PnPXAI: A Universal XAI Framework Providing Automatic Explanations Across Diverse Modalities and Models

Seongun Kim*, Sol A Kim*, Geonhyeong Kim*, Enver Menadjiev, Chanwoo Lee, Seongwook Chung, Nari Kim†, Jaesik Choi†

Kim Jaechul Graduate School of AI, KAIST {seongun, ksola, geonhyeong.kim, enver, shiningstone23, seongwook.chung, nari.kim, jaesik.choi}@kaist.ac.kr

Abstract

Recently, post hoc explanation methods have emerged to enhance model transparency by attributing model outputs to input features. However, these methods face challenges due to their specificity to certain neural network architectures and data modalities. Existing explainable artificial intelligence (XAI) frameworks have attempted to address these challenges but suffer from several limitations. These include limited flexibility to diverse model architectures and data modalities due to hard-coded implementations, a restricted number of supported XAI methods because of the requirements for layerspecific operations of attribution methods, and sub-optimal recommendations of explanations due to the lack of evaluation and optimization phases. Consequently, these limitations impede the adoption of XAI technology in real-world applications, making it difficult for practitioners to select the optimal explanation method for their domain. To address these limitations, we introduce PnPXAI, a universal XAI framework that supports diverse data modalities and neural network models in a Plug-and-Play (PnP) manner. PnPXAI automatically detects model architectures, recommends applicable explanation methods, and optimizes hyperparameters for optimal explanations. We validate the framework's effectiveness through user surveys and showcase its versatility across various domains, including medicine and finance.

Code — https://github.com/OpenXAIProject/pnpxai API Doc. — https://openxaiproject.github.io/pnpxai/ Demo — https://openxaiproject.github.io/pnpxai/demo

Introduction

In recent years, various post hoc explanation methods have emerged as a promising approach for enhancing the model transparency. These methods aim to provide insights into the decision-making process of complex neural networks by attributing the model's output to its input features. Techniques such as gradient-based methods (Srinivas and Fleuret 2019; Smilkov et al. 2017; Adebayo et al. 2018), relevance propagation methods (Bach et al. 2015; Nam et al. 2020), and model-agnostic methods (Ribeiro, Singh, and Guestrin 2016; Lundberg and Lee 2017) have gained significant attention for their ability to generate interpretable explana-

tions. These methods are crucial for building trust in AI systems, especially in mission-critical applications like health-care (Bassi, Dertkigil, and Cavalli 2024), finance (Misheva et al. 2021), and robot manipulation (Kim and Choi 2021).

However, applying these input attribution methods is challenging due to several factors. Some methods are tailored to specific neural network architectures, like assuming a series of convolutional layers followed by linear layers, which limits their generalizability across different models (Selvaraju et al. 2017). Others require operations specific to certain layers, adding complexity to their application on complex architectures (Bach et al. 2015; Nam et al. 2020). Additionally, some methods assume extra operations based on data modalities (Sundararajan, Taly, and Yan 2017).

Recent efforts have attempted to address the limitations by providing a set of implementations of explanation methods with additional features (Kokhlikyan et al. 2020; Agarwal et al. 2022; Cugny et al. 2022), collectively known as explainable artificial intelligence (XAI) frameworks. However, these frameworks have their own constraints (see Table 1). They face challenges when applied by end users, including (1) limited flexibility for diverse and custom models due to hard-coded implementations (Agarwal et al. 2022; Hu et al. 2023), (2) a restricted number of supported XAI methods because of layer-specific requirements (Agarwal et al. 2022; Hu et al. 2023; Cugny et al. 2022), and (3) sub-optimal explanations due to the lack of integrated evaluation and optimization phases (Kokhlikyan et al. 2020; Yang et al. 2022). Finally, yet equally significant, most XAI frameworks lack user-friendly tools to help users select and optimize suitable algorithms for their own tasks, hindering the effective adoption of XAI in real-world applications.

