Explainable AI Process, Algorithms and Service Architecture for Cloud and Open-Source AI Vision Models

Zerui Wang

A Thesis
in
Department of Electrical and Computer Engineering

Presented in Partial Fulfillment of the Requirements
For the Degree of
Doctor of Philosophy (Electrical and Computer Engineering) at
Concordia University
Montréal, Québec, Canada

August 2025

© Zerui Wang, 2025

CONCORDIA UNIVERSITY
School of Graduate Studies

This is to certify that the thesis prepared

By: **Zerui Wang**

Entitled: **Explainable AI Process, Algorithms and Service Architecture for Cloud and Open-Source AI Vision Models**

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

Doctor of Philosophy (Electrical and Computer Engineering)

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

_____ Chair
Dr. Chair

_____ External Examiner
Dr. External

_____ External to Program
Dr. ExternalToProgram

_____ Examiner
Dr. Examiner1

_____ Examiner
Dr. Examiner2

_____ Thesis Supervisor
Dr. Supervisor

Approved by _____
Dr. GPD, Graduate Program Director

October 6, 2025 _____
Dr. , Dean
Gina Cody School of Engineering and Computer Science

Abstract

Explainable AI Process, Algorithms and Service Architecture for Cloud and Open-Source AI Vision Models

Zerui Wang, Ph.D.

Concordia University, 2025

This thesis develops and validates a framework for Explainable Artificial Intelligence (XAI) to produce interpretable, robust, and reproducible AI services. The research addresses key limitations in the field, including the architectural fragmentation of XAI tools, the opacity of proprietary cloud AI services, the vulnerability of explanations to adversarial attacks, and the specific challenges of spatiotemporal data analysis.

The contributions of this research are as follows: First, a methodological foundation is established through an analysis of the XAI landscape, resulting in a taxonomy of explanation techniques and a formalized XAI process. This work introduces quantitative metrics to evaluate the consistency of explanations across different methods and the stability of explanations from a single method, providing a data-driven assessment protocol.

Second, a modular, OpenAPI-compliant microservice architecture is designed to implement these principles. This architecture includes: (1) XAIport, a service for integrating XAI into the Machine Learning Operations (MLOps) lifecycle to support early and reproducible explainability. (2) XAIpipeline, a service that automates and orchestrates multi-step XAI workflows for both cloud and open-source models.

Third, to explain proprietary closed box models, a cloud-agnostic service is developed. It interacts with commercial cloud AI services via a unified API to deliver model-agnostic

feature explanations. This enables systematic auditing by evaluating attributes such as explanation consistency, attribution provenance, and service quality.

For open-source models, an assessment framework is created to systematically evaluate five quality attributes: computational cost, predictive performance, adversarial robustness, explanation deviation, and explanation resilience. This analysis demonstrates the vulnerability of XAI explanations themselves to adversarial perturbations when applied to a range of computer vision and tabular models.

For video analysis, this thesis introduces STAA (Spatio-Temporal Attention Attribution), an XAI method for video transformer models. It functions by directly extracting and aggregating attention weights from internal model layers in a single forward pass to generate joint spatio-temporal explanations. It is demonstrated to outperform existing methods in both faithfulness and monotonicity while reducing explanation latency. Additionally, this work develops a novel adversarial attack framework that explicitly targets the spatio-temporal attention mechanisms unique to video transformers. Unlike traditional frame-wise perturbations that treat videos as independent image sequences, our method jointly perturbs both spatial and temporal features by exploiting the self-attention vulnerabilities in transformer architectures. The attack achieves a high Attack Success Rate (ASR) on the Kinetics-400 dataset, significantly outperforming existing approaches, including frame-wise attacks, sparse attacks, and the V-BAD method. We quantify the attack's impact using spatial similarity metrics and temporal coherence degradation measures, revealing systematic disruption of attention patterns across both dimensions.

In summary, the contributions form an integrated framework of methodologies and deployable tools for building, explaining, and validating AI systems, with particular emphasis on understanding and securing video transformer models against adversarial threats.

Acknowledgments

I would like to express my sincere gratitude to my supervisor, Dr. Yan Liu, for her meticulous guidance, unwavering support, and valuable feedback throughout my doctoral journey. Her weekly discussions and consistent mentorship have been instrumental in shaping this research and my development as a scholar. I am deeply grateful to Dr. Wahab Hamou-Lhadj for his insightful guidance and collaborative efforts, which led to our joint publication in ICSE. The opportunity to present our work together at the conference was a significant milestone in my academic career. My appreciation goes to Dr. Essam Mansour, who served on my PhD committee and provided invaluable suggestions and evaluations throughout the entire doctoral process. His constructive feedback has substantially improved the quality of this research. I would also like to thank Dr. Paula Lago for her detailed reviews and thoughtful comments during my PhD proposal and seminars. Her thorough assessment has helped refine the direction and methodology of this work. Special thanks to Dr. Hassan Rivaz for his advice in the computer vision domain, particularly for introducing augmentation methods that enhanced the model robustness of my research. Finally, I extend my deepest gratitude to my wife for her unwavering support and sacrifices that made this doctoral journey possible, and I look forward with joy to welcoming our new baby into our lives as this chapter of my academic journey concludes.

Contents

List of Figures **xi**

List of Tables **xviii**

1 Introduction **1**

1.1 Current State and Fundamental Challenges 2

1.2 Research Objectives 3

1.3 Research Contributions 4

1.4 Thesis Structure 6

1.5 Mapping of Publications and Thesis Chapters 7

2 Background and Literature Review **11**

2.1 The Imperative for Explainable AI 11

2.1.1 Regulatory and Compliance Requirements 11

2.1.2 Technical Imperatives for Model Development 12

2.1.3 Ethical Considerations in AI Deployment 13

2.1.4 Practical User Needs and Trust Building 14

2.2 A Taxonomy of XAI Methods 15

2.2.1 Inherently Interpretable Models 15

2.2.2 Post-Hoc Explanation Methods 16

2.3 Challenges in Cloud AI Service Explainability 18

2.3.1 Closed Source AI Services 18
2.3.2 Frameworks and Packages for Implementing XAI 21
2.3.3 Data Provenance for XAI 23
2.3.4 Current Solutions and Limitations 23
2.4 XAI Evaluation Metrics and Adversarial Robustness 25
2.4.1 Evaluation of XAI Methods 25
2.4.2 Taxonomy of Adversarial Attacks 26
2.4.3 The Intersection of Robustness and Explainability 28
2.5 Challenges in Video Model Explainability 29

3 A Unified Service Framework for Explainable AI 31

3.1 An Open API Architecture to Discover the Trustworthy Explanation of Cloud AI Services 32
3.1.1 XAI Service Framework Foundation and Implementation 35
3.1.2 Framework Validation Through Scenario Studies 46
3.1.3 Evaluation of XAI Service Performance 58
3.1.4 Summary 61
3.2 XAIport - A Service Framework for the Early Adoption of XAI in AI Model Development 62
3.2.1 XAI Methods and Evaluation in the Practice 64
3.2.2 The Context of Early Adoption of XAI Services 66
3.2.3 A Pilot Evaluation 70
3.2.4 Summary 72

4 Automated Orchestration for Evaluating AI Models and Explanations 74

4.1 A Micro Process for AI Model Explanation 75
4.1.1 Motivation for the XAI process 75

4.1.2 An Analysis of the Current XAI Practice and Criterion 77
4.1.3 Process Development for Feature Contribution Explanation 79
4.1.4 Metric Definition for XAI Method 82
4.1.5 A Case Study for Image Classifier 85
4.1.6 Summary 88
4.2 XAI Assessment for Open Repository Models Under Adversarial Attacks . 89
4.2.1 Related XAI Tools Libraries and Image Adversarial Attacks 90
4.2.2 Overview of AI System's Quality Attributes 95
4.2.3 Pipelines of XAI Centric Assessment of Quality Attributes 99
4.2.4 Comparative Evaluation and Service Integration of Existing XAI Frameworks 103
4.2.5 Assessment Scenarios and Quality Attribute Analysis 105
4.2.6 Summary 116
4.3 Automated Orchestration of Explainable AI Services for Cloud AI and Open-source Models 117
4.3.1 Limitations in Current XAI Cloud Service and Integration Workflow 120
4.3.2 Configuration Driven XAI Service 121
4.3.3 Service Integration and Application Scenarios 126
4.3.4 Summary 131

5 XAI Methods and Adversarial Attacks for Spatio-Temporal Video Model 132
5.1 Spatio-Temporal Attention Attribution for Video Transformer Models . . . 134
5.1.1 The Challenges for Video XAI 137
5.1.2 Dataset Specifications and Characteristics 141
5.1.3 Methodologies for explaining Video Transformer 146
5.1.4 Comparison with Recent Video XAI Methods 160
5.1.5 Real-Time Video XAI Cloud Architecture 170

5.1.6 Summary 175
5.2 Adversarial Attacks on Video Transformer Models Through XAI-guided Perturbation 176
5.2.1 Recent Studies: Adversarial Attacks on Video Models 179
5.2.2 Spatiotemporal Adversarial Attack on Video Transformers 186
5.2.3 Spatiotemporal XAI Analysis on Video Transformers 190
5.2.4 Empirical Study on Effectiveness of Existing Adversarial Attacks . 197
5.2.5 Implications for Model Robustness Enhancement 209
5.2.6 Summary 211

