044}$ | $0.340_{\pm 0.092}$ | $0.482_{\pm 0.005}$ |
| LSTM [90] | 9.00 | $73.77_{\pm 2.605}$ | $0.684_{\pm 0.014}$ | $0.590_{\pm 0.019}$ | $0.260_{\pm 0.006}$ | $0.515_{\pm 0.010}$ | $76.07_{\pm 0.190}$ | $0.810_{\pm 0.012}$ | $0.772_{\pm 0.001}$ | $0.439_{\pm 0.002}$ | $0.546_{\pm 0.009}$ |
| TCN [51] | 6.73 | $80.96_{\pm 1.237}$ | $0.641_{\pm 0.005}$ | $0.297_{\pm 0.006}$ | $0.258_{\pm 0.002}$ | $0.539_{\pm 0.003}$ | $82.52_{\pm 0.185}$ | $0.721_{\pm 0.007}$ | $0.427_{\pm 0.009}$ | $0.337_{\pm 0.005}$ | $0.561_{\pm 0.023}$ |
| Transformer [108] | 29.7 | $23.25_{\pm 0.756}$ | $0.460_{\pm 0.010}$ | $0.236_{\pm 0.015}$ | $0.138_{\pm 0.004}$ | $0.335_{\pm 0.008}$ | $25.75_{\pm 0.320}$ | $0.513_{\pm 0.054}$ | $0.315_{\pm 0.007}$ | $0.193_{\pm 0.006}$ | $0.373_{\pm 0.007}$ |

methods to this data, aggregating absolute attributions across each channel, with channel importance calculated by summing attributions, $\sum_{t=1}^{T} |a_{\hat{c},t}| / \sum_{c=1}^{C} \sum_{t=1}^{T} |a_{c,t}|$.

Temporal convolutional neural networks (TCN) significantly outperformed other models, likely due to their convolution's transition-invariant properties, which effectively encode learning biases. Similar to our findings in vision data, models and attribution methods showed better performance with structural noise compared to random noise. Integrated gradients underperformed relative to other methods across all models, potentially due to two factors: (I) The use of a zero baseline might introduce bias, and (II) integrating gradients along a straight pathway could deviate from the data manifold, leading to errors.

## Feature Selection with Neural Networks and Post-hoc Attributions

Feature selection [73] aims to reduce the number of input variables for building predictive models. Traditional machine learning methods commonly employ univariate filtering, embedding, and wrapper methods. A key wrapper method is Recursive Feature Elimination (RFE), which starts with all features and iteratively removes the least important ones based on model coefficients that signify feature importance. However, RFE's reliance on model transparency limits its direct application to neural networks, which are typically opaque. To bridge this gap, we introduce an adaptation known as Recursive Feature Elimination with Neural Networks and Post-hoc Attribution (RFEwNA), detailed in Algorithm 2, enabling RFE's application in more complex, black-box models.

In this section, we extend our analysis by integrating neural networks with attribution methods into the feature selection pipeline, transforming it from an open-loop system (see Figure 2.9a) to a closed-loop system (see Figure 2.9c). We conduct experiments using all four attribution methods across both classification and regression tasks. We compare our

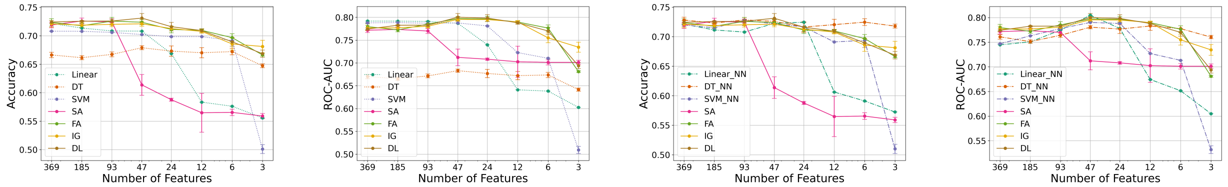
**Algorithm 2** RFEwNA

**Input**: Dataset $X$ with $m$ features, an neural networks model $F$, an post-hoc attribution explainer $g$
**Parameter**: Drop feature rate $dr\%$, target number of features $k$
**Output**: Dataset $X^*$ with selected features, trained model $F^*$

```
Start with the full set of $m$ features
while Number of features in the selected set is greater than $k$ do
    Train and evaluate $F$ on $X$
    Evaluate the importance of each feature on the validation set with $g$
    Remove the least important $dr\%$ features from the selected set
end while
```

approach against traditional Recursive Feature Elimination (RFE) methods using statistical models such as linear models, decision trees (DT), and support vector machines (SVM). We anticipate that this closed-loop configuration will yield better prediction accuracy and more effectively identify relevant features.

### RFEwNA on Classification



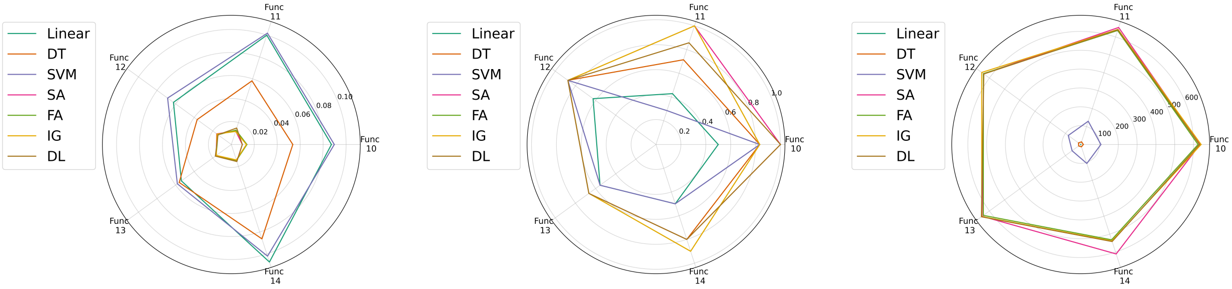
(a) Uni-modual Accuracy (b) Uni-modual IOU (c) Bi-modual Accuracy (d) Bi-modual IOU

Figure 2.14: The performance of RFE on Santander Customer Satisfaction dataset. Figure 2.14a and Figure 2.14b show the test performance of using the same classifier (dotted line) for both RFE and final classification. As a comparison, Figure 2.14c and Figure 2.14d show the test performance of using a classifier (dash-dot line) for RFE and train a neural network for the final classification.

We solved a binary classification task to predict customer satisfaction utilizing the Santander Customer Satisfaction dataset. The dataset comprises 369 features, which underwent min-max scaling preprocessing. Due to the dataset's highly imbalanced labels and the unavailability of original test labels, we performed random undersampling on the training data of the majority class, resulting in 6,016 instances. Then it is split into 80% training data and 20% validation data.

To assess the effectiveness of RFEwNA, we conducted experiments using a drop rate of 50% and targeting three features. We repeated each experiment five times with randomness. We report validation accuracy and intersection over union metrics. In these tests, the same classifier was used for both feature selection and making predictions, a method we refer to as the uni-module strategy. Our interest lies in both the peak performance as the number of features decreases and the outcomes when only a few predictive features remain. The FA, IG, and DL methods consistently outperformed traditional statistical models, indicating our method is better in prediction than classic RFE. One might question whether this superiority stems merely from the inherent predictive strength of neural networks over statistical models. To validate the efficacy of our feature selection, we implemented a bi-module strategy: selecting features using statistical models and then training a neural network on these features. This approach was then compared against the uni-module strategy. Results demonstrated that RFEwNA significantly surpassed the performance of linear and SVM models. Notably, the decision tree model not only achieved comparable outcomes but also excelled when the feature count was drastically reduced. This indicates that the selected features of our method are more predictive than the original RFE.

### RFEwNA on Regression



(a) Mean Absolute Error

(b) Functional Precision

(c) Running Time

Figure 2.15: The performance of RFE on the last 5 synthetic symbolic functional data. Figure 2.15a shows the test *MAE* as the result of feature selection and regressor training. The smaller the value, the better the performance. Figure 2.15b shows the test *FPrec* as the result of feature selection. The larger the value, the better the performance. Figure 2.15c shows the logarithmic transformed test *running time (s)* which is $log(1 + T)$.

We conducted empirical tests of our algorithm on the last five functions from our symbolic functional data, comparing them to the methodologies applied in the classification task. We assessed three key metrics: predictive performance measured by mean absolute error (MAE), feature selection efficacy via functional precision (FPrec), and computational cost using running time. Our method not only reduces MAE significantly, indicating superior accuracy but also outperforms RFE in feature selection across all tests. However, despite GPU acceleration, our methods are more time-intensive.

In summary, RFEwNA outperforms RFE in both prediction accuracy and feature selection but at a significantly higher computational cost. It is most suitable for small-scale datasets or scenarios where enhanced performance justifies the extra resource expenditure.

## Summary

**Related Works** Existing research has established benchmarks for various XAI methods. Our work uniquely assesses predictive feature selection in post-hoc attribution methods across different modalities and models in low SNR environments. In contrast, [76, 93, 16] focus on other types of explanations like counterfactual or global explanations; [66, 2, 107] target specific domains such as medical or tabular data; [83, 109, 54] explore different model types such as graph neural networks or visual language models; [45, 63, 11] examine other aspects of attribution methods like faithfulness and fairness. Besides, we are the first to integrate the attributions in the recursive feature elimination pipeline.

**Limitations and Future Works** While our analysis covered four distinct attribution methods, there remain many other significant techniques that warrant investigation. Our explorations were limited to specific models, hyperparameters, and noise levels in vision and audio data. Future work will aim to incorporate a more diverse array of attributions, models, and noise conditions.

**Conclusion** : Our paper explores the performance of neural network models and attribution methods under various configurations, providing key insights into their operational effectiveness. We discovered that saliency attribution (SA) excels in low SNR environments and that the predictive capabilities of models significantly influence the effectiveness of XAI methods. Our research also underscores the differential impact of structural versus random background noise, with neural networks demonstrating enhanced proficiency in filtering out structural noise. Additionally, we explore leveraging attribution methods to adapt the RFE approach, showing that the adapted method is better in prediction and feature selection yet computationally costly. Our study may impact algorithm design and feature selection in machine learning applications across various domains.

# Chapter 3

# Applications on Machine Learning Tasks

Explainable Artificial Intelligence (XAI) plays a critical role [30] in making AI decisions transparent and understandable, which is essential for actionable decision-making across various sectors. In healthcare, XAI aids in diagnosing diseases and predicting patient outcomes by providing clear explanations of the AI's decisions. For instance, when using task-based fMRI data to predict schizophrenia prognosis, XAI helps clinicians understand which brain patterns are indicative of treatment responses, enabling more informed and personalized treatment plans.