To address these challenges, we introduce PnPXAI, a universal XAI framework that embraces diverse data modalities and neural network models for ease of use. PnPXAI is equipped with the capability to automatically detect the model architecture and recommend applicable explanation methods without necessitating an in-depth understanding from the end user. We validate the effectiveness of our framework through a user survey, demonstrating the satisfaction and usefulness of the PnPXAI framework. Furthermore, we showcase the versatility of our framework by presenting use cases across various domains and data modalities, including but not limited to medical image classification

^{*}Equal contribution

[†]Co-corresponding

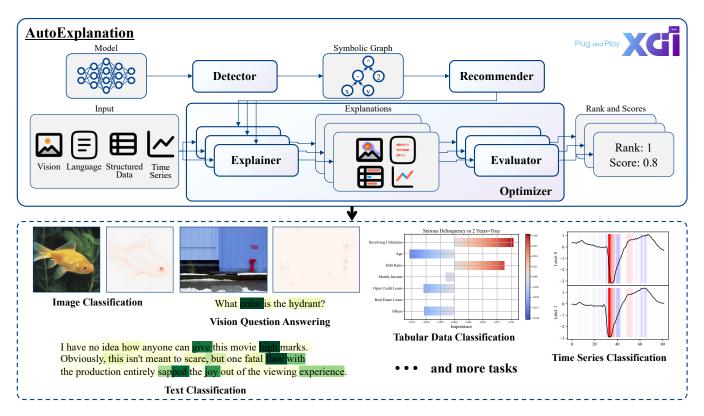


Figure 1: **Overview of the proposed framework, PnPXAI.** The detector module automatically identifies the provided neural network model architecture, which the recommender module uses to filter applicable explanation methods. The evaluator module then optimizes the explanation results through hyperparameter optimization before presenting them to end users.

| Framework | Modalities | | | Methods | | Modules | | | | | |
|-----------------------------|------------|--------------|--------------|--------------|-----------|--------------|-----------|--------------|--------------|--------------|--------------|
| | SD | V | L | TS | MA | MS | Det. | Recom. | Eval. | Opt. | AutoExp. |
| PnPXAI (ours) | < | √ | √ | √ | ✓ | √ | ✓ | √ | ✓ | √ | √ |
| Captum ¹ | ✓ | \checkmark | \checkmark | \checkmark | \ | \checkmark | | | \checkmark | | |
| OmniXAI ² | ✓ | \checkmark | \checkmark | \checkmark | \ | \checkmark | | | | | |
| AutoXAI ³ | ✓ | | | | \ | | | \checkmark | \checkmark | \checkmark | \checkmark |
| Xaitk-Saliency ⁴ | | \checkmark | | | \ | | | | | | |
| OpenXAI ⁵ | ✓ | | | | ✓ | \triangle | | | \checkmark | | |

Table 1: Comparision between PnPXAI and other XAI frameworks. MS for OpenXAI is marked as △ because it doesn't support explainers that require layer-wise operations. ¹Captum (Kokhlikyan et al. 2020). ²OmniXAI (Yang et al. 2022). ³AutoXAI (Cugny et al. 2022). ⁴Xaitk-Saliency (Hu et al. 2023). ⁵OpenXAI (Agarwal et al. 2022). Abbreviations: SD - Structured Data, V - Vision, L - Language, TS - Time Series, MA - Model-Agnostic, MS - Model-Specific, Det. - Detector, Recom. - Recommender, Eval. - Evaluator, Opt. - Optimizer, and AutoExp. - Automatic Explanation.

and fraud account detection.