6 Conclusion **213**
6.1 Summary of Contributions 213
6.2 Limitations and Considerations 215
6.2.1 Computational and Scalability Constraints 216
6.2.2 Explanation Quality and Human Factors 216
6.2.3 Black-box Model Limitations 217
6.2.4 Adversarial Robustness Scope 218
6.3 Future Work 218
6.3.1 Advancing Toward Causal and Multimodal Explanations 219
6.3.2 Extension to Video Language Models 220
6.3.3 Scalable Deployment and Real-time Adaptation 221
6.3.4 Regulatory Alignment and Standardization 222
6.4 Broader Impact and Concluding Remarks 223

A Additional XAI Case Study for Tabular and NLP Model **225**
A.1 Case Study for Tablular model: Academic Paper Ranker 225
A.2 Case Study for NLP Model: Code Vulnerability Classifier 228

Bibliography **232**

List of Figures

1 Taxonomy of explainable AI methods. The colored path indicates the methods deployed and analyzed in the case studies presented in this thesis. . . . 15

2 Grad-CAM Visual Explanations on ImageNet Dataset Samples 36

3 Workflow for Integrating Cloud AI Services with State-of-the-Art Deep Learning Models to Approximate Feature Contribution Values Using Post-Hoc XAI Methods 38

4 Microservice-Based Reference Architecture for XAI Service 40

5 Directed Graph Model of XAI Provenance Data Capturing Microservices, Tasks, and Pipeline Relationships 44

6 Comparison of Prediction Change Values for ResNet and DenseNet Approximation Models in CAM-based XAI Methods 48

7 Distribution of Prediction Change Percentages Across Cloud AI Services with Multiple CAM-based XAI Methods 49

8 Evaluation of XAI Consistency Across Cloud AI Services Using CAM-based XAI Methods: Lower Mean Values Signify Better Consistency Metric 50

9 Provenance Graph by XAI Service. Operations using ResNet and Azure Cognitive Service with Grad-CAM, and explanation evaluation 56

10 Kernel Density Estimation of prediction confidence changes for 'Our XAI Service' vs. 'OmniXAI'. 60

11 An Illustrating Scenario of Adopting XAI to Multiple Computer Vision Cloud AI Model Development. 68
12 XAI Operations and Framework Deployment Time 73
13 A feature contribution explanation micro process: main activities 80
14 Original image, saliency map generated by the Grad-CAM method, and the masked image. The ground truth label is "collie". The prediction score is 7.5674 on the original image and 5.4336 on the masked one, resulting in a prediction change of 28.17%. 86
15 Explanation stability and consistency. Left: Stability across methods; mean value indicates robustness. Right: Consistency of Grad-CAMEW compared to other methods. Green lines denote the mean values. 87
16 Five CAM-based Visual Explanations from Vision Transformer Model with One Image Example. 91
17 Top 10 out of 83 SHAP Feature Importance Explanations from FT Transformer on RT-IoT Cybersecurity Threats Dataset. 92
18 Taxonomy of Adversarial Attacks 94
19 Assessment Pipelines for Open-source AI Model Quality Attributes. 98
20 Cloud-based XAI Service Architecture. 101
21 Sample JSON Template for XAI Methods Configuration 102
22 Explanation Deviation Analysis for XAI Service (a) Versus OmniXAI (b) Using GradCAM on ResNet 104
23 Vision Models F1 Scores: Original Dataset and Adversarial Dataset Averages. 108
24 Comparison of K-S Statistics to Assess Model Robustness under Three Levels and Types of Perturbations (0: Identical, 1: Highly Divergent). . . . 109

25 Heatmaps Illustrating Median Prediction Change Percentage for Original and Adversarial Perturbed Images. Lower Values Indicate Better Explanation Deviation. 111
26 Comprehensive Overview of Multiple Quality Attributes Assessment for the Selected Models. 115
27 Sample JSON template for XAI pipeline configuration 123
28 XAIpipeline service components. (1) data processing service, including augmentation and perturbation; (2) model configuration, whether cloud-based or local model is integrated into the server; (3) XAI method and parameter setting, external tools can be configured; (4) specification of evaluation metrics; and (5) the pipeline execution. 124
29 Feature importance results from Google Vertex Explainable AI. This bar chart ranks features by their influence on model predictions. Higher values indicate features with a greater impact on the model's decision-making process. 126
30 Feature importance results from XAIpipeline's Sampled Shapley method. The violin plots illustrate the distribution and impact magnitude of the top seven selected features across two hundred samples, with median values marked. Positive values indicate features that increase prediction probability, while negative values decrease it. 127
31 Temporal feature importance across video frames for three different videos. Each subplot displays how the model's attention varies throughout the temporal dimension, with the x-axis representing frame indices and the y-axis indicating importance scores. 129
32 Measured average time consumption from XAIpipeline's service components for each image data sample. 130

33 SHAP-based temporal feature attribution method. The process begins with video segmentation into N equal temporal chunks (left), followed by analyzing the contribution of each temporal segment through model predictions with different segment combinations. The method generates temporal importance scores (right) that highlight which time periods in the video are most critical for the model's decision, enabling identification of key moments in actions while ignoring less relevant time segments. 148

34 LIME-based spatial feature attribution method: The process extracts K key frames at regular intervals from the video (left-top), then creates perturbed versions of each frame by applying random masks to different spatial regions. The model evaluates these perturbed samples to identify which spatial regions most significantly impact the prediction. The resulting spatial importance maps (left-bottom) highlight action-relevant objects and regions within each frame. 151

35 Visualized overview of the Spatio-temporal Attention Attribution (STAA) method for transformer-based video models 152

36 Example of attention visualization: heatmap overlaid on a video frame, where blue colors indicate regions of higher importance, while red indicates lower importance. 159

37 Comparison between the Transformer Interpretability (Chefer et al.'s method) and STAA (Our method). Note: Bluer colors indicate regions of higher importance. For the Deletion/Insertion Curve, the x-axis label is Proportion of Pixels Removed/Added, and the y-axis label is the Prediction Probability Scores 166

38 STAA explanations for five representative samples from the Something-Something V2 dataset. 168

39 Cloud-based architecture for real-time video XAI 171
40 Cumulative Distribution Function (CDF) of the STAA processing latency . 174
41 Overview of our spatiotemporal attack pipeline. The process maintains temporal consistency through frame processing and reconstruction while leveraging TimeSformer's attention mechanisms for effective perturbation generation. 187
42 Screenshot of the XAI-generated saliency map example. 191
43 Distribution of SSIM values comparing clean and adversarial attention maps under adversarial attack. The histogram shows the density of SSIM scores across all 20,000 analyzed videos. Lower SSIM values represent greater attention map distortion, demonstrating the impact of adversarial perturbations on spatial attention patterns. 192
44 Temporal analysis of attention patterns comparing clean and adversarially perturbed videos. The blue line represents the original video's attention progression. The orange dashed line demonstrates the adversarial video's attention pattern, revealing significant disruption in temporal coherence. The Pearson correlation coefficient quantifies the temporal attention deviation. 194

45 Comparative analysis of spatiotemporal attention patterns through 3D visualization. From top to bottom: (1) Action label from Kinetics-400 dataset, (2) Sample frames from original video showing the action sequence, (3) 3D visualization of clean video attention patterns, (4) Sample frames from adversarially perturbed video, and (5) 3D visualization of attention patterns under adversarial attack. **3D Visualization Axes**: X-axis represents spatial width pixel positions, Y-axis represents spatial height pixel positions, and Z-axis represents accumulated attention intensity. **Color Encoding**: Yellow indicates high temporal attention accumulation (consistent focus across frames), blue indicates low accumulation (sporadic attention), with intermediate colors representing varying degrees of temporal consistency. The comparison reveals how adversarial attacks (row 5) disrupt the temporal consistency of spatial attention, creating scattered attention accumulation patterns compared to the coherent, concentrated attention peaks in clean videos (row 3). 195

46 Examples of common corruptions applied to video frames. From left to right: (a) Original frame, (b) Gaussian Noise, (c) Shot Noise, (d) Motion Blur, and (e) Zoom Blur. The corruptions are applied consistently across all frames in each video sequence to simulate real-world distortions. 199

47 Flow chart of our sparse attack implementation. The attack pipeline consists of two parallel branches: (1) A gradient-based perturbation branch that applies FGSM to selected keyframes. (2) An optical flow analysis branch that identifies frames with significant motion. The outputs from both branches are combined to generate the final sparse adversarial attack that maintains temporal consistency while reducing computational overhead compared to frame-wise attacks. 202

48 The V-BAD attack [91] pipeline demonstrates a systematic approach to compromising R3D models. The method employs partition-based gradient estimation, followed by refined perturbation application across video frames. 203

49 Confidence score distribution analysis using violin plots across attack methods. 207

50 Explanation stability and consistency. Left: Stability of each method; higher mean values indicate better robustness across runs. Right: Consistency of explanations compared to ALE. Green lines denote mean values across samples. 227