In finance [110], XAI enhances transparency in credit scoring, fraud detection, and investment strategies. By explaining why a loan was approved or denied, XAI ensures compliance with regulations and builds trust with customers by making financial decisions understandable. Similarly, in autonomous systems [7] like self-driving cars, XAI can clarify the reasoning behind a vehicle's decisions, improving safety and reliability by allowing developers and users to understand and trust the technology.

XAI is also crucial in cybersecurity [14], where AI models detect anomalies and potential threats. By explaining why certain activities were flagged as suspicious, XAI enables security analysts to respond more effectively, ensuring robust security measures are maintained.

In the Natural Language Processing (NLP) domain [24], XAI is used to interpret models involved in tasks like sentiment analysis, machine translation, and text summarization. For example, in sentiment analysis, XAI can elucidate why a model classified a review as positive or negative, helping businesses understand customer feedback more deeply. In machine translation, XAI can highlight which parts of the input text were crucial for generating the translation, providing insights into the model's decision-making process and helping improve translation quality.

In this chapter, we present two works on the application of XAI in the neural image domain and natural language domain. We only briefly provide information about the application domain and machine learning method but put our focus on how XAI contributes to the works. Chapter 3.1 demonstrates a use case of XAI to validate the decision of the deep convolutional neural networks. Chapter 3.2 shows how XAI combined with large language models (LLM) are able to summarize the exemplar results.

## 3.1 XAI Application for Deep Learning on Task-fMRI Data

Our work [95] explores the application of deep learning to task-fMRI (t-fMRI) data, focusing on prognosis in treatment scenarios such as childhood schizophrenia. We propose a multi-view multi-instance (MVMI) architecture that addresses the unique challenges of t-fMRI data, including varied trial types and repeated trials per subject. Our model leverages transfer learning to handle unbalanced trial types and employs a multi-layer perceptron ensemble for subject-wise predictions. The proposed method demonstrates superior performance compared to existing approaches, achieving notable improvements in predictive accuracy. We also incorporate the mixup technique to augment the data for training which improves the performance of the deep learner. We designed to frame blocking experiment, which offers the importance of frames in making a decision. The result of this validation experiment partially justified our model in decision-making.

### Introduction

Functional magnetic resonance imaging (fMRI) is widely used for analyzing brain activity, typically focusing on resting-state fMRI (rs-fMRI). However, rs-fMRI is limited for prognosis tasks. Instead, task-based fMRI (t-fMRI), which captures brain activity during specific tasks, offers more dynamic and detailed insights. This paper targets the use of t-fMRI for prognosis, particularly in predicting treatment outcomes for childhood schizophrenia.

Analyzing t-fMRI data presents unique challenges compared to rs-fMRI, as shown in Table 3.1. The AX-CPT task, for example, involves multiple event types with varying repetitions across subjects (see Figure 3.1), but only one clinical label per subject. This necessitates a multi-view (each trial type as a view) and multi-instance (each trial as an instance) approach. To tackle these challenges, we propose a deep learning architecture illustrated in Figure 3.2a. We build individual models for each trial type (AX, BX, AY, BY) and use transfer learning to handle the uneven frequency of trial types. Finally, a multi-layer perceptron (MLP) ensemble model combines these models for subject-level predictions.

In this study, we use deep learning to solve the auto-prognosis problem on t-fMRI data.

The experiment used baseline neuroimaging data from a cognitive control task involving 82 individuals with recent-onset psychosis. The classification task aimed to predict clinical improvement after one year of treatment.

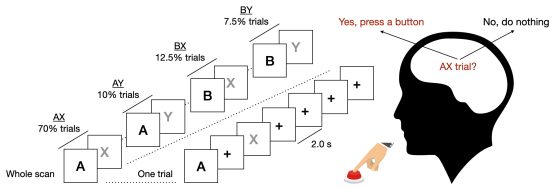
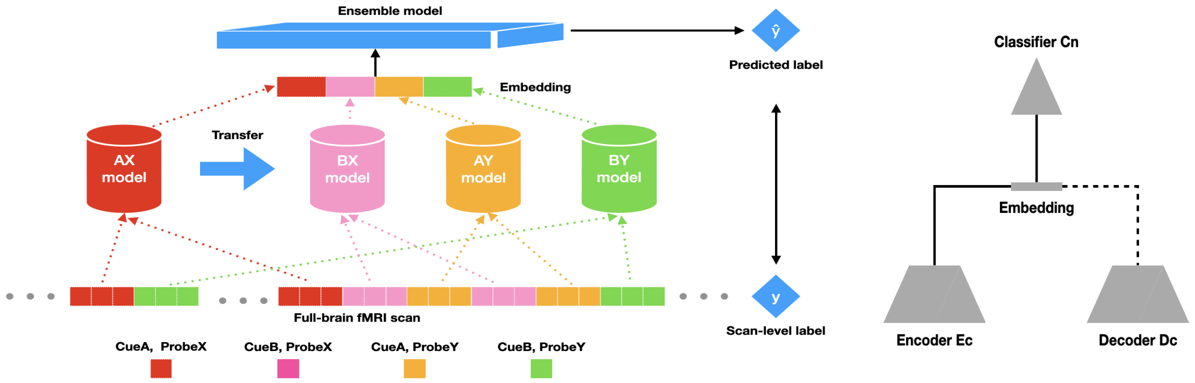


Figure 3.1: An illustration of the AX-CPT task. Each trial is started with a cue ('A' or 'B') and followed by some `Rest` frames ('+') and then a probe ('X' or 'Y'). The subject is expected to press a button only for the combination where a `CueA` is followed by a `ProbeX`. This task elicits an executive reasoning network in the brain. There are 4 types of trials (`CueA`→`ProbeX`, `CueA`→`ProbeY`, `CueB`→`ProbeX`, `CueB`→`ProbeY`). Each type of trial repeats for a varying number of times across trial types and subjects.



(a) Model architecture.

(b) Trial-type model.

Figure 3.2: An overview of our architecture. (a) The long scans are splitted into four types of trials (AX, BX, AY, BY) and a model is built for each. Then an ensemble model is learned to concatenate embeddings extracted by the trial-type models as input to train an ensemble predictor. The final predictions are supposed to match the subject-level labels. (b) The trial-type model is trained with both classification and reconstruction tasks. The embeddings of each training instance are extracted by the trial-type model for the training of the ensemble model.

As presented in Figure 3.3, analyzing t-fMRI data presents three main challenges:

Table 3.1: Differences between rs-fMRI and AX-CPT t-fMRI.

| | Events During Trial | Number of Trials |
|---|---|---|
| rs-fMRI | None | One |
| AX-CPT | CueA, ProbeX, CueB, ProbeY | Varies by subject and trial-type |

Table 3.2: The correspondences between medical terminologies (first line) and our multi-view multi-instance setting (second line).

| | | | | |
|---|---|---|---|---|
| scan/subject | trial-type | trial | frames | voxel |
| data example | view | instance | channels | pixel |

- **Multi-view Problem**: Different trial types provide varied perspectives on brain activity.
- **Multi-instance Problem**: Different trial types provide varied perspectives on brain activity.
- **Small Data Entity Problem**: A limited number of subjects necessitates effective data augmentation and transfer learning strategies.

To formalize the problem, we propose a multi-view multi-instance (MVMI) setting, which integrates both learning types by treating each trial type as a view and each view as a bag of trials for the subject. This approach mitigates the small data problem by increasing the number of training examples through trial segmentation. We outline the medical-to-MVMI terminologies in Table 3.2. Unlike classic multi-view learning [112], which assumes a fixed set of features, our data comprises variable-sized bags of instances for each view. Moreover, unlike traditional multi-instance learning [15] focused on instance-level predictions, our goal is to make bag-level predictions.

We formalize the problem of multi-view multi-instance (MVMI) learning as follows. Assume the input features is $\mathcal{X}$ and the label space is $\mathcal{Y}$, then the dataset is defined as $\mathcal{D} = \{(X_n, Y_n)|n = 1...N\}$, where $X_n \in \mathcal{X}$ and $Y_n \in \mathcal{Y}$. As this is a multi-view multi-instance setting, each training example consists of $V$ views when each view being a bag of instances rather than just one instance. For example, the n-th example $X_n$ in $\mathcal{D}$ is a set of $V$ views, $X_n = \{\mathcal{B}_{n,v}|v = 1...V\}$. Each view $\mathcal{B}_{n,v}$ is a multiset (bag) of $M_{n,v}$ instances,

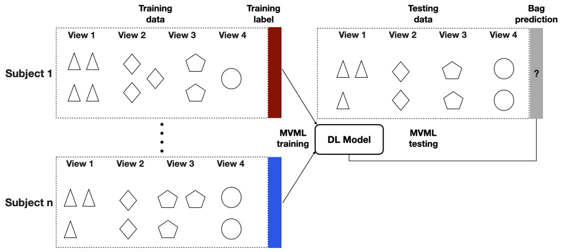


Figure 3.3: Our multi-instance multi-view setting. Unlike previous work, the number of instances per view is not constant and there is a single label per subject not per instance.

$\mathcal{B}_{n,v} = \{x_{n,v,m} | m = 1...M_{n,v}\}$, where $M_{n,v}$ (varies by $n$ and $v$) denotes the number of instances in the v-th view of the n-th example. The goal of the MVMI is to estimate a predictor $f(\cdot\,, \theta) : \mathcal{X} \rightarrow \mathcal{Y}$, with a hypothesis $\theta$.

## Approach

As discussed above, the task is framed as a multi-view multi-instance (MVMI) learning problem, where each subject's data consists of multiple views (trial types) and instances (repeated trials). The goal is to predict treatment response based on these inputs.

**Data Conversion and Preprocessing** Data was collected from t-fMRI scans of subjects performing the AX-CPT task. T-fMRI scans are segmented into trials, with each trial being a sequence of 3D brain images. These trials are categorized by type and used to train trial-specific models. Preprocessing involved normalizing voxel values and removing outliers due to motion.

**Model Architecture** We propose a deep learning architecture with separate models for each trial type, each consisting of convolutional layers followed by a classification head. A multi-layer perceptron (MLP) ensemble combines these models' outputs to make subject-level predictions. During training, we randomly sample trials from all four trial types and concatenate their features to feed into the ensemble model, significantly enlarging the training set through random combinations. The number of possible trial instance combinations grows polynomially. For validation, we apply multi-instance fusion methods to obtain a vector representation for each scan, creating a unique prediction.