PnPXAI Framework

To automatically detect the model architecture and recommend applicable explanation methods, we propose a novel XAI framework named PnPXAI, which enables end users to achieve optimal explanations in a plug-and-play manner. PnPXAI addresses the aforementioned challenges of existing XAI frameworks by modularizing it into multiple modules: detector, recommender, explainer, evaluator, and hyperparameter optimizer. As illustrated in Figure 1, the detector module automatically detects neural network architec-

tures that are used by the recommender to filter the applicable explanation methods. Explanation results from the suggested explainers are then optimized through hyperparameter optimization with the evaluator module. This modular approach ensures that PnPXAI can adapt to a wide range of models, including linear, convolution, recurrent, and transformer modules, as well as complex operations like residual connections. Additionally, PnPXAI supports various input data modalities, including vision, language, time series, and structured data. Consequently, PnPXAI provides users with accurate and reliable explanations, a feature we term *Auto-Explanation*.

| Method | Data Modalities | Architectures |
|--|-----------------|---|
| LIME, KernelSHAP | V, L, SD, TS | Linear, Convolution, Recurrent, Transformer, Decision Trees |
| Gradient, Gradient × Input | V, L, TS | Linear, Convolution, Recurrent, Transformer |
| Grad-CAM, Guided Grad-CAM | V, TS | Convolution |
| FullGrad, SmoothGrad, VarGrad | V, L, TS | Linear, Convolution, Recurrent, Transformer |
| Integrated Gradients, LRP, RAP | V, L, TS | Linear, Convolution, Recurrent, Transformer |
| AttentionRollout, TransformerAttribution | V, L | Transformer |

Table 2: **Mapping table for the recommender module.** The recommender module filters the applicable explanation methods by intersecting the identified data modalities and model architectures by the detector module. The abbreviations in the data modalities column are as follows: V for Vision, L for Language, SD for Structured Data, and TS for Time Series.

Detector

The first process undertaken by our framework is the detection of the neural network model architecture. The detector module traces and stores the symbolic graph by iterating over the model provided by end users, which will subsequently be used by the recommender module. The detected symbolic graph not only enables a detailed analysis of the model substructure for selecting applicable attribution methods but also provides layer-wise manipulability, facilitating the automation of layer-specific operations required by relevance propagation methods. If it is not necessary or possible to derive a symbolic graph from the model, such as in the case of tree-based models, the detector module provides basic information about the model architecture.

Explainer

The explainer module, which is a pool of explanation methods to be explored by the recommender module, aims to provide a large set of state-of-the-art methods applicable to a model. Whereas previous work on automatic XAI frameworks (Cugny et al. 2022) aimed to fit explanations to specific tasks and user contexts by limiting the scope of applicable methods to model-agnostic ones, such as LIME (Ribeiro, Singh, and Guestrin 2016) and SHAP (Lundberg and Lee 2017), the PnPXAI framework supports both model-specific and model-agnostic methods. These model-specific attribution methods include an equivalent number of methods as non-automatic XAI frameworks (Yang et al. 2022), such as gradient-based methods (Srinivas and Fleuret 2019; Smilkov et al. 2017; Adebayo et al. 2018), CAM-based methods (Selvaraju et al. 2017), relevance propagation methods (Bach et al. 2015; Nam et al. 2020), and attention-specific methods (Abnar and Zuidema 2020; Chefer, Gur, and Wolf 2021). This approach has advantages over previous ones in that it provides users with the opportunity to access a wider range of potentially more accurate and reliable explanations. While most preset methods are composed of promising XAI toolkits, such as Captum (Kokhlikyan et al. 2020) and Zennit (Anders et al. 2021), we also provide a flexible generic class for custom methods, allowing them to work seamlessly with the framework.