List of Tables

1 Comparative Analysis of XAI Frameworks and Libraries 21

2 Performance Metrics for Cloud-Based AI Services with Different Data Augmentation Strategies 51

3 Quantitative Analysis of LoC for Algorithm Integration into Different Microservices 59

4 Comparative Energy Consumption of Microservices (Average per Data Sample, in 10^{-6} kWh) 61

5 Model and XAI Evaluation Results 71

6 Operational Overhead Comparison Between XAI Service and Tool Framework 104

7 Selected State-of-the-Art Vision Models from the Huggingface Repository . 106

8 Analysis of Computational Costs for Vision Models (per 1,000 images) . . 107

9 Summary of Orginal Deviation, Adversarial Deviation and Explanation Resilience for Vision Models 112

10 Analysis of Computational Costs for Tabular Models Based on per Thousand Rows 113

11 Analysis of Model Performance and XAI Deviation in Tabular Scenarios . . 114

12 Explanation Deviation and Energy Consumption of the Selected Models . . 116

13 Video Model XAI Evaluation Results 163

14 Computation Time Breakdown by Processing Stage (in seconds) 173

15 Comparison of Video Models: CNN-based, RNN-based, and Transformer-based Architectures 181
16 Recent Studies on Adversarial Attacks on Video Models 184
17 Impact of Common Corruptions on Image Model Performance 199
18 Impact of Common Corruptions on TimeSformer Video Model Performance 200
19 Performance comparison of different adversarial attack methods 206
20 Adversarial Training Configuration and Defense Effectiveness Against Various Attack Methods 210
21 Summary of feature importance order across XAI methods (feature name with importance score) 226
22 Feature Importance Order Summary of Code Vulnerability Detection Case Study 231

Chapter 1

Introduction

Artificial intelligence (AI) has transformed from a theoretical concept to a pervasive force across numerous domains, with deep learning models achieving success in computer vision, natural language processing, healthcare, and autonomous systems. These sophisticated models, while delivering remarkable performance, often operate as opaque black boxes, making their decision-making processes inscrutable to users and developers [46, 75]. This opacity becomes particularly problematic as AI systems increasingly influence critical decisions affecting human lives, safety, and well-being.

The field of explainable artificial intelligence (XAI) has thus emerged to address this fundamental challenge, seeking to develop techniques that render AI systems interpretable without sacrificing performance [13]. The need for explainability in AI systems stems from multiple imperatives. Regulatory frameworks such as the European Union's General Data Protection Regulation (GDPR) explicitly mandate that automated decisions affecting individuals be explainable. Industries deploying AI in high-stakes domains such as healthcare diagnostics [109], financial lending, and criminal justice require transparency to validate model decisions and ensure compliance with ethical and legal standards. Beyond compliance, explainability fosters trust among users, facilitates error detection during development, and enables effective human-AI collaboration [123].

Despite significant advances in XAI research, several critical gaps persist in current approaches. First, the landscape of XAI methods remains fragmented, with numerous techniques developed for specific model architectures or data types [85], yet limited standardization or evaluation protocols exist to guide practitioners. The divergence in explanations produced by different XAI methods for identical inputs further complicates the selection of appropriate techniques [124]. Second, as AI systems increasingly operate as cloud-based services [7, 69, 132], traditional XAI methods struggle to explain proprietary models whose parameters and architectures remain inaccessible. Third, most XAI techniques impose substantial computational overhead [94], limiting their practical deployment in resource-constrained environments or real-time applications. Fourth, while considerable work has addressed static data explanations, the interpretation of dynamic spatiotemporal data—essential for video analysis and time-series forecasting—remains underexplored. Finally, the robustness of explanations against adversarial perturbations has received insufficient attention [225], despite its critical importance for secure AI systems.

1.1 Current State and Fundamental Challenges

Fundamental problems impede effective XAI deployment in production systems. First of all, XAI methods lack standardized interfaces and deployment patterns. Each method requires custom integration work, making XAI adoption time-consuming and error-prone. Development teams often abandon XAI integration due to technical complexity rather than questionable value.

Secondly, commercial cloud AI services, which dominate production deployments, provide no explanation capabilities. Organizations cannot audit or interpret decisions from these black-box services, creating accountability gaps in critical applications.

Thirdly, different XAI methods produce conflicting explanations for identical inputs.

Without systematic evaluation frameworks, practitioners cannot assess explanation reliability or select appropriate methods for their specific requirements.

Lastly, video analysis and sequential data processing require explanation methods that capture temporal relationships. Current approaches analyze individual frames independently, ignoring the temporal patterns that drive model decisions.

1.2 Research Objectives

This research pursues five interconnected objectives, each addressing a specific gap in current XAI practices while collectively forming a unified framework for explainable AI services.

The first objective centers on designing a modular microservice architecture for XAI integration. The integration of explainability into AI development workflows currently requires substantial expertise and often occurs as an afterthought rather than an integral component of the development process [200]. This research creates an extensible, OpenAPI-compliant framework that facilitates the early integration of XAI functionality into MLOps pipelines, supporting reproducibility standards and minimizing operational overhead. Proprietary cloud models present unique challenges for XAI due to their inaccessible internal structures [40]. This research develops a cloud-agnostic attribution and auditing architecture that delivers model-agnostic feature explanations through unified APIs, enabling benchmarking and behavioral auditing of commercial cloud models without requiring access to their internal parameters.

The second objective focuses on formalizing evaluation metrics for explanation consistency. Current XAI methods often produce divergent explanations for identical inputs, creating uncertainty about their reliability [85, 112]. This research develops quantitative measures to assess the agreement and stability of explanations across different methods,

model architectures, and datasets. These metrics provide a systematic framework for evaluating XAI techniques and guide the selection of appropriate methods for specific applications.

As AI systems face increasingly sophisticated attacks [80], understanding how adversarial perturbations affect both model performance and explanation quality becomes essential for security. This research introduces a comprehensive service framework that systematically evaluates five quality attributes: computational cost, performance, robustness, explanation deviation, and explanation resilience, revealing critical insights into the vulnerability of explanations to adversarial attacks.

The third objective focuses on developing efficient XAI methods for spatiotemporal data. Video models pose distinct challenges for explainability due to their temporal dynamics and computational complexity. This research introduces a lightweight, transformer-specific XAI method that directly extracts and aggregates attention weights to produce spatial and temporal attributions in a single forward pass, significantly outperforming existing methods in both faithfulness and computational efficiency.

1.3 Research Contributions

This thesis makes four primary contributions to the field of explainable AI. Here, we briefly list the potential contributions of this work.

First, it presents a cloud-agnostic XAI service architecture that delivers model-agnostic feature explanations for proprietary cloud models through unified APIs. This architecture enables the benchmarking and behavioral auditing of commercial cloud AI services by evaluating explanation consistency, attribution provenance, and service quality, thus supporting explainability compliance in opaque AI ecosystems. The architecture is demonstrated across three major cloud platforms—Microsoft Azure Cognitive Services [132], Google Cloud Vertex AI [69], and Amazon Rekognition [7]—revealing insights into their

relative explainability characteristics and potential areas for improvement. XAIport supports the early-stage deployment of XAI functionality with minimal operational overhead, enabling practitioners to incorporate explainability as an integral component of AI development rather than a post-hoc consideration.

Second, it introduces an automated assessment framework for open repository models that systematically evaluates five quality attributes: computational cost, performance, robustness, explanation deviation, and explanation resilience [202]. Applied to both computer vision and tabular models, this framework reveals critical insights into the trade-offs between explanation quality and computational demands while demonstrating the previously underexplored vulnerability of explanations to adversarial attacks. The framework is demonstrated through extensive experiments across six vision models and three tabular models, providing empirical evidence of the relationship between model architecture, XAI method, and explanation quality.

Third, it presents a new XAI method for SOTA video models. Spatio-Temporal Attention Attribution (STAA) is a transformer-specific XAI method for video analysis. STAA directly extracts and aggregates attention weights from internal model layers to produce spatial and temporal attributions in a single forward pass. Enhanced with dynamic attention-thresholding and visual refinement, STAA outperforms existing model-agnostic methods such as SHAP [126] and LIME [154] in both faithfulness and monotonicity, while reducing explanation latency by over 97%, enabling its deployment in resource-constrained or real-time applications.

Fourth, it introduces an XAI-guided adversarial attack framework targeting video transformer models. This approach perturbs self-attention dynamics to simultaneously degrade both predictive performance and explanation stability, revealing critical vulnerabilities in temporal modeling pipelines. Through 3D attribution visualization and structural similarity analysis, the framework identifies failure modes in video models, demonstrating their

susceptibility to spatially coherent but temporally disruptive attacks and highlighting the importance of explanation robustness in security-critical applications.

Collectively, these contributions form a comprehensive and deployable framework for explainable, robust, and reproducible AI services, advancing the state of practice in trustworthy machine learning for both research and industrial applications.

1.4 Thesis Structure

This thesis is organized into six chapters. The structure is designed to guide the reader from foundational concepts and literature analysis to the architectural design, evaluation methodologies, and advanced applications of the framework, culminating in a comprehensive solution for trustworthy AI.

Chapter 1, Introduction, establishes the context for the entire dissertation. It articulates the need for XAI in systems, outlines the primary research objectives designed to address specific gaps in current practices, summarizes the principal contributions of this work, and presents the overall organizational structure of the thesis.

Chapter 2, Background and Literature Review, provides a comprehensive overview of the key domains that underpin this research. It delves into a detailed taxonomy of XAI methods, analyzes the distinct challenges of explaining proprietary cloud AI services, examines the principles of adversarial robustness, and outlines the unique requirements for explaining spatiotemporal data in video models. This chapter synthesizes existing knowledge to establish the technical context and identify the research gaps.

Chapter 3, A Unified Service Framework for Explainable AI, presents the core architectural contributions of this research. It details the design of a modular and cloud-agnostic microservice framework. This chapter describes the architectural patterns for integrating explainability into MLOps pipelines using an OpenAPI-driven approach. The objective is to enable configurable, reproducible, and automated XAI operations for both

commercial cloud platforms and open-source models.