Table 3.3: Transfer learning schemes and results. We present results transferring the AX source model using a variety of schemes (rows) to other models (columns). If a layer is not frozen, it is fine-tuned. The base accuracy of the AX model is 72.6% and the base accuracy for the other models is in the first row. In all cases, the target model is initialized with the source model's parameters.

| Schemes | Model AY (Target) | Model BX (Target) | Model BY (Target) |
|---|---|---|---|
| Train from scratch | 62.7% | 66.7% | 54.9% |
| Fine-tune all parameters | **68.6**% | **70.5**% | **62.7**% |
| Freeze Only Conv1 | 68.6% | 68.6% | 60.8% |
| Freeze Only Conv1 & 2 | 64.8% | 64.8% | 56.8% |
| Freeze Only Conv1 & 2 & 3 | 58.8% | 60.8% | 54.9% |

**Transfer Learning** To address data scarcity, we transfer knowledge from models trained on abundant trial types (e.g., AX) to those with fewer instances (e.g., AY, BX, BY). This involves initializing target models with parameters from source models and fine-tuning them. In the experiment, different transfer learning strategies were tested, with fine-tuning all layers proving the most effective. This approach significantly improved performance compared to training models from scratch.

Table 3.4: Various methods of multi-instance fusion and multi-view combination for prediction. The three rows denote the multi-instance (MI) fusion methods: prediction majority voting, max pooling, and mean pooling. Columns 2 to 5 denote this fusion method applied across instances for a single trial type. Columns 6 to 8 columns denote the multi-view (MV) fusion method applied across model types.

| Fusion methods | AX model | AY model | BX model | BY model | Majority voting | Weighted average | Ensemble model |
|---|---|---|---|---|---|---|---|
| Vote aggregation | 66.7% | 60.7% | 62.7% | 58.4% | 64.6% | 66.7% | 68.6% |
| Max pooling | 74.5% | 64.6% | 68.6% | 58.4% | 68.6% | 72.6% | 72.6% |
| Mean pooling | 72.6% | 68.6% | 70.5% | 62.7% | 68.6% | 72.6% | **75.6**% |

**Combining Trial Type Models** The ensemble model aggregates embeddings from trial-specific models to make final predictions. This method leverages the strengths of each trial-type model and improves overall accuracy. Various fusion methods were evaluated for

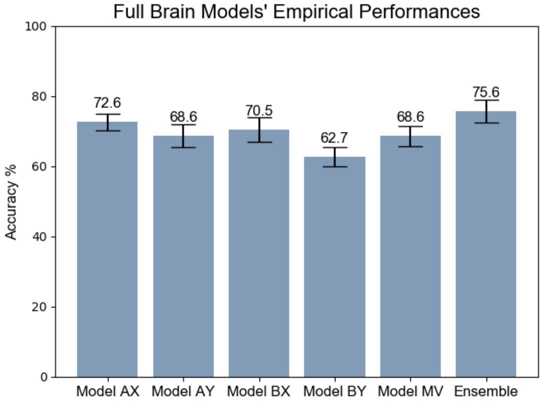


Figure 3.4: Performances of trial-type models and an ensemble model. The four trial-type models (AX, BX, AY, BY) are evaluated by mean pooling across all the instances of the same model type. The performance of 'Model MV' is evaluated on the average prediction across the four trial-type models. 'Ensemble' is the ensemble model on the embedding vectors from trial-type models.

aggregating trial-level predictions into subject-level predictions. Mean pooling emerged as the best technique, providing a balance between noise reduction and performance. The ensemble model outperformed individual trial-type models, demonstrating the benefit of combining multiple perspectives. It achieved a subject-wise accuracy of 75.6%, compared to the previous best of 72.6%.

**Mixup Augmentation** We employed the Mixup data augmentation technique to improve the performance of a deep learning classifier on fMRI data. Mixup generates new training examples by linearly combining pairs of existing instances and their labels, which helps in smoothing the decision boundary and reducing overfitting. The results showed that applying Mixup significantly enhanced the model's accuracy from 76.5% to 80.1%. The Mixup method's performance improvements were observed in both the Improver and Non-Improver classes. This suggests that Mixup is particularly effective for small fMRI datasets, providing a robust tool for enhancing machine learning performance in neuroimaging-based prognostic tasks. The study highlights Mixup's potential in overcoming the small sample size challenge inherent in many neuroimaging datasets, paving the way for more reliable and actionable clinical predictions.

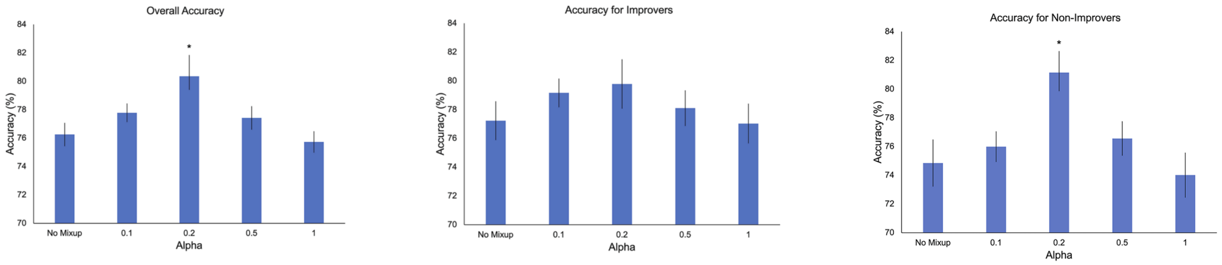


Figure 3.5: Mixup results. Error bars represent the standard error. Post-hoc tests for overall accuracy when $\alpha = 0.2$: $*p < .001$ vs no Mixup, $p = .029$ vs $\alpha = 0.1$, $*p = .014$ vs $\alpha = 0.5$, $*p < .001$ vs $\alpha = 1$. Post-hoc tests for accuracy for Non-Improvers when $\alpha = 0.2$: $*p < .002$ vs no Mixup, $p < .011$ vs $\alpha = 0.1$, $p < .022$ vs $\alpha = 0.5$, $p < .001$ vs $\alpha = 1$.

## Frame Blocking Experiment

Recall the input to each trial-type model is six frames (3 associated with the cue and 3 with the probe). To understand which frames are important for the model to make predictions, we designed the blocking studies on the input frames. At evaluation time, we block one of the frames (from cue or probe) by making all pixel values zeros in the 3D image and perform the cross-validation experiment using such input. A blocked frame which causes a large drop in accuracy is indicative of an important frame.

The results of cues and probes are shown in Table 3.5. To illustrate the notation, the "Cue" and "Probe" columns specify the three frames in each type of event. The "Block #index" means, we make all of the pixels in this indexed frame of this event to be zeros and leave the other two frames unchanged. The numbers shared by multiple columns are accuracies of the controlled trial-type models in same the row. Compared with the baseline performance, the frames that are most significant to the prediction for each trial type are frame 2 of `CueA` in AX, frame 3 of `CueB` in BX, frame 2 of `CueA` in AY, frame 3 of `CueB` in BY.

Table 3.5 reveals how the trial-type models put attention on the time series. The peak BOLD signals of the human brain seeing cues and probes are variant between 4-8 seconds after the event occurs. The frames that the models focus on thus roughly align with the peak BOLD response as expected. This experiment shows how the XAI method can validate a machine-learning model in its attention to make a decision.

Table 3.5: Frame importance by the blocking experiment. Recall the input into the deep learner for each probe and cue is three temporally adjacent frames. Here we aim to determine the most important by blocking (assign zeros to all voxels) various frames during the prediction of the performances of the trial-type models' changes. The blocked frame that most decreases performance is the most important.

| Trial-types | w/o blocking | Cue | | | Probe | | |
|---|---|---|---|---|---|---|---|
| | | Block 1 | Block 2 | Block 3 | Block 1 | Block 2 | Block 3 |
| AX | 72.6% | 66.7% | **60.8**% | 64.8% | **64.8**% | 64.8% | 66.7% |
| BX | 70.5% | 64.8% | 62.8% | **60.8**% | 66.7% | **64.8**% | 68.6% |
| AY | 68.6% | 60.8% | **56.9**% | 58.8% | **60.8**% | 62.7% | 66.7% |
| BY | 62.7% | 56.9% | 54.9% | **52.9**% | 58.8% | **56.9**% | 60.8% |

**Conclusion.** In this study, we first bring forward some intrinsic challenges of existing methods on task-fMRI data. These challenges put limits on the utilization of some existing methods such as co-activation matrix, multi-view co-training methods, and variations of recurrent neural networks. Through scrutiny, we came up with a novel multi-view multi-instance learning setting that perfectly fits the task. Then, we proposed a deep learning architecture that is able to extract task-specific features from different types of trials through trial-type models and concatenate the features to make subject-wise predictions through an ensemble model. The CNN-based trial-type models are trained on varied numbers of repeated trials. Transfer learning is used between different trial-type models, which enables the knowledge injection between them. Transferring the parameters from the most frequent trial-type model to other trial-type models by initialization and fine-tuning has a tremendous impact on the performance. Our deep architecture involving multiple trial-type models and an ensemble model is an adaptable new paradigm in task-fMRI analysis with multiple types of repeated trials.

## 3.2 Extension to Exemplars: A Natural Language Topic Modeling Example with Large Language Models (LLMs)

Exemplar-based methods in XAI focus on providing explanations for AI model decisions by referring to specific instances (exemplars) in the data. These exemplars serve as concrete examples or case studies that illustrate why the model made certain decisions. Typically, this approach relies on extracting prototypes, which are instances that are most representative of patterns learned by the model to explain normal behavior or decision-making in commonly occurring scenarios. This approach helps users better understand the behavior and reasoning of AI systems through familiar and tangible comparisons.

As an example of this approach, [26] introduced an explainable-by-design clustering method that not only identifies clusters but also selects exemplars to clarify each cluster. This method leverages exemplars, grounded in the concept of exemplarity from psychology, to tackle the complex task of explaining data clusters. However, while it effectively uses exemplars to enhance explaining clusters, it does not address two critical questions:

- **Q1**: How are the exemplars interpreted within the context of a cluster?
- **Q2**: How do the exemplars facilitate cluster-level summarization?

To address these gaps, we suggest employing large language models (LLM) [120] in the natural language processing domain as a follow-up to explore these unresolved issues. In this study, we present a topic modeling task with simultaneous clustering and exemplar extraction aiming to enhance the explainability of selected papers of the “arXiv dataset” [101], which comprises abstracts and categories of research papers in the computer science domain. To facilitate better presentation and visualization, we create a data map that shows the main areas where the papers are from. Our study creatively combines exemplar-based explanations, clustering results, topic modeling, and LLM-generated content into one unified pipeline for systematical data mining and knowledge conveying. Our work introduces the following key contributions:

- Cost-effective Data Mining: We reduce the expense of text-based summarization with LLMs by employing embedding models, clustering methods, and exemplar extraction techniques, enabling efficient data mining.
- Enhanced Summarization Diversity: Beyond LLMs, which often prioritize the most frequent content, our exemplar extraction method enhances more diverse summarization.
- Structural Summarization Framework: We develop a prompt-based framework using OpenAI tools for low-cost, automatic structural summarization, addressing LLMs' limitations in structural summarization.