Recommender

The role of the recommender module is to determine a set of applicable explanation methods by considering input data modalities and model architecture information provided by the detector module. As summarized in Table 2, it utilizes a mapping table that consists of two sets: data modalities and neural network architectures. The recommender module supports four types of data modalities—vision, language, structured data, and time series—as well as multi-modalities such as vision and language for VQA tasks. It also supports five types of neural network modules, including linear, convolutional, recurrent, transformer, and decision trees. The module selects candidate explanation methods by intersecting these two sets. For example, if the user-provided model is ResNet50 and the task is VQA, the recommender module suggests nine candidate attribution methods, including LIME, KernelSHAP, Gradient, Gradient × Input, Smooth-Grad, VarGrad, IG, LRP, and RAP. Additionally, the use of the mapping table makes our framework extensible. If users need to implement their custom attribution methods, they only need to add their supported data modalities and model architectures to the mapping table. We believe that this extensibility facilitates advanced usage, such as benchmark studies on new explanation methods.

Evaluator

The evaluator is a module that objectively assesses the plausibility of explanations from various perspectives. As human-grounded evaluations of attribution methods can be misleading and may not accurately measure what a model attends to (Adebayo et al. 2018), we evaluate the explanations using quantitative metrics to assess their desirable properties. We are inspired by Co-12 (Nauta et al. 2023), which categorizes the evaluation properties; we employ three of them, correctness, continuity, and compactness satisfying the following conditions. Criteria requiring human subjective intervention are excluded, and frequently used and distinguishable from other properties are selected. Similar to the explainer module, our implementation provides a flexible generic class for evaluation metrics, allowing users to incorporate their custom metrics into our framework.

Hyperparameter Optimizer

In addition to the evaluator module, the optimizer module optimizes the selected explanation methods by tuning the set of hyperparameters. This helps mitigate the problem of recommending sub-optimal explanations (Yang et al. 2022), as the selection of hyperparameters significantly affects the quality of explanations (Arras, Osman, and Samek 2022; Cugny et al. 2022). The module optimizes an explanation on

Code 1: Example code snippet for running AutoExplanation in a plug-and-play manner. This code demonstrates how to initialize and execute the AutoExplanation, showcasing its ease of integration and use within the PnPXAI framework.

the user-provided dataset through a grid search. It then evaluates the explanation's properties using the selected quantitative metrics. A detailed explanation of the experimental results regarding hyperparameter optimization is provided in the following section, along with Figure 2.

AutoExplanation

By processing user-provided inputs with the aforementioned modules, PnPXAI offers *AutoExplanation*, which automatically generates optimal explanations with just a few lines of code, as illustrated in Code 1. By inputting a custom-implemented neural network model and its dataset into the AutoExplanation function provided by the PnPXAI framework and executing it, all user inputs are automatically processed through these modules, and the explanation results are recorded. Illustrative attribution heatmaps for the ImageNet classification task, IMDB movie sentiment analysis task, vision question answering task, credit scoring task, and ECG classification task, achieved by running this AutoExplanation function, are provided in Figure 1.

Use Cases

Liver Tumor Detection

To demonstrate the practical application of our framework in a medical image classification task, we validate whether the attribution heatmaps highlight the liver segments labeled as tumors in sliced computed tomography (CT) images. To this end, we prepare 2D CT images sliced along the axial axis from 3D CT images, which contain the internal structure of organs. These 3D CT images are obtained from a liver tumor segmentation dataset (Bilic et al. 2023), which provides ground truth segmentation masks of primary and secondary liver tumors labeled by seven hospitals and research institutions. We train a ResNet50 model on the sliced CT images, where the model outputs "tumor" if the input sliced image contains a segment of liver tumor, and "normal" otherwise.

By running the *AutoExplanation*, we obtain the optimized attribution heatmaps. As shown in the top center of Figure 2, PnPXAI selects the set of hyperparameters for each explanation method that achieves the best score with respect to the pre-defined objective. ABPC (Han et al. 2023) is chosen

as the default objective with the evaluation property of correctness as it measures whether the explanation is faithful to the model's decision without requiring the ground truth attribution mask.