Chapter 4, Methodology for Evaluating AI Models and Explanations, establishes the scientific methodology for assessment within the framework. It introduces a systematic process for evaluating both AI models and the explanations they generate. This includes the definition of metrics for quantifying explanation consistency and a comprehensive set of quality attributes, particularly under adversarial conditions.

Chapter 5, XAI Methods and Evaluations for Spatio-Temporal Video Model, addresses an advanced application domain by introducing novel techniques for video analysis. This chapter presents STAA (Spatio-Temporal Attention Attribution), a new XAI method specifically designed for video transformers that achieves high faithfulness and computational efficiency. It then introduces a novel joint spatiotemporal adversarial attack method that targets the attention mechanisms of video transformers.

Chapter 6, Conclusion and Future Work, concludes the dissertation by synthesizing the key findings and contributions. It reflects on the broader implications of the unified framework for the field of trustworthy AI. This chapter also discusses the limitations of the current research and proposes specific, promising directions for future work, outlining a roadmap for continued innovation in explainable, robust, and reproducible AI services.

1.5 Mapping of Publications and Thesis Chapters

There are multiple publications related to this thesis. Here we list these manuscripts:

Related to Chapter 3:

*[1] Z. Wang, Y. Liu and J. Huang, "An Open API Architecture to Discover the Trustworthy Explanation of Cloud AI Services," **IEEE Transactions on Cloud Computing**, vol. 12, no. 2, pp. 762-776, April-June 2024, doi: 10.1109/TCC.2024.3398609.*

*[2] Z. Wang, Y. Liu, A. A. Thiruselvi and A. Hamou-Lhadj, "XAIport: A Service Framework for the Early Adoption of XAI in AI Model Development," **2024 IEEE/ACM 46th***

***International Conference on Software Engineering (ICSE 24)**, Lisbon, Portugal, 2024, pp. 67-71, doi: 10.1145/3639476.3639759.*

Related to Chapter 4:

[3] J. Huang, Z. Wang*, D. Li and Y. Liu, "The Analysis and Development of an XAI Process on Feature Contribution Explanation," **2022 IEEE International Conference on Big Data (Big Data 22)**, Osaka, Japan, 2022, pp. 5039-5048, doi: 10.1109/ BigData55660.2022.10020313.*

*[4] Z. Wang and Y. Liu, "Cloud-Based XAI Services for Assessing Open Repository Models Under Adversarial Attacks," **2024 IEEE International Conference on Software Services Engineering (SSE 24)**, 2024, pp. 141-152, doi: 10.1109/ SSE62657.2024.00031.*

*[5] Z. Wang and Y. Liu, "XAIpipeline: Automated Orchestration of Explainable AI Service for Cloud AI and Open-Source Models," **2025 IEEE International Conference on Software Services Engineering (SSE 25)**.*

*[6] Z. Wang and Y. Liu, "The Role of Provenance Modeling in Tracing and Reproducing Explainable AI Pipelines" in **2025 World Congress in Computer Science, Computer Engineering, & Applied Computing (CSCE 25)**.*

Related to Chapter 5:

*[7] Z. Wang and Y. Liu, "STAA: Spatio-Temporal Attention Attribution for Real-Time Interpreting Transformer-Based AI Video Models," in **IEEE Access**, vol. 13, pp. 101647-101661, 2025, doi: 10.1109/ACCESS.2025.3575440.*

*[8] Z. Wang and Y. Liu, "Joint Spatio-temporal Adversarial Attacks on Video Transformer Models Through XAI-guided Gradient Perturbation," in **ACM Transactions on Multimedia Computing, Communications, and Applications**, 2025, doi: 10.1145/3766071*

*[9] Z. Wang and Y. Liu, "Spatio-temporal Explanation for Adversarial-Aware Cloud Vision Services." **IEEE Computer Society Signature Conference on Computers, Software, and Applications (COMPSAC 25)**, JC (Journal/Conference) program.*

The publications comprising this thesis demonstrate a systematic research progression from foundational concepts to advanced applications. This multi-year publication strategy serves to validate each component of the framework through rigorous peer review while reaching diverse research communities spanning software engineering, cloud computing, and artificial intelligence.

The research timeline began with foundational work on XAI processes and consistency metrics. Publication [3] (IEEE Big Data 2022) established the theoretical groundwork by formalizing the XAI process and introducing quantitative metrics for evaluating explanation consistency and stability across different methods. This early work revealed a critical gap: existing XAI frameworks lacked standardized evaluation protocols and practical deployment patterns, motivating the subsequent architectural contributions.

Building upon these foundations, Publications [1] and [2] (IEEE Transactions on Cloud Computing 2024 and ICSE 2024) address the practical challenge of integrating XAI into production AI systems. Publication [1] introduces a cloud-agnostic architecture that enables explainability for proprietary cloud models without requiring access to internal parameters, demonstrating the approach across Microsoft Azure Cognitive Services, Google Cloud Vertex AI, and Amazon Rekognition. Publication [2] complements this by presenting XAIport, a microservice framework that facilitates early-stage XAI adoption in MLOps pipelines, shifting explainability from a post-hoc activity to an integral development practice. Together, these works establish the architectural foundation presented in Chapter 3, enabling both cloud-based and open-source model explanation through unified interfaces.

The assessment methodology presented in Chapter 4 draws from four publications that systematically evaluate AI models and their explanations. Publication [4] (IEEE SSE 2024) extends the architectural framework by introducing a comprehensive assessment protocol for open-source models, evaluating five quality attributes, including adversarial robustness, explanation deviation, and explanation resilience. This work reveals that explanations

themselves are vulnerable to adversarial perturbations, a previously underexplored dimension of AI security. Publications [5] and [6] (IEEE SSE 2025 and CSCE 2025) advance the framework's automation capabilities by introducing orchestration mechanisms for complex multi-step XAI workflows and provenance modeling techniques that ensure reproducibility across experiments. These contributions collectively establish Chapter 4's methodology for systematic evaluation under diverse conditions.

The advanced applications in Chapter 5 represent the culmination of the framework's development, extending XAI capabilities to the challenging domain of video understanding. Publication [7] (IEEE Access 2025) introduces STAA (Spatio-Temporal Attention Attribution), a novel XAI method specifically designed for transformer-based video models. Unlike existing methods that analyze spatial and temporal dimensions independently, STAA extracts attention weights directly from model layers to produce unified spatio-temporal explanations in a single forward pass. Publications [8] and [9] (ACM TOMM and IEEE COMPSAC 2025) extend this work by investigating adversarial robustness in video transformers, developing joint spatio-temporal attack methods that simultaneously perturb spatial and temporal features while tracking explanation stability through XAI-guided analysis.

Chapter 2

Background and Literature Review

This chapter provides the background and context for understanding the research presented in this thesis. It begins with an exploration of the motivations driving explainable AI research, followed by a formal thesis statement. The chapter then examines the landscape of explainable AI methods, the challenges of cloud-based AI services, adversarial robustness considerations, and the unique requirements for explainability in spatiotemporal data.

2.1 The Imperative for Explainable AI

The rapid advancement and widespread deployment of artificial intelligence systems have created an urgent need for explainability across multiple dimensions. This section explores the primary motivations driving XAI research from regulatory, technical, ethical, and practical perspectives.

2.1.1 Regulatory and Compliance Requirements

Recent years have witnessed a global shift toward regulating automated decision-making systems, with explainability emerging as a central requirement. The European Union's General Data Protection Regulation (GDPR) includes a "right to explanation," requiring

organizations to provide meaningful information about the logic involved in automated decisions that significantly affect individuals [68]. Similarly, the proposed EU AI Act establishes tiered requirements for AI system transparency based on risk levels, with high-risk systems facing stringent explainability obligations. In the United States, sector-specific regulations such as the Fair Credit Reporting Act and Equal Credit Opportunity Act implicitly require financial institutions to explain algorithmic lending decisions [165]. These regulatory frameworks aim to protect individuals from potentially harmful algorithmic decisions by ensuring that such decisions can be scrutinized and challenged. For organizations deploying AI systems, achieving compliance with these regulations necessitates the development and implementation of explainability techniques that can translate complex model operations into understandable terms for both regulators and affected individuals [193].

2.1.2 Technical Imperatives for Model Development

Beyond regulatory concerns, explainability serves critical technical functions in the development and deployment of effective AI systems. During model development, explanations facilitate the identification of spurious correlations, dataset biases, and unintended behaviors that may not be apparent from performance metrics alone [159]. For instance, a model might achieve high accuracy by exploiting dataset artifacts rather than learning generalizable patterns, a problem that explainability techniques can help detect [108]. In model deployment and maintenance, explainability enables more effective error analysis and debugging. When models produce unexpected outputs in real-world settings, explanations can reveal whether these outputs stem from conceptual misalignments, distribution shifts, or implementation flaws [20]. This diagnostic capability is particularly valuable in complex deep learning systems where traditional debugging approaches prove insufficient [162]. The technical imperative for explainability extends to model selection and comparison as well. When evaluating alternative models that exhibit similar performance

metrics, explanations can reveal substantial differences in their decision-making processes, guiding developers toward models that align with domain knowledge and intended behaviors [126].