These contributions significantly improve the efficiency, diversity, and structural clarity of data summarization.

## Methodology

Our methodology encompasses a series of modules designed to leverage the power of LLMs to enhance the explainability and visualization of complex data mining.

Table 3.6: Cost comparison of various OpenAI models for different applications across 1 million tokens.

| Chat Models | | | Embedding Models | |
|---|---|---|---|---|
| Model | Input | Output | Model | Usage |
| gpt-4o | \$5.00 | \$15.00 | text-embedding-3-small | \$0.02 |
| gpt-4-turbo | \$10.00 | \$30.00 | text-embedding-3-large | \$0.13 |
| gpt-3.5-turbo | \$1.50 | \$2.00 | ada v2 | \$0.10 |

**Creating Embeddings with LLM Embedding Models** . By transforming abstracts into high-dimensional vectors, we preserve the contextual information inherent in the text. Compared with chat models, embedding models are much cheaper and more efficient as demonstrated in Table 3.6. Therefore, **transforming texts into embeddings and then extracting exemplars is a cost-effective way to do data mining for large-scale data**

**than pure text-based summarization with a chat model**. We created the embeddings of text data with the "text-embedding-3-small" model from OpenAI.

**Feature Reduction and Clustering** . The embeddings of exemplars are mapped to a 2D space with Uniform Manifold Approximation and Projection (UMap [70]). The cosine distance function which is popularly used in the Retrieve Augmented Generation (RAG) system is adopted as it is particularly well-suited for embeddings generated by LLMs. The clustering method we use is Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN[69]) with Euclidean distance, which extends density-based clustering by converting it into a hierarchical clustering algorithm and then uses a technique to extract a flat clustering based on the stability of clusters.

**Topic Extraction with BerTopic Modling** . BerTopic [39] is a topic modeling technique that leverages embeddings and hierarchical clustering to generate coherent topic representations and extract meaningful topics from textual data. With the clustering results as a topic, it extracts cluster-specific keywords that are significant within a particular topic but not necessarily frequent across all topics using a class-based TF-IDF procedure. These keywords are then fed into an OpenAI representation generator ("gpt-3.5-turbo-instruct") to generate a short sentence-based topic representation.

**Exemplars Selection within Each Cluster** . An exemplar is a representative instance of the surroundings. In our context, it is the abstract of an article. We compare two exemplar extraction approaches within clusters. The greedy approach, selects the exemplars close to the cluster center, focusing on extracting exemplars that represent the central or most informative instances within each cluster. The set cover approach, as done in [26], exemplars of non-overlapping subsets in high-density regions are selected to pay more attention to diversity.

**Automatic Exemplar Qualification and cluster-level Summarization** . **Since exemplars still rely on human eye-checking to provide interpretability, how to use the selected exemplars in downstream data mining tasks remains a challenge**.

We explore using LLMs to qualify each exemplar and organize them in human interpretable visualizations. This procedure involves prompt engineering, a question-answer (QA) system, and an LLMs agent application.

In the following sections, we will delve into the technical details and experiments.

## Dataset Overview

The “arXiv dataset”, available at `https://www.kaggle.com/datasets/Cornell-University/arXiv` is a comprehensive collection of meta-data of research papers across various scientific domains, sourced from the preprint repository arXiv. This dataset encompasses metadata such as titles, abstracts, authors, categories, and full-text PDFs, offering a rich resource for conducting natural language processing (NLP) and machine learning experiments. Given its breadth and diversity, the arXiv dataset serves as an invaluable benchmark for developing and evaluating advanced models in tasks such as text summarization, classification, clustering, and topic modeling. Its extensive coverage of scientific literature makes it particularly suited for studies aiming to advance scholarly communication and information retrieval in the academic sphere.

In our study, specifically, we use the papers in computer science created in 2022 from 6 domains “Networking and Internet Architecture”, “Distributed Parallel, and Cluster Computing”, “Data Structures and Algorithms”, “Artificial Intelligence”, “Software Engineering”, and “Human-computer Interaction”. The text data we used are the titles and abstracts of papers saved as JSON documents. With the help of embedding conversion and UMap feature reduction, a data map of the 4589 selected documents is shown as Figure 3.6. We clearly see the UMap with cosine distance preserving the global structure of the data, including topological structure and distances between points captured by the semantic similarities.

## Topic Modeling and Exemplar Selection

Topic modeling is a powerful technique in academic research for uncovering hidden themes within large text corpora. In our study, we employ BERTopic to perform topic modeling on the dataset. Our methodology begins with generating embeddings using OpenAI’s models, which capture the semantic nuances of the text. These embeddings are then reduced in

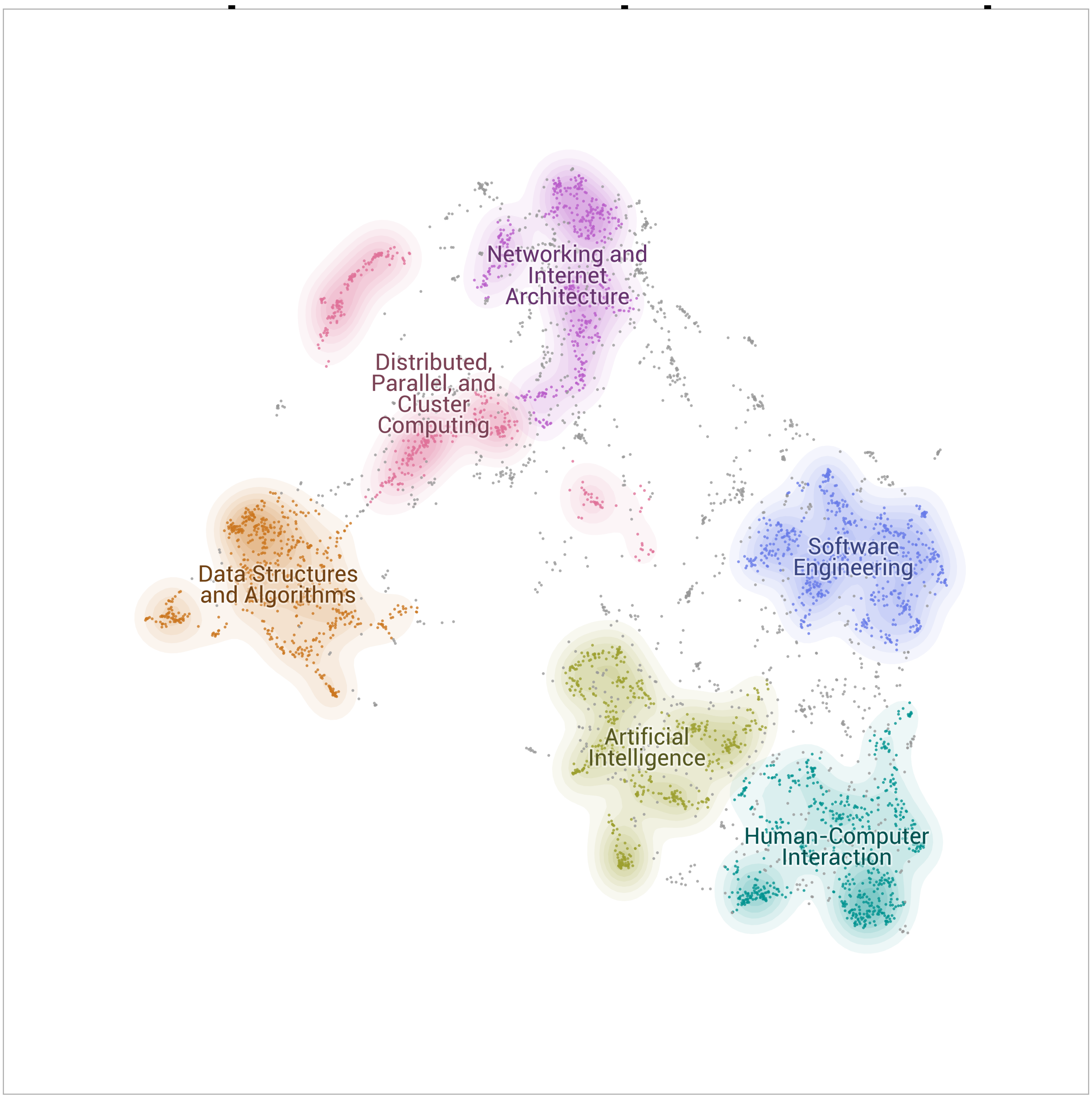


Figure 3.6: The data map of selected papers from the arXiv dataset.

dimensionality using UMAP to facilitate efficient clustering. We apply HDBSCAN with Euclidean distance[1] to identify dense clusters within the reduced space. For topic extraction, we utilize BERTopic, leveraging the C-TF-IDF algorithm to pinpoint the most representative terms for each topic from the documents. Finally, we use large language models (LLMs) to generate detailed representations of these topics, providing a comprehensive and interpretable summary of the underlying themes in the dataset. This approach ensures a robust, scalable, and interpretable topic modeling process as shown in Figure 3.7.

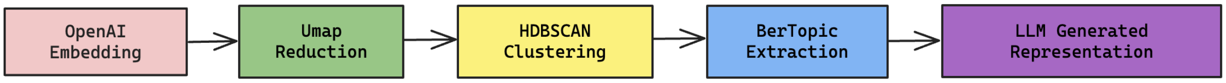


Figure 3.7: The pipeline to create topic modeling representations.

Worth noting that, with the Prompt 3.2 following the convention of BerTopic, a title-like topic name, for instance, "Edge Computing for Low-Latency Applications", is generated to summarize the documents in a cluster. For the following quantitative evaluation, we use this topic as an example for demonstration.

**The prompt to generate a topic representation**

This is a list of texts where each collection of texts describes a topic. After each collection of texts, the name of the topic they represent is mentioned as a short-highly-descriptive title.