As a result of running the *AutoExplanation*, attribution heatmaps from the default and optimized hyperparameters are illustrated at the bottom rows of Figure 2, where the first row and the second row demonstrate attribution heatmaps from the default and optimized hyperparameters, respectively. Notably, most explanation methods attribute the liver segment as identified by the ground truth segmentation masks, despite the presence of various other organs in the input image. Specifically, for relevance propagation-based methods, including variants of LRP and RAP, the attribution heatmaps highlight liver segments and minimally highlight segments of other organs. Additionally, perturbation-based attribution methods, including KernelSHAP and LIME, attribute the liver segment more accurately after hyperparameter optimization.

We also quantitatively analyze whether hyperparameter optimization improves ground truth relevance accuracy (Arras, Osman, and Samek 2022), including mass accuracy and rank accuracy. Relevance mass accuracy measures the ratio of the sum of the attribution values within the ground truth mask to the sum of all relevance values over the entire image. Similarly, relevance rank accuracy measures how much of the high-intensity attributions lie within the ground truth. We set both healthy liver segments and liver tumor segments as ground truth attributions, as the model could identify the tumor by recognizing either the segments of the tumor directly or the shape of the healthy liver segments.

The results are depicted on the top right of Figure 2. Hyperparameter optimization increases both relevance mass accuracy and relevance rank accuracy for most of the recommended explainers. Specifically, for perturbation-based attribution methods, including KernelSHAP and LIME, the accuracy improves significantly, implying that the choice of feature mask that creates superpixels is the most important hyperparameter for such methods.

Acute Kidney Injury Detection

Another use case is acute kidney injury (AKI) detection, where the PnPXAI aims to find insights into the reasons behind the model's decision-making. To test this, we preprocess a MIMIC III (Johnson et al. 2016) dataset, by extracting 79 features and setting the target to be an existence of AKI in the next 7 days after the patient's admission to the intensive care unit (ICU). Our model is the 8-layer ReLU-activated linear model with 256 hidden neurons in each layer. Following medical studies (Makris and Spanou 2016), we identify a list of biomedical markers, proven to be AKI detectors, namely estimated glomerular filtration rate (eGFR), creatinine, and blood urea nitrogen (BUN).

First, we verify the framework's applicability in explaining the model, by analyzing the ability of LRP, Integrated Gradients, LIME, and KernelSHAP to attribute to the most important features. We select the top 5 features and verify that they match the expected markers, having creatinine-related features and eGFR being the most attributed among

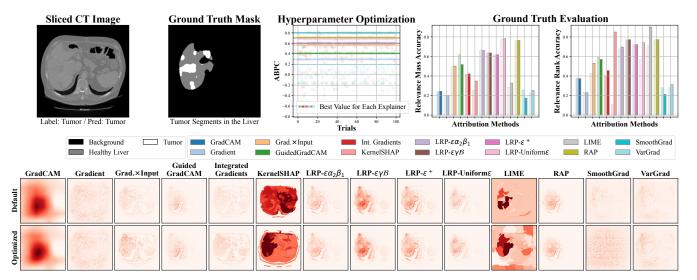


Figure 2: **Illustration of AutoExplanation for liver tumor detection.** PnPXAI recommends 14 applicable explanation methods and optimizes the selection of hyperparameters for each method on the pre-defined objective, ABPC (Han et al. 2023). This optimization improves relevance accuracy when evaluated against the ground truth segmentation mask. The attribution heatmaps at the bottom rows, where higher attribution scores are indicated by more intense red colors, demonstrate that Pn-PXAI enables to identify whether the model attributes the segments of the liver.