2.1.3 Ethical Considerations in AI Deployment

Ethical considerations represent another powerful motivator for XAI research. AI systems increasingly make or influence decisions with significant ethical implications, from criminal risk assessment to medical diagnostics and resource allocation. The opacity of these systems raises concerns about fairness, accountability, and potential discrimination [166]. Explainability helps address these concerns by enabling the detection and mitigation of algorithmic bias. By revealing how different features influence model outputs, XAI techniques allow practitioners to identify when protected attributes including race, gender, or age disproportionately affect decisions, either directly or through proxy variables [46]. This visibility supports both proactive bias mitigation during development and retrospective analysis of deployed systems. Beyond bias detection, explainability supports broader ethical governance of AI by facilitating meaningful human oversight. Complex AI systems can only be effectively supervised by humans if their decisions are sufficiently transparent [98]. Explanations bridge the gap between algorithmic complexity and human understanding, enabling stakeholders to exercise appropriate judgment about when to trust or override AI recommendations [133].

2.1.4 Practical User Needs and Trust Building

The practical adoption of AI systems across domains depends significantly on user trust, which in turn requires a degree of transparency about system behavior. Research in human-computer interaction has demonstrated that users are more likely to accept algorithmic decisions when they understand the underlying rationale, even if they disagree with the outcome [51]. Conversely, unexplained AI decisions often face resistance, particularly in high-stakes contexts [44]. Different stakeholders require different forms of explanation. Domain experts may need technical explanations that align with their specialized knowledge, allowing them to validate model reasoning against established domain principles [227]. End users typically benefit from simpler, more intuitive explanations that convey how their inputs influenced the outcome and what actions might lead to different results. System developers need detailed, actionable explanations that facilitate debugging and improvement [20]. In domains including healthcare, explainability serves both trust-building and practical clinical functions. Physicians are more likely to incorporate AI recommendations into their decision-making when they can assess the reasoning behind those recommendations [109]. Moreover, explanations that highlight relevant features in medical images or patient data can augment physicians' own observations, potentially improving diagnostic accuracy and providing educational value [27]. The importance of explainability increases proportionally with the stakes of the decisions being made. For low-risk applications such as movie recommendations, users may tolerate opacity in exchange for convenience. However, for high-stakes decisions affecting health, safety, liberty, or economic opportunity, explainability becomes an essential requirement for responsible deployment [159].

2.2 A Taxonomy of XAI Methods

The landscape of explainable AI (XAI) encompasses a diverse array of methods, each with distinct objectives and technical approaches. To provide a structured overview, we organize these methods into a clear taxonomy, as illustrated in Figure 1. This taxonomy categorizes XAI approaches based on their fundamental strategy, their relationship to the AI model, and the type of explanation they provide.

At the highest level, a primary distinction is drawn between creating **inherently interpretable models** and applying **post-hoc explanation methods** to complex, "black-box" models. The latter forms the primary focus of this thesis.

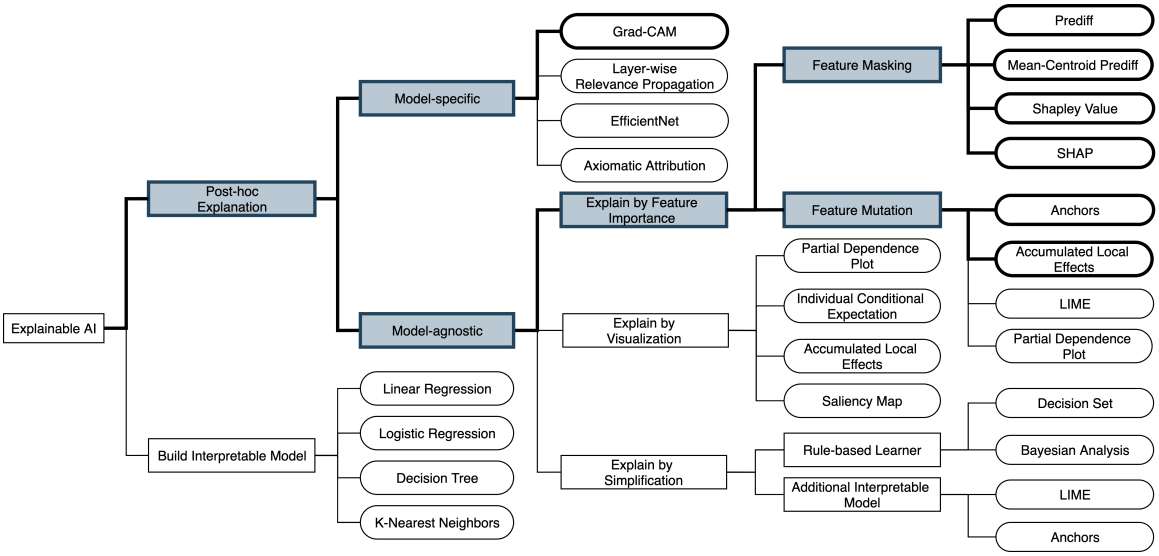


Figure 1: Taxonomy of explainable AI methods. The colored path indicates the methods deployed and analyzed in the case studies presented in this thesis.

2.2.1 Inherently Interpretable Models

The most direct path to explainability is to use models that are transparent by design [159]. Machine learning algorithms such as Linear Regression, Logistic Regression, and simple Decision Trees fall into this category. Their structures allow for the direct extraction of

decision rules, enabling a clear understanding of how inputs map to outputs. However, this transparency often comes at the cost of predictive performance, as these models may struggle to capture the complex, non-linear patterns that modern deep learning architectures handle with ease. In many of the high-stakes domains explored in this thesis, their performance is insufficient, necessitating the use of post-hoc explanation methods for more powerful but opaque models.

2.2.2 Post-Hoc Explanation Methods

Post-hoc methods aim to explain a model's behavior after it has been trained, without altering its internal structure [74]. This approach is essential for providing insights into the complex deep neural networks that dominate the current AI landscape. These methods can be further divided into two main categories: **model-specific** and **model-agnostic**.

Model-Specific Methods

Model-specific techniques are designed for a particular class of model architecture, leveraging its internal properties to generate explanations efficiently and with high fidelity.

A prominent example is the family of Class Activation Mapping (CAM) methods for Convolutional Neural Networks (CNNs). Techniques such as Grad-CAM [168] use the gradients flowing into the final convolutional layers to produce heatmaps that highlight the image regions most influential to a specific prediction. Other examples include Layer-wise Relevance Propagation (LRP) for Recurrent Neural Networks [12] and attention-based methods that directly interpret the self-attention weights in Transformer architectures [47, 120].

Model-Agnostic Methods

Model-agnostic methods offer greater flexibility by treating the AI model as an opaque black box, relying only on its input-output behavior to infer explanations. This makes them universally applicable to any model. These methods can be further classified by their explanatory strategy.

Explanation by Feature Importance. This is a major class of methods that assesses the influence of each input feature on the model's prediction. They typically operate in one of two ways:

- **Feature Masking:** Techniques such as SHAP (SHapley Additive exPlanations) [126] systematically remove or mask features (i.e., by replacing them with a baseline value) and measure the resulting impact on the model's output. SHAP is grounded in cooperative game theory and provides robust theoretical guarantees about feature contributions.

- **Feature Mutation:** Techniques such as Partial Dependence Plots (PDP) [61] work by systematically altering the value of one or more features across their range and observing the average effect on predictions. This helps reveal the marginal effect of a feature on the model's decision-making.

Explanation by Simplification. This strategy involves training a simpler, inherently interpretable "surrogate" model to approximate the behavior of the complex black-box model, either globally or, more commonly, locally.

- **Local Surrogate Models:** The most well-known method is LIME (Local Interpretable Model-agnostic Explanations) [154]. LIME explains an individual prediction by generating a neighborhood of perturbed data points around it and training a

simple model, such as a linear regressor, on these points to approximate the local decision boundary.

- **Rule-Based Learners:** Methods such as Anchors [155] aim to find a set of "if-then" rules that are sufficient to "anchor" a prediction, meaning the prediction holds for any other data point satisfying those rules.

The case studies throughout this thesis will draw upon a selection of these methods, as highlighted in Figure 1, to build and evaluate a comprehensive framework for explainable AI services.

2.3 Challenges in Cloud AI Service Explainability

The proliferation of cloud-based AI services has democratized access to advanced machine learning capabilities, enabling organizations without specialized expertise to incorporate AI into their applications. However, these services present unique challenges for explainability, stemming from their proprietary nature, limited access to model internals, and diverse deployment patterns.

2.3.1 Closed Source AI Services

Cloud-based AI services, which offer customized AI models and pre-trained models through APIs for various tasks, have attracted substantial interest due to their versatility and ease of use [132]. A study [40] provided a comparative analysis of cloud computer vision services, focusing on their accuracy, performance, and cost. However, the study did not delve into the specifics of the AI models used or draw conclusions based on the comparative evaluation. Image classification uses machine learning algorithms to categorize images based on their content, offering potential applications across various fields. The lack of explanation

can hinder the adoption and trust of these AI systems. For instance, a survey [54] illuminated the potential biases in visual datasets, emphasizing the necessity for bias discovery and quantification to ensure fairness and transparency in AI solutions. A recent publication introduces the tool named Threshy [39], which helps software developers assess an AI service's confidence score. Integrating configuration into client applications and monitoring systems represents an advance in the practical use and safe deployment of AI services. Both our work and Threshy [39], aim to assess cloud AI services outputs, but significantly differ in goals and methodologies. While Threshy focuses on decision threshold selection, we concentrate on XAI results from service.