[Few Shot Exemples]

Sample texts from this topic:

{DOCUMENTS}

Keywords: {KEYWORDS}

Topic name:

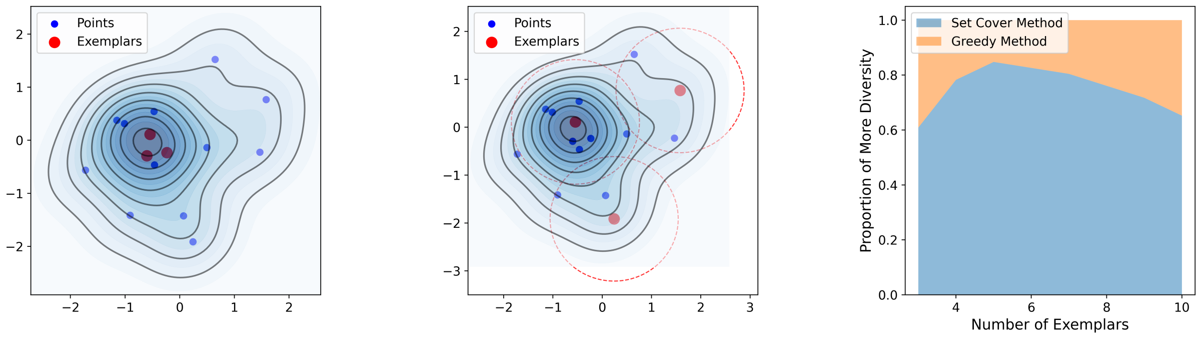
[1] Since UMap already used Cosine distance to capture the semantic similarities, Euclidean distance is suitable for downstream tasks to avoid duplication effect.

From each cluster, in other words, a topic, we select exemplars using two distinct approaches. The first approach, the greedy method (Figure 3.8a), selects exemplars that are closest to the density center of the cluster. The second approach, the set cover method (Figure 3.8b), selects exemplars based on a density definition similar to DBSCAN, which relies on epsilon distance. **This is a natural extension to [26] on the density-based clustering method.** The density is defined by the number of points within this epsilon radius. We create a density matrix with pairwise density values, indicating the number of points within the epsilon distance for each pair. The epsilon distance is defined as the maximum intra-cluster pairwise distance divided by the square of the number of exemplars desired to be selected.

The selection process begins by identifying the point with the highest density and marking all points within its density radius as covered. We then update the density matrix to remove the influence of these covered points. This process is repeated, greedily selecting the next point with the highest remaining density, until we either reach a predefined number limit or all points are covered.

Theoretically, the set cover approach is designed to select more diverse exemplars than the greedy approach by ensuring that selected points cover distinct areas of the cluster, reducing redundancy. This results in a more representative and comprehensive selection of exemplars, enhancing the quality and interpretability of the clusters.

The titles of the selected exemplars by two approaches are shown in Table 3.7. To justify whether the set cover approach actually finds more diverse exemplars, we adopt the LLMs auto validation. Basically, the group of titles is fed into ChatGPT to answer the question as Prompt 3.2. Method A and method B in the prompt represent greedy and set cover results respectively. **We do not use their actual names to avoid inductive bias as disclosing the names may sway LLM's judgment.** We collect the answers for all topics of different numbers (from 3 to 10) of exemplars selected and draw the plot as Figure 3.8c. The area means the proportion of topics favoring one approach is more inclusive than the other considered by ChatGPT. We found the set cover approach actually provides more diverse examples and the maximum difference between the two approaches is achieved when the number of exemplars is 5. This is plausible since at the beginning both methods focus on



(a) Greedy Approach

(b) Set Cover Approach

(c) ChatGPT Auto Qualification

Figure 3.8: Figure (a) is the greedy method to select exemplars close to the local density center. Figure (b) is ours' approach leveraging set cover to select representative exemplars without redundancy. Figure (c) is the expected binary answer to the question that which approach provides more diverse exemplars from ChatGPT.

the high-density cluster center. As the number of exemplars increases, the set cover approach selects exemplars from non-overlapping regions thus increasing diversity. However, when the number of exemplars grows larger and larger, the greedy method also starts including more points away from the density center. When the number of selected exemplars is the same as the number of total points in the cluster, the two methods are equally good. We only study the range of 3 to 10 since any number below 2 is trivial, and 10 more samples may exceed the input token limit.

**The prompt to judge diversity**

You are a helpful AI assistant. Given two sets of documents created by two approaches, which set of documents is more diverse? Please give me some reasons.
Documents of Approach A:
{Document Set A}
Documents of Approach B:
{Document Set B}

Table 3.7: Comparison of Exemplars for a Topic Selected by Greedy and Set Cover Approaches

**Topic: Edge Computing for Low-Latency Applications**

**Greedy Approach**

- 6G Network AI Architecture for Everyone-Centric Customized Services.
- Resource Provisioning in Edge Computing for Latency-Sensitive Applications.
- 6G Survey on Challenges, Requirements, Applications, Key Enabling Technologies, Use Cases, AI integration issues, and Security aspects.
- Deterministic Service Function Chaining over Beyond 5G Edge Fabric.
- Dynamic Service Provisioning in the Edge-cloud Continuum with Provable Guarantees.

**Set Cover Approach**

- Resource Optimization for Blockchain-based Federated Learning in Mobile Edge Computing.
- Cybertwin-enabled 6G Space-air-ground Integrated Networks: Architecture, Open Issue, and Challenges.
- EdgeKeeper: Resilient and Lightweight Coordination for Mobile Edge Computing Systems.
- PCRAFT: Capacity Planning for Dependable Stateless Services.
- VNF and CNF Placement in 5G: Recent Advances and Future Trends.

For qualitative evaluation, by human eye-checking, the bottom five papers in Table 3.7 are more diverse than the top which aligns with the choice of LLM. However, to build trust in LLM's judgment, we read the following analysis generated by ChatGPT to justify its choice.

**Comparison of Diversity Generated by ChatGPT**:

> In comparing the diversity of the two sets of documents created by Method A and Method B, we can consider several factors:
>
> 1. **Topic Variety**: Method A seems to focus more on 6G networks, AI integration,

> edge computing, and service provisioning. On the other hand, Method B covers topics such as blockchain-based federated learning, cybertwin-enabled networks, capacity planning, and VNF/container placement. Method B appears to have a broader range of topics compared to Method A.
>
> 2. **Technological Diversity**: Method A documents seem to concentrate on specific technologies and architectures related to 6G networks and edge computing. In contrast, Method B documents cover a wider array of technologies such as blockchain, cybertwin, coordination systems, and capacity planning. Method B appears to offer a more diverse technological perspective.
>
> 3. **Application Scenarios**: Method A documents focus on challenges, requirements, and applications related to 6G networks and edge computing. In contrast, Method B documents discuss topics like federated learning, space-air-ground networks, and lightweight coordination systems. Method B seems to present a more varied set of application scenarios.
>
> Based on the above considerations, it appears that the set of documents created by Method B is more diverse in terms of topics, technological coverage, and application scenarios compared to the set of documents created by Method A.

All these reasons make sense from our point of view. Therefore, given both the quantitative and qualitative clues, we argue that the LLM-based auto-qualification justified our hypothesis that set cover selects more diverse exemplars.

### Automatic Structural Cluster-level Summary

Now we answer the two questions proposed at the beginning. The final goal is instead of purely presenting the exemplars, improving the topic summarization and presenting it structurally with the exemplars to enhance interpretability. The answers to both questions are crucial to achieve this goal. The pipeline to attain the final result is illustrated in Figure 3.9. Our method employs a 4-step procedure that leverages LLMs in a memory-efficient way.

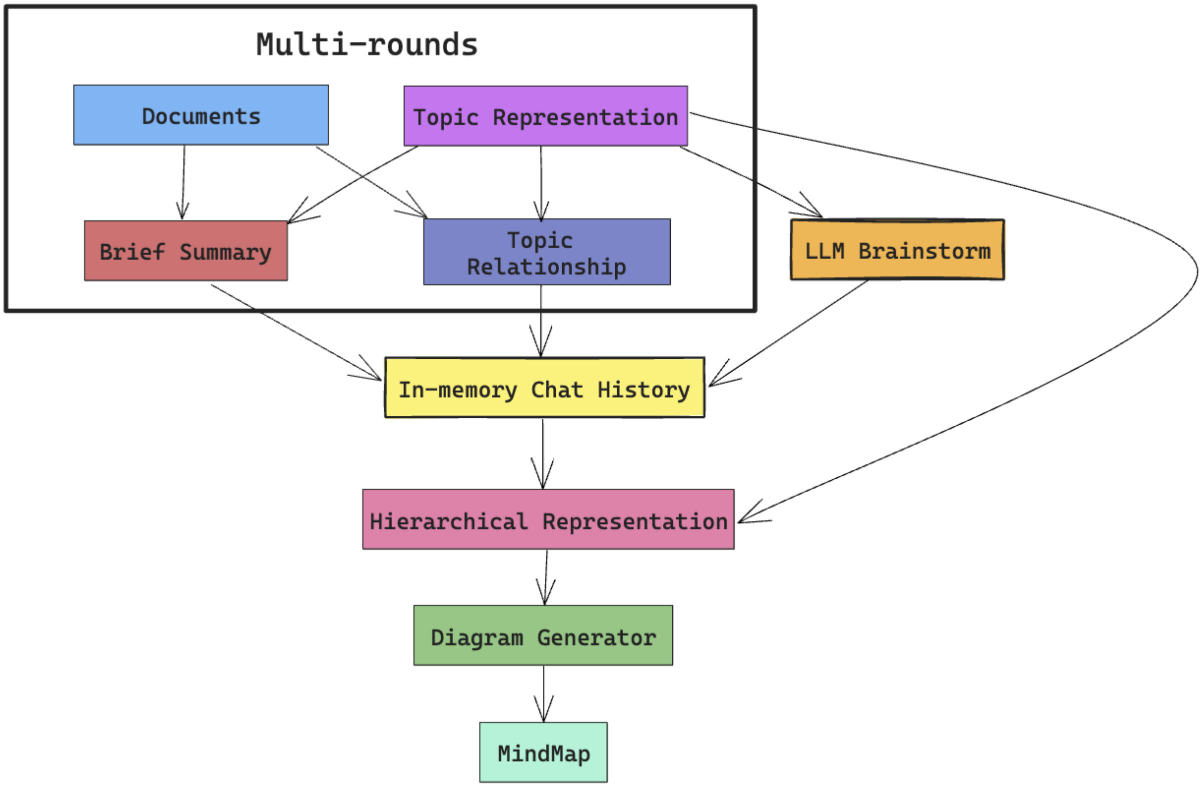


Figure 3.9: The cluster-level hierarchical representation generation pipeline with LLM.

**Step 1: Topic-related knowledge brainstorm.** LLMs possess implicit knowledge due to self-supervised pre-training. In the first step, we awake the LLM of computer science-related knowledge using Prompt 3.2. It is inspired by the chain-of-thought [111] prompting practice. Towards the topic of interest, the AI-generated child-level concepts are “Edge devices”, “Low-latency computing”, and “Edge computing architecture”; the parent-level concepts are “cloud computing”, “Internet of Things (IoT)”, and “Distributed computing”. This step makes the LLM prepared to generalize the topic to a broader view by brainstorming.

**The prompt to brainstorm given a topic**

You're a helpful AI assistant. You'll be provided with a topic and a few documents. Consider the relation between them based on your knowledge of computer science. Tell me up to 3 child-level concepts of the topic and 3 parent-level concepts of the topic. Answer each question with succinct words or a phrase.