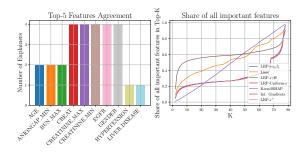


Figure 3: **Evaluation of PnPXAI in AKI detection.** The top 5 features identified by the selected explanation methods are compared against known AKI biomarkers (eGFR, creatinine, and BUN). The line graph illustrates the alignment of the ABPC metric with the share of expected features among the most attributed ones.

all explainers, and BUN highlighted by the two explainers (Figure 3). Second, we verify the evaluation ability of the framework by identifying the appearance of creatinine-related features, and eGFR in the top-k of the most attributed features, where $k \in \{2..79\}$. The line graph in Figure 3 depicts better identification of the most important features by LIME, Integrated Gradients, which aligns with the highest scores in ABPC metric. Thus, we verify the model's correctness with the framework's help by detecting the overlap of expected and the most attributed biomedical markers.

Bank Account Fraud Detection

To demonstrate the application of PnPXAI in a real-world scenario, we develop a use case for explaining fraud detection models, a crucial task in the finance sector. This use case focuses on bank account fraud detection, utilizing the Bank Account Fraud (BAF) dataset (Jesus et al. 2022), which contains information submitted during account opening and corresponding fraud status. The model is trained to detect whether an account opening request is associated with fraud based on the provided information. We employ various model structures, including ResNet, Logistic Regression, and XGBoost, to showcase the framework's versatility.

Figure 4 illustrates the user interface of the application developed for this use case. The key functionalities, including automatic recommendation of explainers based on the given model structure and calculation of explanation results and evaluation metrics, are easily implemented using the Pn-PXAI. This ease of use allows developers to create such applications without requiring extensive expertise in XAI, showcasing the framework's potential for practical, user-friendly implementations.

User Survey

A user survey, designed with the user's perspective at its core, was conducted to validate the effectiveness and convenience of PnPXAI. The user interface for our framework, provided through Gradio (Abid et al. 2019), was tested on ImageNet data for image classification tasks where XAI is the most frequently applied (Nauta et al. 2023). The survey targeted machine learning/deep learning developers/researchers with direct or indirect experience with XAI, facilitating a reliable evaluation process. A total of 31 participants were recruited from five graduate research groups and one company specializing in AI-driven solutions.

The survey consists of questions regarding user experience with XAI algorithms and questions assessing satisfaction with each PnPXAI feature. As presented in Table 3, the

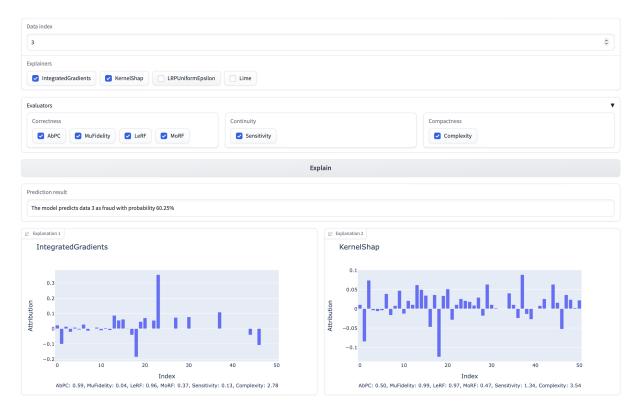


Figure 4: **Illustration of the interface of a web application for a bank account fraud detection task.** The interface allows users to choose specific explainers and evaluators for given data points. It demonstrates the importance of various features in the model's decision-making process and provides evaluation scores for each explanation algorithm.

| Features | Satisfaction ↑ | Importance ↓ |
|---------------------|-------------------|-----------------|
| Automatic detection | $ 4.06 \pm 0.15$ | 2.35 ± 0.21 |
| Recommendation | 4.13 ± 0.13 | 1.90 ± 0.16 |
| Hyperparameter opt. | 3.94 ± 0.16 | 2.32 ± 0.22 |
| Evaluation | 3.97 ± 0.16 | 2.26 ± 0.20 |

Table 3: Average user satisfaction score and importance rank for each PnPXAI feature. Satisfaction is assessed using a 5-point Likert scale, where users rated the convenience and usefulness of each feature. Importance is evaluated by ranking the four key features of PnPXAI in order of their significance in distinguishing PnPXAI from other XAI tools.

feature that automatically recommends applicable XAI algorithms received the highest satisfaction score, though others also demonstrated high satisfaction, with scores close to 4.