Cloud AI services typically follow a service-oriented architecture, exposing machine learning functionality through well-defined APIs that abstract away the underlying implementation details [59]. This architectural pattern offers several advantages, including scalability, accessibility, and reduced operational overhead for end users. However, it also creates a significant barrier to explainability by restricting access to model parameters, intermediate representations, and training data—all of which may be instrumental for generating meaningful explanations [7,69,132]. The deployment patterns for cloud AI services vary across providers and use cases. Pre-trained models offered as services, such as computer vision APIs, natural language processing tools, or speech recognition systems, typically provide fixed functionality with limited customization options [7, 69, 132]. Custom model training services allow organizations to develop bespoke models using proprietary frameworks, but still restrict access to the underlying implementation once deployed [69]. Serverless machine learning deployments further abstract infrastructure management, enabling efficient scaling but potentially introducing additional opacity in resource allocation and performance characteristics [93]. These diverse deployment patterns create a complex landscape for XAI integration, requiring approaches that can adapt to varying levels of model access and service constraints. The architecture proposed in this thesis addresses

this challenge through a cloud-agnostic design that leverages available APIs to generate explanations without requiring access to model internals.

Cloud AI services pose several distinct challenges for explainability, exacerbating the general opacity of complex machine learning models with additional layers of abstraction and access limitations. These challenges can be categorized into technical, organizational, and methodological dimensions. From a technical perspective, the limited access to model internals constrains the application of many model-specific explanation methods that rely on gradients, feature maps, or attention weights [40]. The absence of direct control over model training or architecture selection further restricts opportunities for incorporating inherent interpretability or explanation-friendly designs. The variability in input processing, feature extraction, and output formatting across different services complicates the development of standardized explanation approaches [41]. Organizational challenges stem from the proprietary nature of cloud AI services, where providers may have limited incentives to expose model details or support third-party explanation tools that could reveal potential weaknesses or biases. The separation between service providers and end users creates an information asymmetry that complicates accountability for model decisions and explanations. The dynamic nature of cloud services, which may undergo frequent updates or modifications without explicit notification, introduces additional challenges for maintaining consistent and reliable explanations over time [41]. Methodologically, the black-box nature of cloud AI services necessitates a predominant reliance on model-agnostic explanation techniques, which generally incur higher computational costs and may provide less precise explanations than their model-specific counterparts [85]. The lack of access to training data or data distributions complicates the generation of counterfactual examples or baseline references often used in explanation methods. The diversity of services and use cases requires explanation approaches that can adapt to different input modalities, output formats, and decision types. Despite these challenges, the ubiquity and impact of cloud AI

services make their explainability an essential research priority. This thesis addresses these challenges through a novel architecture that bridges the gap between cloud services and explainability techniques, enabling comprehensive explanation and evaluation of proprietary models through standardized interfaces.

2.3.2 Frameworks and Packages for Implementing XAI

There are several frameworks and packages have been developed to facilitate the implementation of XAI. Dalex [15], for example, constructs a wrapper around prediction models to enable their exploration and comparison using a multitude of model-level and prediction-level explanations. IBM's Explainability 360 toolkit [14] incorporates an array of model explanation methods within its Python library. Meanwhile, Microsoft's InterpretML [144] supports eight XAI methodologies. Other libraries such as Captum [99] and OmniXAI [215] provide extensive collections of XAI techniques. Table 1 shows a comparative summary of various XAI frameworks.

Table 1: Comparative Analysis of XAI Frameworks and Libraries

Framework	**Supported Data Types**	**Supported Methods**	**Cloud Platform Deployment**
Dalex [15]	Tab	1, 2, 3, 4, 5	N/A
IBM AIX360 [14]	Tab/Image/Txt	1, 2, 6	Docker
InterpretML [144]	Tab/Txt	1, 2, 3, 10	N/A
Captum [99]	Image/Txt	1, 2, 6, 7, 8	N/A
OmniXAI [215]	Tab/Image/Txt	1, 2, 3, 4, 5, 6, 7	N/A
Vertex XAI [69]	Tab/Image/Txt	1, 2, 8	GCP service
XAI Service	Tab/Image/Txt	Encapsulate all methods and frameworks	API-based

Supported XAI Methods: 1. LIME [154], 2. SHAP [126], 3. PDP [61], 4. ICE [66], 5. ALE [9], 6. LRP [22], 7. CAM family [30, 48, 62, 92, 140, 168, 176], 8. Integrated Gradients [182], 9. Counterfactuals [194], 10. Decision Rules [214].
Note: Tab = Tabular, Txt = Text.

Different frameworks and libraries present unique characteristics and orientations. For instance, InterpretML [144] is primarily designed for tabular and minor text data, while

Captum [99] is specifically tailored for PyTorch models. Furthermore, these libraries exhibit significant differences in their interfaces, as highlighted by OmniXAI [215]. It is also worth noting that comparing the results or performance of these diverse tools across various use cases may prove challenging due to their distinct features and capabilities:

- **Expertise Requirements.** These frameworks require expertise for implementation, potentially limiting their accessibility to a broader user base. A paper [147] discusses the challenges and opportunities of developing user-friendly XAI tools that non-experts can use.

- **Limited Data and Model Types Support.** The support for various data types and models is limited [215], restricting their applicability across diverse custom models and databases.

- **Variation in Method Support.** There is a difference in the methods supported by frameworks. Practitioners encounter difficulties selecting the available methods and performing the necessary data preprocessing.

- **Insufficient Explanation Evaluation.** Many works [124, 226] states that XAI lack standardized evaluation procedures. The current frameworks often lack robust mechanisms for evaluating the explanation results, limiting their effectiveness and potential improvements [112].

- **Limited Support for Cloud AI Services.** These XAI frameworks rarely support cloud AI services, undermining their utility in multiple cloud platforms. Integrating configuration into client applications and monitoring systems represents an advance in the practical use and safe deployment of AI services.

These limitations suggest that the frameworks' inflexibility and extensive preparation may hinder the efficient application of XAI methods.

2.3.3 Data Provenance for XAI

Data provenance in the context of XAI is a component used to trace and reproduce operations. A review [207] explores the ethical considerations and presents the implementation of data provenance to ensure the AI system is responsible. Data provenance ensures transparency and accountability [207] of operations. The PROV-DM model [17] offers standardized components of representing provenance information. It defines concepts and relationships to capture entities involved in a process, activities that took place, and their interconnections.

Regarding practical implementations, a machine learning pipeline is proposed [163] emphasizing reproducible as a form of data provenance. Renku [101] is an open online platform tracking data, code, and results with Git. It assists researchers in evaluating, reproducing, and reusing data and algorithms. WholeTale [25] also promotes reproducibility by enabling researchers to capture and share data, code, and workflow. The work introduces a system [164] designed to extract, store, and manage both metadata and provenance information for common artifacts in machine learning experiments, including datasets, models, predictions, evaluations, and training runs. The experiment [24] enables provenance to be available as metadata. This study aims to enable the XAI service system to provide native provenance data. The XAI operations are reproducible, as provided graph-formatted provenance data, which includes datasets, models, XAI methods, and operational settings.

2.3.4 Current Solutions and Limitations

Current approaches to explaining cloud AI services fall into several categories, each with distinct advantages and limitations. Native explainability features provided by cloud vendors represent one approach, with services such as Google Cloud's Vertex Explainable AI [69], Amazon SageMaker Clarify, and Azure Machine Learning Interpretability offering built-in explanation capabilities for specific model types or use cases. While these native

features benefit from direct access to model internals, they typically support limited explanation methods, focus primarily on tabular data, and may not extend to all services within a provider's ecosystem. Third-party XAI frameworks such as LIME [154], SHAP [126], and Captum [99] offer more flexible and comprehensive explanation approaches that can be applied to cloud services as black boxes. However, these frameworks generally require significant expertise to implement, impose substantial computational overhead, and lack standardized integration patterns for cloud environments. The divergence in interfaces and output formats across different frameworks further complicates their adoption and comparison. Proxy model approaches represent another strategy, where a simpler, inherently interpretable model is trained to approximate the behavior of a complex cloud service [154]. While this approach can provide global insights into model behavior, it introduces additional uncertainty about the faithfulness of explanations to the original model and may not capture complex decision boundaries or feature interactions. Recent research has explored several promising directions for improving cloud AI explainability. Threshy [39] helps software developers assess AI service confidence scores by integrating configuration into client applications and monitoring systems. However, this tool focuses specifically on decision threshold selection rather than comprehensive explanation. XAI service frameworks [201] propose microservice architectures for delivering explainability as a service, but may lack integration with established XAI libraries or support for diverse data modalities. The current landscape of solutions for cloud AI explainability reveals several persistent limitations. The lack of standardization in explanation interfaces and outputs complicates comparison across different services or methods. The computational overhead of generating explanations, particularly for complex inputs such as images or time series, constrains real-time or large-scale applications. The absence of comprehensive evaluation frameworks for assessing explanation quality across different cloud services leaves users without clear guidance for selecting appropriate services or methods. This thesis addresses these limitations

through a unified framework that provides standardized interfaces for explaining diverse cloud services, integrates with established XAI libraries, and offers comprehensive evaluation metrics for assessing explanation quality across different services and methods. By addressing both architectural and methodological challenges, the research aims to advance more accessible and reliable approaches to cloud AI explainability.