**Step 2: Exemplar-to-topic relationship extraction.** We ask ChatGPT to answer Q1 by considering the relationship between an exemplar with its corresponding topic. This should be applied to every exemplar in a cluster, which makes it multi-round. Prompt 3.2 is used for completing this task. This comparative analysis is important for understanding the specific attributes that qualify an exemplar as a representative of the cluster. Take the second paper selected by the greedy approach as an example, the AI response is that *Ai response to content1: The document is about "Resource Provisioning in Edge Computing for Low-Latency IoT Applications." It is a child-level concept of the topic "Edge Computing for Low-Latency Applications."*

One key part is we use a **in-memory chat history** to customize strings instead of the default chat history. The succinct AI outputs are stored locally, ensuring that only essential information is retained for the next step. It's with noting that, by avoiding repeated feedings of entire chat histories (including prompts, human inputs, and AI outputs), we save on computational resources and costs. This method ensures that the token limits imposed by LLMs are respected, preventing memory overflow and reducing expenses.

**The prompt to for multi-round summarization**

Topic: {Topic}
Given the topic above and a related document, provide an analysis of how the document is related to the topic. Provide a very succinct summary of the document in the document in a phrase of keywords. If the document discusses a child-level concept of the topic, indicate where it is located in the topic's taxonomy. If the document is not directly related to a child-level concept, propose a parent-level concept of both the document and the topic.
Document:
{Document}
The document is about [summary in 10 words]. It is [relationship in 15 words].

**Step 3: Text-based hierarchical representation summary.** For this step, we generate a summary of exemplar documents in the topic with a one-shot example like Prompt 3.2. The implicit input to this prompt is the in-memory chat history we saved before. Two prominent characteristics of this prompt are: (a) it asks ChatGPT to generalize the context; and (b) it asks ChatGPT to create tree-based text representations that are good for interpretability and visualization creation in the next step. The generated text looks close to the one-shot example in the prompt as phrases with indentations.

### The prompt to generate text-based hierarchical representation of exemplars and a topic

Based on your knowledge, and the chat, propose a hierarchical representation up to 2 levels deep to summarize information of documents according to the topic. If a document is a child-level concept of the topic, add its summary phrase to the child-level of the representation. For all documents not a child-level concept of the topic, come up with a shared parent-level concept of the topic and documents, and put it at the root of hierarchical representation. The representation should be structured in a hierarchical manner, starting from the most general concept at the top and branching into more specific sub-concepts.

Here is an example:

---

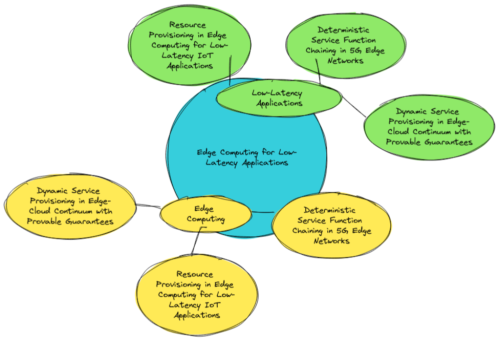
Topic:
Transformer models – neural network architectures utilizing self-attention for sequential data

Hierarchical representation up to 2 levels deep:

```
Transformers
    Encoder-Decoder Models
        Original Transformer
        Text-to-Text Transfer Transformer (T5)
        BART
    Encoder Only
        Bidirectional Encoder Representations from Transformers (BERT)
        ELECTRA
    Decoder Only
        Generative Pre-trained Transformer (GPT)
        Conditional Transformer Language Model (CTRL)
        Segment-Level Recurrence with Memor (Transformer-XL)
```

---

Do not include concepts that are not related to the document or link the document to the topic.
Hierarchical representation up to 2 levels deep:


Resource Provisioning in Edge Computing for Low-Latency IoT Applications
Deterministic Service Function Chaining in 5G Edge Networks
Low-Latency Applications
Dynamic Service Provisioning in Edge-Cloud Continuum with Provable Guarantees
Edge Computing for Low-Latency Applications
Dynamic Service Provisioning in Edge-Cloud Continuum with Provable Guarantees
Edge Computing
Deterministic Service Function Chaining in 5G Edge Networks
Resource Provisioning in Edge Computing for Low-Latency IoT Applications
Resource Optimization for Blockchain-based Federated Learning in Mobile Edge Computing
VNF and container placement optimization techniques
Edge devices
EdgeKeeper Resilient Lightweight Coordination for Mobile Edge Computing Systems
Distributed computing
Low-latency computing
Capacity planning for dependable services
Edge Computing for Low-Latency Applications
Edge computing architecture -
Cybertwin-enabled 6G SAGIN architecture and challenges

(a) Greedy Approach

(b) Set Cover Approach

Figure 3.10: The comparison of mind maps generated from the greedy approach and the set cover approach respectively.

**Stage 4: Mindmap Generation.** We create a mindmap with the hierarchical representation as input. The mindmap is a diagram for representing concepts linked to and arranged around a central concept in a tree-like structural way. We utilize the LLM-based agent,"Diagram & Data", of OpenAI to call the "Blocks And Arrows" service for the mindmap generation. The prompt we used is Prompt 3.2

**The prompt to generate a mindmap diagram**

Come up with a mindmap diagram illustration of the following hierarchical content.
{Hierarchical Text}

Finally, we show the results of cluster-level summarization from both sets of exemplars in Figure 3.10 to answer Q2. As shown, the mind map visualizations organize cluster-level knowledge in a human-interpretable tree-like diagram. The text in the diagram is concise and promising and supplements the concepts between the exemplars and the topic. Also, we can easily see the mind map of set cover selected exemplars possess more diverse concepts than the greedy one. To conclude, our automatic pipeline qualifies exemplars, and generating cluster-level summaries using LLMs offers a memory-efficient, cost-effective, and scalable solution for enhancing explainability in natural language datasets.

## Conclusion

Our approach combines the strengths of LLMs, advanced clustering, and exemplar selection algorithms to enhance the human interpretability of the natural language dataset. Our design takes into account ease of use and cost savings by employing embedding models, prompt engineering, and customized in-memory chat history. To ensure reproducibility and alleviate hallucination, we use a small value temperature of 0.01.

In addition to the contribution to data mining, our work is also an unignorable practice to inject domain knowledge into LLM. LLMs are autoregressive models adept at local sequence completion, but they face challenges in integrating global knowledge, which is typically infused during the pre-training phase. When directly applied to self-contained domains with zero-shot capabilities, injecting specific global knowledge into LLMs becomes a significant hurdle. In our approach, each cluster represents a self-contained domain, and our method efficiently imparts global knowledge about a cluster to LLMs at a low cost. This approach ensures that we capture both the fine-grained details and the broader context, facilitating deeper insights into the dataset.

# Chapter 4

# Discussion and Conclusion

**Summary.** This dissertation presents a comprehensive study of advances in Explainable Artificial Intelligence (XAI), focusing on enhancing transparency and interpretability in deep learning models. The research is structured into four main chapters, each addressing critical aspects of XAI.

- Chapter 1 introduces the necessity of XAI in modern AI systems, particularly in applications requiring high transparency and accountability. It categorizes XAI approaches into parameter space explanations and feature space explanations, setting the stage for detailed explorations in subsequent chapters.
- Chapter 2 explores explanations in the parameter space through the analysis of loss landscapes. We developed novel visualization tools, such as LossLens, to provide insights into model diagnostics. This work not only enhances understanding of model behavior but also offers practical tools for model improvement.
- Chapter 3 shifts the focus to feature space explanations, introducing methods like the Shapley Value Integrated Gradients (SIG), which enhances local explanations of feature importance. Additionally, we developed a benchmark to evaluate XAI methods in low signal-to-noise ratio environments, contributing a critical resource for the field.
- Chapter 4 applies XAI techniques to real-world machine learning tasks, demonstrating their utility in areas such as clinical prognosis using task-fMRI data and topic modeling in

natural language processing. These applications showcase the practical benefits of XAI, not only in improving model trustworthiness but also in aiding decision-making processes.

**Future Works.** As the field of artificial intelligence evolves with the advent of Large Language Models (LLMs) and more generic AI systems, XAI research must adapt to address new challenges and opportunities. One critical area is the development of XAI methods that can handle the complexity and scale of LLMs, which often involve billions of parameters and intricate decision pathways. Future research could focus on scalable XAI techniques that provide meaningful insights into these large models without overwhelming the end users with technical details.

Moreover, as AI systems become more generalized, and capable of handling diverse tasks and domains, ensuring the transparency and safety of these systems becomes paramount. XAI research can play a crucial role in developing frameworks that assess and ensure the fairness, accountability, and ethical use of AI. This includes creating standards for interpretability and developing tools that allow stakeholders to audit and understand AI decisions in a holistic manner.

Besides, as the trend of generative models such as GPT, Diffusion models, and GANs are emerging in the AI application for ordinary people, the interaction of generative AI and XAI is crucial for the next-generation AI system. On one hand, XAI can enhance the responsiveness and reliability of generative models by providing transparency, improving user trust, and facilitating better human-AI collaboration. By elucidating the decision-making processes of models like GANs and VAEs, XAI aids in debugging and optimizing model performance through better feature selection and hyperparameter tuning. It helps in identifying and mitigating biases, ensuring ethical AI practices, and making models more accessible and interpretable, thereby fostering user adoption. Additionally, XAI supports compliance with regulations by making models auditable and accountable, which is crucial in sensitive applications like healthcare and finance. By offering interactive feedback, XAI allows for iterative refinement of inputs, enabling more efficient and effective customization of generative outputs, ultimately leading to higher-quality and fairer results. On the other hand, generative models enhance XAI by creating synthetic data that can be used to probe and understand the behavior of complex AI systems. These models can generate counterfactual examples, illustrating how slight changes in input can lead to different outputs, thereby clarifying decision boundaries and feature importance. Additionally, generative models can simulate rare or edge-case scenarios that are not well-represented in the training data, providing insights into the model's robustness and potential biases. By synthesizing diverse and varied data, generative models facilitate a deeper exploration of AI models' inner workings, enabling more comprehensive and accurate explanations of their decision-making processes. My future work will focus more on the combination of XAI and generative AI to boost user experiences of AI systems.

Finally, I would like to extend my deepest gratitude to my primary instructor, Professor Ian Davidson, whose guidance and support have been invaluable throughout my research journey. His expertise and encouragement have been a cornerstone of this dissertation. I also wish to thank my committee members for their insightful discussions, collaboration, and unwavering support. Their contributions have significantly enriched this work. Thank you all for being an integral part of this academic endeavor.