Among the 31 participants, 27 (87.1%) had direct experience using XAI algorithms. When asked about the challenges of using these tools, 70% of them responded that "it is difficult to trust explanations", followed by "it is difficult to understand explanations" (51.9%) and "it is difficult to find and apply XAI algorithms" (48.1%).

According to user feedback, PnPXAI enhances reliability by aligning with multiple metrics for each of the three categorized evaluation properties and provides more accurate explanations through hyperparameter optimization of XAI methods. Moreover, it increases time efficiency by automatically detecting model structures and recommending & applying applicable XAI methods.

Conclusion

In this paper, we introduced PnPXAI, a universal framework designed to address the limitations of current XAI frameworks, such as their inflexibility to diverse model architectures and data modalities, lack of easy-to-use evaluation and optimization, and barrier to utilizing various explanation algorithms at its most. By modularizing the framework into detector, recommender, explainer, and evaluator modules, PnPXAI offers a comprehensive solution that supports diverse data modalities and neural network architectures in a plug-and-play manner. We validated the usefulness of each key functionality of PnPXAI through a user survey of 31 participants. Additionally, we demonstrated its ability to provide accurate and reliable explanations through ground truth evaluation in practical use cases across various domains, including medicine and finance.

For future work, we aim to expand PnPXAI's capabilities to include explainers for generative large language models (LLMs). With its proven, easy-to-integrate design, we anticipate collaboration with the global AI research community to develop state-of-the-art explanation methods, enhancing the interpretability and trustworthiness of LLMs.

References

- Abid, A.; Abdalla, A.; Abid, A.; Khan, D.; Alfozan, A.; and Zou, J. 2019. Gradio: Hassle-Free Sharing and Testing of ML Models in the Wild. *arXiv preprint arXiv:1906.02569*.
- Abnar, S.; and Zuidema, W. 2020. Quantifying attention flow in transformers. *arXiv preprint arXiv:2005.00928*.
- Adebayo, J.; Gilmer, J.; Muelly, M.; Goodfellow, I.; Hardt, M.; and Kim, B. 2018. Sanity checks for saliency maps. *Advances in neural information processing systems*, 31.
- Agarwal, C.; Krishna, S.; Saxena, E.; Pawelczyk, M.; Johnson, N.; Puri, I.; Zitnik, M.; and Lakkaraju, H. 2022. Openxai: Towards a transparent evaluation of model explanations. *Advances in neural information processing systems*, 35: 15784–15799.
- Anders, C. J.; Neumann, D.; Samek, W.; Müller, K.-R.; and Lapuschkin, S. 2021. Software for dataset-wide XAI: from local explanations to global insights with Zennit, CoRelAy, and ViRelAy. *arXiv preprint arXiv:2106.13200*.
- Arras, L.; Osman, A.; and Samek, W. 2022. CLEVR-XAI: A benchmark dataset for the ground truth evaluation of neural network explanations. *Information Fusion*, 81: 14–40.
- Bach, S.; Binder, A.; Montavon, G.; Klauschen, F.; Müller, K.-R.; and Samek, W. 2015. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7): e0130140.
- Bassi, P. R.; Dertkigil, S. S.; and Cavalli, A. 2024. Improving deep neural network generalization and robustness to background bias via layer-wise relevance propagation optimization. *Nature Communications*, 15(1): 291.
- Bilic, P.; Christ, P.; Li, H. B.; Vorontsov, E.; Ben-Cohen, A.; Kaissis, G.; Szeskin, A.; Jacobs, C.; Mamani, G. E. H.; Chartrand, G.; et al. 2023. The liver tumor segmentation benchmark (lits). *Medical Image Analysis*, 84: 102680.
- Chefer, H.; Gur, S.; and Wolf, L. 2021. Transformer interpretability beyond attention visualization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 782–791.
- Cugny, R.; Aligon, J.; Chevalier, M.; Roman Jimenez, G.; and Teste, O. 2022. Autoxai: A framework to automatically select the most adapted xai solution. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, 315–324.
- Han, X.; Jiang, Z.; Jin, H.; Liu, Z.; Zou, N.; Wang, Q.; and Hu, X. 2023. Retiring Δ DP: New Distribution-Level Metrics for Demographic Parity. *arXiv* preprint *arXiv*:2301.13443.
- Hu, B.; Tunison, P.; RichardWebster, B.; and Hoogs, A. 2023. Xaitk-saliency: An open source explainable ai toolkit for saliency. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 15760–15766.
- Jesus, S.; Pombal, J.; Alves, D.; Cruz, A.; Saleiro, P.; Ribeiro, R.; Gama, J.; and Bizarro, P. 2022. Turning the tables: Biased, imbalanced, dynamic tabular datasets for ml evaluation. *Advances in Neural Information Processing Systems*, 35: 33563–33575.