2.4 XAI Evaluation Metrics and Adversarial Robustness

2.4.1 Evaluation of XAI Methods

The evaluation of XAI methods presents unique challenges distinct from traditional machine learning evaluation. While model performance can be assessed against ground truth labels, explanations often lack clear ground truth references, complicating their assessment [46]. The diversity of explainability objectives and stakeholder needs further complicates evaluation, as different metrics may prioritize different aspects of explanation quality [124]. Nevertheless, several evaluation approaches have emerged in the literature. Functionally-grounded evaluation assesses explanations against formal properties or theoretical guarantees, such as faithfulness, robustness, or completeness [13]. Application-grounded evaluation examines how explanations influence decision-making in specific domains, such as clinical diagnostics or financial lending [27]. For feature attribution methods specifically, several metrics have been proposed to assess explanation quality [118]. Faithfulness measures the degree to which attributed feature importance reflects the model's actual dependence on those features, often assessed by measuring performance degradation when important features are removed [85]. Consistency captures the stability of explanations across similar inputs or minor perturbations [64, 85]. Localization, particularly relevant for visual explanations, evaluates how precisely explanations identify relevant objects or regions in images, typically requiring ground truth annotations [118]. The computational

efficiency of explanation methods represents another important evaluation dimension, particularly for real-time applications or deployment on resource-constrained devices [202]. This aspect encompasses both the runtime and memory requirements of generating explanations, as well as any preprocessing or model modifications needed for explanation generation. Despite these advances, significant challenges remain in XAI evaluation. The lack of standardized benchmarks and metrics complicates comparative assessments of different methods [124]. The divergence in explanations produced by different methods for identical inputs raises questions about their reliability and trustworthiness [102]. The vulnerability of explanations to adversarial manipulations introduces additional concerns about their robustness in adversarial environments [225]. This thesis addresses these challenges by developing systematic evaluation protocols that quantify explanation consistency across different methods and assess explanation robustness under adversarial conditions. By establishing clear metrics and evaluation frameworks, the research aims to advance more rigorous and standardized approaches to XAI assessment.

As AI systems become increasingly integrated into critical applications, their vulnerability to adversarial manipulations has emerged as a significant concern. Adversarial examples—inputs specifically crafted to mislead models while remaining perceptually similar to legitimate inputs—pose serious challenges for the reliability and security of AI systems. This section explores the landscape of adversarial attacks, their implications for explainable AI, and the relationship between explainability and robustness.

2.4.2 Taxonomy of Adversarial Attacks

Adversarial attacks on machine learning models span a diverse spectrum of techniques, objectives, and constraints. These attacks can be taxonomized along several dimensions [198], providing a structured framework for understanding their characteristics and implications.

Based on the attacker's knowledge of the target model, adversarial attacks can be categorized as white-box or black-box [198]. White-box attacks assume complete access to the model's architecture, parameters, and gradients, enabling direct optimization of adversarial perturbations. Prominent white-box methods include the Fast Gradient Sign Method (FGSM) [67], which generates perturbations by following the sign of the gradient of the loss with respect to the input; C&W attacks [28], which formulate adversarial example generation as an optimization problem with customizable constraints; and DeepFool [138], which iteratively finds the minimal perturbation needed to cross a decision boundary. Black-box attacks, conversely, operate without direct access to model internals, relying instead on output probabilities, decision boundaries, or transferability properties. These approaches include query-based methods such as Zeroth Order Optimization (ZOO) [33], which estimate gradients through numerous model queries; transferability attacks, which leverage adversarial examples created for one model to attack another; and perturbation benchmarks such as ImageNet-C [80], which apply standardized corruptions to test model robustness. Adversarial attacks can also be distinguished by their targeting specificity. Targeted attacks aim to induce the model to predict a specific incorrect class, while non-targeted attacks seek any misclassification. Universal adversarial perturbations (UAPs) [137] represent another category, designed to cause misclassifications when applied to a wide range of inputs rather than being optimized for specific instances. The constraints imposed on adversarial perturbations provide another dimension for categorization. L_p-norm constraints limit the magnitude of perturbations according to various distance metrics, with L_0 (limiting the number of modified features), L_2 (limiting the Euclidean distance), and L_∞ (limiting the maximum change to any feature) being most common [28, 67]. Physical-world constraints address the requirements for attacks that must operate in real-world environments, such as robustness to image transformations or camera angles [104]. Recent years have witnessed the emergence of attacks specifically targeting

explanation methods [173], aiming to manipulate the explanations produced by XAI techniques while preserving the model's predictions. These attacks exploit the vulnerability of explanation methods to imperceptible input manipulations, raising concerns about the reliability of explanations in adversarial settings.

2.4.3 The Intersection of Robustness and Explainability

The proliferation of adversarial attacks carries profound implications for explainable AI, challenging both the reliability of explanations and their role in enhancing system security. Several key considerations emerge at the intersection of adversarial robustness and explainability. First, the vulnerability of explanation methods to adversarial manipulations raises fundamental questions about their trustworthiness. Research has demonstrated that saliency maps and feature attributions can be significantly altered through imperceptible input perturbations, creating explanations that highlight irrelevant features or obscure truly important ones [64]. This vulnerability is particularly concerning in high-stakes applications where explanations inform critical decisions or regulatory compliance. Second, the divergence between a model's decision process and its vulnerability to adversarial examples creates challenges for generating faithful explanations. Models that appear to rely on semantically meaningful features when processing natural inputs may actually depend on imperceptible patterns that adversarial examples exploit [88]. This disconnect complicates the task of explaining model behavior in a way that accounts for both normal operation and adversarial vulnerabilities. Third, the relationship between explanation fidelity and adversarial robustness remains incompletely understood. Some research suggests that models trained to align with human-interpretable features may exhibit greater robustness to certain adversarial attacks [158]. Conversely, robust training techniques such as adversarial training sometimes produce models whose explanations better align with human perception [52]. However, these relationships are not universal and depend significantly on the

specific models, datasets, and attack methods involved. Fourth, explanations themselves may serve both defensive and offensive roles in adversarial settings. On the defensive side, explanations can help identify potential vulnerabilities by revealing unexpected feature dependencies or decision boundaries [185]. On the offensive side, knowledge of how a model makes decisions, particularly which features it prioritizes, can facilitate more effective targeted attacks [67]. The dynamic interplay between adversarial robustness and explainability underscores the need for research that addresses both properties simultaneously. This thesis contributes to this emerging area by developing methods to assess explanation robustness under adversarial conditions and exploring how explainability techniques can inform the understanding of model vulnerabilities.

2.5 Challenges in Video Model Explainability

The explanation of models operating on spatiotemporal data, such as videos, time series, or sequential interactions, presents unique challenges that transcend those encountered in static data domains. This section explores these challenges, the current approaches to spatiotemporal explainability, and the open research questions that this thesis addresses.

Spatiotemporal data differs fundamentally from static data in its incorporation of temporal dependencies and dynamic patterns that evolve over time. This temporal dimension introduces several distinct challenges for explainability. First, the relevance of features often varies across time, with certain time steps or sequences holding disproportionate importance for predictions [76]. Traditional feature attribution methods that assign fixed importance scores to spatial features fail to capture this temporal variability, potentially leading to incomplete or misleading explanations. Second, temporal dependencies and causality relationships between features at different time steps create complex interaction patterns that simple attribution methods may not adequately represent [73]. Explaining these relationships requires methods that can capture how information flows through the

model over time and how early inputs influence later predictions. Third, the computational complexity of generating explanations scales with the temporal dimension, creating significant efficiency challenges for video or long sequence analysis [94]. Direct application of frame-by-frame explanation techniques incurs prohibitive computational costs and may not capture cross-frame dependencies that influence predictions. Fourth, the visualization and communication of spatiotemporal explanations present additional challenges. While static feature attributions can be visualized through heatmaps or feature importance bars, temporal explanations require more sophisticated visualization approaches that can represent both spatial and temporal attribution patterns effectively. These challenges are particularly pronounced for transformer-based video models, which have gained prominence for their ability to capture long-range dependencies and complex spatiotemporal patterns [47, 120]. The multi-head attention mechanisms in these models create rich but intricate representations that conventional explanation methods struggle to interpret faithfully and efficiently.

Chapter 3

A Unified Service Framework for Explainable AI

This chapter presents the design of an open-API-based explainable AI (XAI) service to provide feature contribution explanations for cloud AI services. Cloud AI services are widely used to develop domain-specific applications with precise learning metrics. However, the underlying cloud AI services remain opaque on how the model produces the prediction. We argue that XAI operations are accessible as open APIs to enable the consolidation of XAI operations into the cloud AI services assessment. We propose a design using a microservice architecture that offers feature contribution explanations for cloud AI services without unfolding the network structure of the cloud models. We can also utilize this architecture to evaluate the model performance and XAI consistency metrics showing cloud AI services' trustworthiness. We collect provenance data from operational pipelines to enable reproducibility within the XAI service. Furthermore, we present the discovery scenarios for the experimental tests regarding model performance and XAI consistency metrics for the leading cloud vision AI services. The results confirm that the architecture, based on open APIs, is cloud-agnostic. Additionally, data augmentations result in measurable improvements in XAI consistency metrics for cloud AI services.