# Bibliography

[1] Amina Adadi and Mohammed Berrada. "Peeking inside the black-box: a survey on explainable artificial intelligence (XAI)". In: *IEEE access* 6 (2018), pp. 52138–52160.

[2] Chirag Agarwal et al. *OpenXAI: Towards a Transparent Evaluation of Model Explanations*. 2024. arXiv: `2206.11104 [cs.LG]`.

[3] Marco Ancona, Cengiz Oztireli, and Markus Gross. "Explaining deep neural networks with a polynomial time algorithm for shapley value approximation". In: *International Conference on Machine Learning*. PMLR. 2019, pp. 272–281.

[4] Marco Ancona et al. "Towards better understanding of gradient-based attribution methods for deep neural networks". In: *arXiv preprint arXiv:1711.06104* (2017).

[5] Plamen Angelov and Eduardo Soares. "Towards explainable deep neural networks (xDNN)". In: *Neural Networks* 130 (2020), pp. 185–194.

[6] Alejandro Barredo Arrieta et al. "Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI". In: *Information fusion* 58 (2020), pp. 82–115.

[7] Shahin Atakishiyev et al. "Explainable artificial intelligence for autonomous driving: A comprehensive overview and field guide for future research directions". In: *arXiv preprint arXiv:2112.11561* (2021).

[8] R. J. Aumann and L. S. Shapley. *Values of Non-Atomic Games*. Princeton University Press, 1974. URL: `http://www.jstor.org/stable/j.ctt13x149m` (visited on 08/14/2023).

[9] Robert J Aumann and Lloyd S Shapley. *Values of non-atomic games*. Princeton University Press, 2015.

[10] Sebastian Bach et al. "On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation". In: *PloS one* 10.7 (2015), e0130140.

[11] Mohamed Karim Belaid et al. "Compare-xAI: Toward Unifying Functional Testing Methods for Post-hoc XAI Algorithms into a Multi-dimensional Benchmark". In: *World Conference on Explainable Artificial Intelligence*. Springer. 2023, pp. 88–109.

[12] Alexander Binder et al. "Layer-wise relevance propagation for neural networks with local renormalization layers". In: *International Conference on Artificial Neural Networks*. Springer. 2016, pp. 63–71.

[13] Cesar F Caiafa et al. *Machine learning methods with noisy, incomplete or small datasets*. 2021.

[14] Nicola Capuano et al. "Explainable artificial intelligence in cybersecurity: A survey". In: *IEEE Access* 10 (2022), pp. 93575–93600.

[15] Marc-André Carbonneau et al. "Multiple instance learning: A survey of problem characteristics and applications". In: *Pattern Recognition* 77 (2018), pp. 329–353.

[16] Stephen Casper et al. "Benchmarking interpretability tools for deep neural networks". In: *arXiv e-prints* (2023), arXiv–2302.

[17] Vinay Chamola et al. "A review of trustworthy and explainable artificial intelligence (XAI)". In: *IEEE Access* (2023).

[18] Hugh Chen, Scott Lundberg, and Su-In Lee. "Explaining models by propagating Shapley values of local components". In: *Explainable AI in Healthcare and Medicine*. Springer, 2021, pp. 261–270.

[19] Hugh Chen, Scott M Lundberg, and Su-In Lee. "Explaining a series of models by propagating Shapley values". In: *Nature communications* 13.1 (2022), p. 4512.

[20] Xue-wen Chen and Jong Cheol Jeong. "Enhanced recursive feature elimination". In: *Sixth International Conference on Machine Learning and Applications (ICMLA 2007)*. 2007, pp. 429–435. DOI: `10.1109/ICMLA.2007.35`.

[21] Yangkang Chen et al. "Improving the signal-to-noise ratio of seismological datasets by unsupervised machine learning". In: *Seismological Research Letters* 90.4 (2019), pp. 1552–1564.

[22] Jean-Baptiste Cordonnier, Andreas Loukas, and Martin Jaggi. "On the relationship between self-attention and convolutional layers". In: *arXiv preprint arXiv:1911.03584* (2019).

[23] Piotr Dabkowski and Yarin Gal. "Real time image saliency for black box classifiers". In: *Advances in neural information processing systems* 30 (2017).

[24] Marina Danilevsky et al. "A survey of the state of explainable AI for natural language processing". In: *arXiv preprint arXiv:2010.00711* (2020).

[25] Arun Das and Paul Rad. "Opportunities and challenges in explainable artificial intelligence (xai): A survey". In: *arXiv preprint arXiv:2006.11371* (2020).

[26] Ian Davidson et al. "An Exemplars-Based Approach for Explainable Clustering: Complexity and Efficient Approximation Algorithms". In: *Proceedings of the 2024 SIAM International Conference on Data Mining (SDM)*, pp. 46–54. DOI: 10.1137/1.9781611978032.6. eprint: https://epubs.siam.org/doi/pdf/10.1137/1.9781611978032.6. URL: https://epubs.siam.org/doi/abs/10.1137/1.9781611978032.6.

[27] Jia Deng et al. "ImageNet: A large-scale hierarchical image database". In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. 2009, pp. 248–255. DOI: 10.1109/CVPR.2009.5206848.

[28] Alexey Dosovitskiy et al. "An image is worth 16x16 words: Transformers for image recognition at scale". In: *arXiv preprint arXiv:2010.11929* (2020).

[29] Jamie Duell et al. "Batch Integrated Gradients: Explanations for Temporal Electronic Health Records". In: *International Conference on Artificial Intelligence in Medicine*. Springer. 2023, pp. 120–124.

[30] Rudresh Dwivedi et al. "Explainable AI (XAI): Core ideas, techniques, and solutions". In: *ACM Computing Surveys* 55.9 (2023), pp. 1–33.

[31] Joseph Enguehard. "Sequential Integrated Gradients: a simple but effective method for explaining language models". In: *arXiv preprint arXiv:2305.15853* (2023).

[32] Tianshu Feng et al. "Comparing Baseline Shapley and Integrated Gradients for Local Explanation: Some Additional Insights". In: *arXiv preprint arXiv:2208.06096* (2022).

[33] Christopher Frye et al. "Shapley explainability on the data manifold". In: *arXiv preprint arXiv:2006.01272* (2020).

[34] Timur Garipov et al. "Loss Surfaces, Mode Connectivity, and Fast Ensembling of DNNs". In: *Advances in neural information processing systems* (2018). DOI: `https://doi.org/10.48550/arXiv.1802.10026`.

[35] Thomas D Gauthier. "Detecting trends using Spearman's rank correlation coefficient". In: *Environmental forensics* 2.4 (2001), pp. 359–362.

[36] Amirata Ghorbani, Abubakar Abid, and James Zou. "Interpretation of neural networks is fragile". In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 33. 01. 2019, pp. 3681–3688.

[37] Amirata Ghorbani and James Zou. "Data shapley: Equitable valuation of data for machine learning". In: *International Conference on Machine Learning*. PMLR. 2019, pp. 2242–2251.

[38] Diego Granziol. "Flatness is a false friend". In: *arXiv preprint arXiv:2006.09091* (2020).

[39] Maarten Grootendorst. *BERTopic: Neural topic modeling with a class-based TF-IDF procedure*. 2022. arXiv: `2203.05794 [cs.CL]`. URL: `https://arxiv.org/abs/2203.05794`.

[40] Riccardo Guidotti et al. "A survey of methods for explaining black box models". In: *ACM computing surveys (CSUR)* 51.5 (2018), pp. 1–42.

[41] David Gunning. "Explainable artificial intelligence (xai)". In: *Defense advanced research projects agency (DARPA), nd Web* 2.2 (2017), p. 1.

[42] David Gunning and David Aha. "DARPA's explainable artificial intelligence (XAI) program". In: *AI magazine* 40.2 (2019), pp. 44–58.

[43] Kaiming He et al. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[44] Jeff Heaton. "An empirical analysis of feature engineering for predictive modeling". In: *SoutheastCon 2016*. IEEE. 2016, pp. 1–6.

[45] Anna HedstrÃ¶m et al. "Quantus: An Explainable AI Toolkit for Responsible Evaluation of Neural Network Explanations and Beyond". In: *Journal of Machine Learning Research* 24.34 (2023), pp. 1–11. URL: `http://jmlr.org/papers/v24/22-0142.html`.

[46] Tom Heskes et al. "Causal shapley values: Exploiting causal knowledge to explain individual predictions of complex models". In: *Advances in neural information processing systems* 33 (2020), pp. 4778–4789.

[47] Roberto Holgado-Cuadrado et al. "Characterization of noise in long-term ECG monitoring with machine learning based on clinical criteria". In: *Medical & Biological Engineering & Computing* 61.9 (2023), pp. 2227–2240.

[48] Gao Huang et al. "Densely connected convolutional networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 4700–4708.

[49] Neil Jethani et al. "Fastshap: Real-time shapley value estimation". In: *ICLR 2022* (2022).

[50] Anupama Jha et al. "Enhanced integrated gradients: improving interpretability of deep learning models using splicing codes as a case study". In: *Genome biology* 21.1 (2020), pp. 1–22.

[51] Nal Kalchbrenner et al. "Neural machine translation in linear time". In: *arXiv preprint arXiv:1610.10099* (2016).

[52] Andrei Kapishnikov et al. "Guided integrated gradients: An adaptive path method for removing noise". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2021, pp. 5050–5058.

[53] Nitish Shirish Keskar et al. "On large-batch training for deep learning: Generalization gap and sharp minima". In: *arXiv preprint arXiv:1609.04836* (2016). DOI: `https://doi.org/10.48550/arXiv.1609.04836`.

[54] Sunnie SY Kim et al. “HIVE: Evaluating the human interpretability of visual explanations”. In: *European Conference on Computer Vision*. Springer. 2022, pp. 280–298.

[55] Narine Kokhlikyan et al. “Captum: A unified and generic model interpretability library for pytorch”. In: *arXiv preprint arXiv:2009.07896* (2020).

[56] Simon Kornblith et al. “Similarity of neural network representations revisited”. In: *International Conference on Machine Learning*. PMLR. 2019, pp. 3519–3529. DOI: `https://doi.org/10.48550/arXiv.1905.00414`.

[57] Satyapriya Krishna et al. “The disagreement problem in explainable machine learning: A practitioner’s perspective”. In: *arXiv preprint arXiv:2202.01602* (2022).

[58] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. “Imagenet classification with deep convolutional neural networks”. In: *Advances in neural information processing systems* 25 (2012).

[59] Mateusz Krzyziński et al. “SurvSHAP (t): Time-dependent explanations of machine learning survival models”. In: *Knowledge-Based Systems* 262 (2023), p. 110234.