- Johnson, A. E.; Pollard, T. J.; Shen, L.; Lehman, L.-w. H.; Feng, M.; Ghassemi, M.; Moody, B.; Szolovits, P.; Anthony Celi, L.; and Mark, R. G. 2016. MIMIC-III, a freely accessible critical care database. *Scientific data*, 3(1): 1–9.
- Kim, S.; and Choi, J. 2021. Explaining the decisions of deep policy networks for robotic manipulations. In 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2663–2669. IEEE.
- Kokhlikyan, N.; Miglani, V.; Martin, M.; Wang, E.; Alsallakh, B.; Reynolds, J.; Melnikov, A.; Kliushkina, N.; Araya, C.; Yan, S.; et al. 2020. Captum: A unified and generic model interpretability library for pytorch. *arXiv preprint arXiv:2009.07896*.
- Lundberg, S. M.; and Lee, S.-I. 2017. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30.
- Makris, K.; and Spanou, L. 2016. Acute kidney injury: definition, pathophysiology and clinical phenotypes. *The clinical biochemist reviews*, 37(2): 85.
- Misheva, B. H.; Osterrieder, J.; Hirsa, A.; Kulkarni, O.; and Lin, S. F. 2021. Explainable AI in credit risk management. *arXiv* preprint arXiv:2103.00949.
- Nam, W.-J.; Gur, S.; Choi, J.; Wolf, L.; and Lee, S.-W. 2020. Relative attributing propagation: Interpreting the comparative contributions of individual units in deep neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, 2501–2508.
- Nauta, M.; Trienes, J.; Pathak, S.; Nguyen, E.; Peters, M.; Schmitt, Y.; Schlötterer, J.; van Keulen, M.; and Seifert, C. 2023. From Anecdotal Evidence to Quantitative Evaluation Methods: A Systematic Review on Evaluating Explainable AI. *ACM Comput. Surv.*, 55(13s): 295:1–295:42.
- Ribeiro, M. T.; Singh, S.; and Guestrin, C. 2016. "Why should i trust you?" Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 1135–1144
- Selvaraju, R. R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; and Batra, D. 2017. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, 618–626.
- Smilkov, D.; Thorat, N.; Kim, B.; Viégas, F.; and Wattenberg, M. 2017. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*.
- Srinivas, S.; and Fleuret, F. 2019. Full-gradient representation for neural network visualization. *Advances in neural information processing systems*, 32.
- Sundararajan, M.; Taly, A.; and Yan, Q. 2017. Axiomatic attribution for deep networks. In *International conference on machine learning*, 3319–3328. PMLR.
- Yang, W.; Le, H.; Laud, T.; Savarese, S.; and Hoi, S. C. 2022. Omnixai: A library for explainable ai. *arXiv preprint arXiv:2206.01612*.