[60] I Elizabeth Kumar et al. “Problems with Shapley-value-based explanations as feature importance measures”. In: *International Conference on Machine Learning*. PMLR. 2020, pp. 5491–5500.

[61] Hao Li et al. “Visualizing the loss landscape of neural nets”. In: *Advances in neural information processing systems* 31 (2018).

[62] Xiao-Hui Li et al. “Quantitative evaluations on saliency methods: An experimental study”. In: *arXiv preprint arXiv:2012.15616* (2020).

[63] Xuhong Li et al. “M4: A unified xai benchmark for faithfulness evaluation of feature attribution methods across metrics, modalities and models”. In: *Advances in Neural Information Processing Systems* 36 (2023).

[64] Yuanzhi Li and Yang Yuan. “Convergence analysis of two-layer neural networks with relu activation”. In: *Advances in neural information processing systems* 30 (2017).

[65] Shuyang Liu et al. *A New Baseline Assumption of Integated Gradients Based on Shaply value.* 2024. arXiv: 2310.04821 [cs.LG]. URL: https://arxiv.org/abs/2310.04821.

[66] Yang Liu et al. "Synthetic benchmarks for scientific research in explainable machine learning". In: *arXiv preprint arXiv:2106.12543* (2021).

[67] Scott M Lundberg and Su-In Lee. "A unified approach to interpreting model predictions". In: *Advances in neural information processing systems* 30 (2017).

[68] Scott M Lundberg et al. "From local explanations to global understanding with explainable AI for trees". In: *Nature machine intelligence* 2.1 (2020), pp. 56–67.

[69] Leland McInnes, John Healy, and Steve Astels. "hdbscan: Hierarchical density based clustering". In: *Journal of Open Source Software* 2.11 (2017), p. 205. DOI: 10.21105/joss.00205. URL: https://doi.org/10.21105/joss.00205.

[70] Leland McInnes, John Healy, and James Melville. *UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction.* 2020. arXiv: 1802.03426 [stat.ML]. URL: https://arxiv.org/abs/1802.03426.

[71] Larry R Medsker and LC Jain. "Recurrent neural networks". In: *Design and Applications* 5.64-67 (2001), p. 2.

[72] Luke Merrick and Ankur Taly. "The explanation game: Explaining machine learning models using shapley values". In: *International Cross-Domain Conference for Machine Learning and Knowledge Extraction.* Springer. 2020, pp. 17–38.

[73] Jianyu Miao and Lingfeng Niu. "A survey on feature selection". In: *Procedia computer science* 91 (2016), pp. 919–926.

[74] Rory Mitchell et al. "Sampling permutations for shapley value estimation". In: *The Journal of Machine Learning Research* 23.1 (2022), pp. 2082–2127.

[75] Christoph Molnar. *Interpretable machine learning.* Lulu. com, 2020.

[76] Catarina Moreira et al. *Benchmarking Counterfactual Algorithms for XAI: From White Box to Black Box.* 2022. arXiv: 2203.02399 [cs.LG].

[77] Pramod Kaushik Mudrakarta et al. “Did the model understand the question?” In: *arXiv preprint arXiv:1805.05492* (2018).

[78] Ramin Okhrati and Aldo Lipani. “A multilinear sampling algorithm to estimate shapley values”. In: *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE. 2021, pp. 7992–7999.

[79] Chanjun Park, Minsoo Khang, and Dahyun Kim. *Model-Based Data-Centric AI: Bridging the Divide Between Academic Ideals and Industrial Pragmatism*. 2024. arXiv: `2403.01832 [cs.AI]`.

[80] Gregory Plumb, Denali Molitor, and Ameet Talwalkar. *Model Agnostic Supervised Local Explanations*. 2019. arXiv: `1807.02910 [cs.LG]`.

[81] Razieh Pourdarbani et al. “Interpretation of Hyperspectral Images Using Integrated Gradients to Detect Bruising in Lemons”. In: *Horticulturae* 9.7 (2023), p. 750.

[82] Gabrielle Ras et al. “Explainable deep learning: A field guide for the uninitiated”. In: *Journal of Artificial Intelligence Research* 73 (2022), pp. 329–396.

[83] Mandeep Rathee et al. “Bagel: A benchmark for assessing graph neural network explanations”. In: *arXiv preprint arXiv:2206.13983* (2022).

[84] General Data Protection Regulation. “General data protection regulation (GDPR)”. In: *Intersoft Consulting, Accessed in October* 24.1 (2018).

[85] Protection Regulation. “Regulation (EU) 2016/679 of the European Parliament and of the Council”. In: *Regulation (eu)* 679 (2016), p. 2016.

[86] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. *"Why Should I Trust You?": Explaining the Predictions of Any Classifier*. 2016. arXiv: `1602.04938 [cs.LG]`.

[87] Ribana Roscher et al. “Explainable machine learning for scientific insights and discoveries”. In: *Ieee Access* 8 (2020), pp. 42200–42216.

[88] Cynthia Rudin et al. “Interpretable machine learning: Fundamental principles and 10 grand challenges”. In: *Statistic Surveys* 16 (2022), pp. 1–85.

[89] Waddah Saeed and Christian Omlin. “Explainable AI (XAI): A systematic meta-survey of current challenges and future opportunities”. In: *Knowledge-Based Systems* 263 (2023), p. 110273.

[90] Haşim Sak, Andrew Senior, and Françoise Beaufays. “Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition”. In: *arXiv preprint arXiv:1402.1128* (2014).

[91] Shibani Santurkar et al. “How does batch normalization help optimization?” In: *Advances in neural information processing systems* 31 (2018).

[92] Matthias Schnaubelt, Thomas G Fischer, and Christopher Krauss. “Separating the signal from the noise–financial machine learning for twitter”. In: *Journal of Economic Dynamics and Control* 114 (2020), p. 103895.

[93] Sarah Schwettmann et al. “Find: A function description benchmark for evaluating interpretability methods”. In: *Advances in Neural Information Processing Systems* 36 (2024).

[94] Lloyd S Shapley. “Notes on the n-Person Game—II: The Value of an n-Person Game.(1951)”. In: *Lloyd S Shapley* (1951).

[95] Ge Shi, Jason Smucny, and Ian Davidson. “Deep Learning for Prognosis Using Task-fMRI: A Novel Architecture and Training Scheme”. In: *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. KDD ’22. Washington DC, USA: Association for Computing Machinery, 2022, pp. 1589–1597. ISBN: 9781450393850. DOI: 10.1145/3534678.3539362. URL: https://doi.org/10.1145/3534678.3539362.

[96] Ge Shi et al. *ChaosMining: A Benchmark to Evaluate Post-Hoc Local Attribution Methods in Low SNR Environments.* 2024. arXiv: 2406.12150 [cs.LG]. URL: https://arxiv.org/abs/2406.12150.

[97] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. “Learning important features through propagating activation differences”. In: *International conference on machine learning.* PMLR. 2017, pp. 3145–3153.

[98] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. "Deep inside convolutional networks: Visualising image classification models and saliency maps". In: *arXiv preprint arXiv:1312.6034* (2013).

[99] Karen Simonyan and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition". In: *arXiv preprint arXiv:1409.1556* (2014).

[100] Pascal Sturmfels, Scott Lundberg, and Su-In Lee. "Visualizing the impact of feature attribution baselines". In: *Distill* 5.1 (2020), e22.

[101] arXiv.org submitters. *arXiv Dataset.* 2024. DOI: `10.34740/KAGGLE/DSV/7548853`. URL: `https://www.kaggle.com/dsv/7548853`.

[102] Mukund Sundararajan and Amir Najmi. "The many Shapley values for model explanation". In: *International conference on machine learning.* PMLR. 2020, pp. 9269–9278.

[103] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. "Axiomatic attribution for deep networks". In: *International conference on machine learning.* PMLR. 2017, pp. 3319–3328.

[104] Richard S Sutton and Andrew G Barto. "Reinforcement learning: An introduction". In: 2018. Chap. 3.7.

[105] Lars Kåre Syversen. "Neural Network Robustness Against Semantic Adversarial Attacks". MA thesis. NTNU, 2021.

[106] Christian Szegedy et al. "Going deeper with convolutions". In: *Proceedings of the IEEE conference on computer vision and pattern recognition.* 2015, pp. 1–9.

[107] Satoshi Tsutsui, Winnie Pang, and Bihan Wen. "WBCAtt: A White Blood Cell Dataset Annotated with Detailed Morphological Attributes". In: *Advances in Neural Information Processing Systems* 36 (2024).

[108] Ashish Vaswani et al. "Attention is all you need". In: *Advances in neural information processing systems* 30 (2017).

[109] Lijie Wang et al. "A fine-grained interpretability evaluation benchmark for neural nlp". In: *arXiv preprint arXiv:2205.11097* (2022).

[110] 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". In: *Management Review Quarterly* 74.2 (2024), pp. 867–907.

[111] Jason Wei et al. *Chain-of-Thought Prompting Elicits Reasoning in Large Language Models*. 2023. arXiv: 2201.11903 [cs.CL]. URL: https://arxiv.org/abs/2201.11903.

[112] Xiaoqiang Yan et al. "Deep multi-view learning methods: A review". In: *Neurocomputing* 448 (2021), pp. 106–129.

[113] Ruo Yang, Binghui Wang, and Mustafa Bilgic. "IDGI: A Framework to Eliminate Explanation Noise from Integrated Gradients". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 23725–23734.

[114] Yaoqing Yang et al. "Taxonomizing local versus global structure in neural network loss landscapes". In: *Advances in Neural Information Processing Systems* (2021), pp. 18722–18733. DOI: https://doi.org/10.48550/arXiv.2107.11228.

[115] Zhewei Yao et al. "Hessian-based analysis of large batch training and robustness to adversaries". In: *Advances in Neural Information Processing Systems* 31 (2018).

[116] Zhewei Yao et al. "Large batch size training of neural networks with adversarial training and second-order information". In: *arXiv preprint arXiv:1810.01021* (2018).

[117] Zhewei Yao et al. "PyHessian: Neural networks through the lens of the hessian". In: *2020 IEEE international conference on big data (Big data)*. IEEE. 2020, pp. 581–590. DOI: https://doi.org/10.48550/arXiv.1912.07145.

[118] Matthew D Zeiler and Rob Fergus. "Visualizing and understanding convolutional networks". In: *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I 13*. Springer. 2014, pp. 818–833.

[119] Daochen Zha et al. *Data-centric Artificial Intelligence: A Survey*. 2023. arXiv: 2303.10158 [cs.LG].

[120] Wayne Xin Zhao et al. “A survey of large language models”. In: *arXiv preprint arXiv:2303.18223* (2023).

[121] Bolei Zhou et al. *Learning Deep Features for Discriminative Localization*. 2015. arXiv: `1512.04150 [cs.CV]`.