3.1 An Open API Architecture to Discover the Trustworthy Explanation of Cloud AI Services

Artificial Intelligence as a service is a rapidly expanding technology paradigm. The AI market is expected to grow as more companies adopt AI technology to remain competitive. Major cloud providers, including Amazon Web Services [7], Microsoft Azure [132], Google Cloud Platform [69], Alibaba Cloud [5], Oracle Cloud, IBM Cloud, and Salesforce Service Cloud, offer AI as a service. This enables customers to develop and deploy AI models using cloud-based platforms.

Cloud AI services commonly achieve learning accuracy by using standard metrics such as precision, recall, and F1 score [7, 69, 132]. However, certain services such as Alibaba Cloud [5] may not explicitly provide these performance metrics. Additionally, a recent study [41] conducts a detailed investigation into the maintenance of AI services, focusing on computer vision. This research uncovers inconsistencies and evolution risks in AI services.

The explainable AI (XAI) aims to develop models and methods that enable human users to comprehend, trust, and manage AI models [75]. A study [131] discusses the role of XAI in computer vision-based decisions. They emphasize that XAI can promote trust in AI computer vision systems through improved understanding and prediction. Another work [85] presents the XAI criterion that refines the functionality objectives for XAI methods. One criterion involves the analysis of feature influence and feature causality.

XAI is increasingly adopted in applications that require transparency, fairness, and trustworthiness in decision-making, particularly in sensitive domains [6]. The limitation of the XAI practices is the existing XAI techniques developed are often tailored to specific types of models or cases [215], which makes the XAI practices less reusable and versatile to other applications. Meanwhile, numerous cloud AI services have been provided by

cloud platforms to support general applications across domains [5,7,69,132]. Together, the trend underscores the need to integrate XAI methods with cloud AI services to foster trust in cloud-based AI applications.

XAI demands that activities conducted during an XAI process be traceable and reproducible [221]. Ensuring that the data utilized in the AI models and XAI methods are trustworthy becomes important. Addressing data provenance in XAI operations is essential for guaranteeing that the generated explanations are reliable, verifiable, and consistently reproducible across diverse settings.

Despite the numerous XAI frameworks available, a noticeable gap exists among essential components [215] including data processing, methods configuration, and evaluation metrics. These components collectively form complex pipelines. This observation motivates our proposal: a design by a cloud-native paradigm based on the microservice architecture. This architectural style benefits from its capacity for the independent deployment of diverse components, streamlined communication via RESTful APIs, and a built-in adaptability that accommodates the introduction of new XAI methods, substitutions of AI models, and adjustments in pipeline configurations. Furthermore, our proposed architecture offers precise record capabilities. This ensures that the provenance of every XAI operation is transparent, facilitating the reproduction of XAI tasks or entire workflows.

For black-box models or cloud-based AI services, the task of revealing the internal structures of AI models becomes unfeasible, especially for those AI services that encapsulate models behind standard RESTful APIs. We address this challenge by drawing inspiration from XAI methods that focus on feature influence and causality, for example, SHAP [126]. Besides, we propose a method that approximates the black-box AI model with a custom-built model and computes the feature contribution values, providing interpretable insights even from opaque AI models. Confronted by these challenges, our work is directed towards the following research questions. How to obtain and evaluate XAI results

without unfolding the cloud AI service model structure? What are the essential components required for XAI service architecture to deliver feature contribution explanations for models? How to collect XAI provenance data from operations to ensure traceability within the XAI service?

In this work, we propose an innovative XAI service architecture specifically designed to feature contribution explanations, illustrated through a showcase scenario drawn from computer vision cloud AI services. This method involves the utilization of approximation models to generate images, emphasizing the most contributing features. These masked images then act as inputs to create the AI services' predictions. We calculate the prediction changes value between the original and masked images. Leveraging these prediction changes, we compute a comprehensive explanation summary for the AI services, providing a transparent overview.

The main contributions are summarized as follows:

- **Design cloud-platform-independent XAI service framework.** The open API architecture is independent of the cloud-specific AI service. The architecture accommodates first-class entities in the XAI process as unified micro-services. The communication is open API-based, thus encapsulating the variance of models, XAI methods, and inputs and outputs from feature engineering.

- **Provide explanations across multiple cloud AI services.** Based on the definition of the XAI consistency metric, we derive an explanation summary cross-validated on multiple clouds to observe both the learning performance of AI services and data augmentation effects.

- **Reproduce XAI operations through configure-and-rerun.** The configuration of services is the receipt of composing an end-to-end explanation workflow. By reserving the configuration of each service given a workflow definition, we accumulate the

provenance of how each explanation is produced. Through the coordination center of the XAI framework, we can rerun the XAI workflow to reproduce the explanation.

The adoption of XAI frameworks is designed for data science and machine learning engineers, effectively functioning as a tool for assessment in the development of complex AI systems. A recent study [143] proposes a multi-level governance pattern that integrates team-level XAI practices with organization-level ethical standards, thereby organizing ethical principles. This work introduces an XAI service framework for AI service practitioners, ensuring alignment with ethical guidelines and organizational values.

3.1.1 XAI Service Framework Foundation and Implementation

This section establishes the foundational components necessary for implementing a comprehensive XAI service framework. The implementation requires careful consideration of both technical specifications and architectural design patterns to ensure scalability, interoperability, and practical deployment across diverse environments.

XAI Methods Selection

We provide a comprehensive background on the selected XAI techniques and evaluation metrics.

Methods for feature contribution explanation reveal and visualize the correlations between specific content elements, such as pixels in an image, and the resulting decision from a model. In explaining vision tasks, Class Activation Mapping (CAM) [228] initially employs the global average pooling layer to localize features in Convolutional Neural Networks (CNNs). However, the method needs to modify the original layer of the model. As optimizations, Grad-CAM obtains [168] the localization weight by the average gradient of one layer instead of replacing it. Grad-CAM++ [30] is an improved version that uses

second-order gradients. EigenCAM [140] takes the first principle component of the activation without class discrimination. LayerCAM [92] spatially weight the activations by positive gradients. XGrad-CAM [62] scales the gradients by the normalized activations.

The example images of the explanation of the CAM-based method for vision AI tasks are shown in Figure 2. The saliency maps in the second row highlight the regions in the image that contribute to the model prediction.



Figure 2: Grad-CAM Visual Explanations on ImageNet Dataset Samples

Metrics for Evaluating XAI

The previous work [85] reviews and compares XAI evaluation criteria and metrics. To compare the XAI outcomes objectively, we use consistency metrics [85] to perform the assessment and comparison of various XAI methods across identical datasets. We outline the specific equation to compute the intra-consistency [85], also known as the XAI stability.

The Prediction Change Value metrics are based on the feature importance, also present as feature contribution, provided by XAI methods. Assume $\hat{f}(x^{[i]})$ denote the model's predicted value for a dataset $x^{[i]}$. We use $x^{[i]+}$ to symbolize the transformed or augmented data sample, an image under masking. We introduce an equation to compute the difference

value induced by the feature, denoted as the prediction change value in Equations 4.

$$\delta(x^{[i]})|_{rel} = \left| \frac{\hat{f}(x^{[i]+}) - \hat{f}(x^{[i]})}{\hat{f}(x^{[i]})} \right| \tag{1}$$

Additionally, the prediction change values of the entire dataset X can be aggregated. The experiment, in Section 3.1.2, uses histograms to present the distribution of prediction changes.

The XAI consistency metrics as described [85] include both *Intra-Method Consistency* and *Inter-Method Consistency*. In our study, the objective is not to compare inter-XAI methods. Therefore, we specifically focus on *Intra-Method Consistency*, also known as XAI stability evaluation, to assess the quality of the generated explanations by a series of XAI methods.

Here, we present the stability evaluation. We define the set of explanations, denoted by $E = \{\xi^1, \xi^2, \ldots, \xi^m\}$, where ξ^i represents the i-th data sample explanation. Next, we define the distance between any two pairs of summaries as $f_d(\xi^i, \xi^j)$, where f_d is the distance function, and ξ^i and ξ^j are two different explanation. Then, we describe the combination of choices for selecting any pair of explanations from E, given by $K = \binom{m}{2}$. This combination represents the number of ways to choose two explanations from the m in total.

$$f_d^K = \frac{1}{K} \sum_{k=1}^{K} f_d(\xi^i, \xi^j), \quad (i \neq j, i \leq m, j \leq m) \tag{2}$$

Here, f_d^K represents the stability metric computed as the average over all K distance values, where each distance value f_d is calculated for a specific pair of explanation ξ^i and ξ^j. This metric is used for the assessment of various XAI methods in scenario three, Section 3.1.2. In this case, the distance value f_d is represented by the prediction change value.

Integrating Cloud AI with Post hoc XAI

In this section, we discuss the post-hoc XAI methods. We explore feature contribution explanation techniques. Then, we present a design that integrates feature contribution XAI methods with cloud AI services. Cloud AI services encapsulate AI models with well-defined RESTful APIs [7, 69, 132]. The opacity of the AI models has limitations on the comprehension of how the prediction is made, even though the prediction produces a high accuracy level. It is a limitation for a certain adoption of the AI services that has an emphasis on transparency of the AI models.

We use post hoc XAI methods for feature influence and causality to devise an approximating model that correlates the inputs and outputs to generate an explanation of feature contribution values. Following this principle, we consider computer vision AI service suitable for adopting post hoc XAI methods for two reasons: (1) A variety of state-of-the-art deep learning models already exist in the field of computer vision as standard models. Even if there is no access to the AI models running within a cloud-based AI service, these established models can serve as suitable approximations; (2) XAI methods specifically designed for elucidating computer vision model explanations are widely available in the literature.

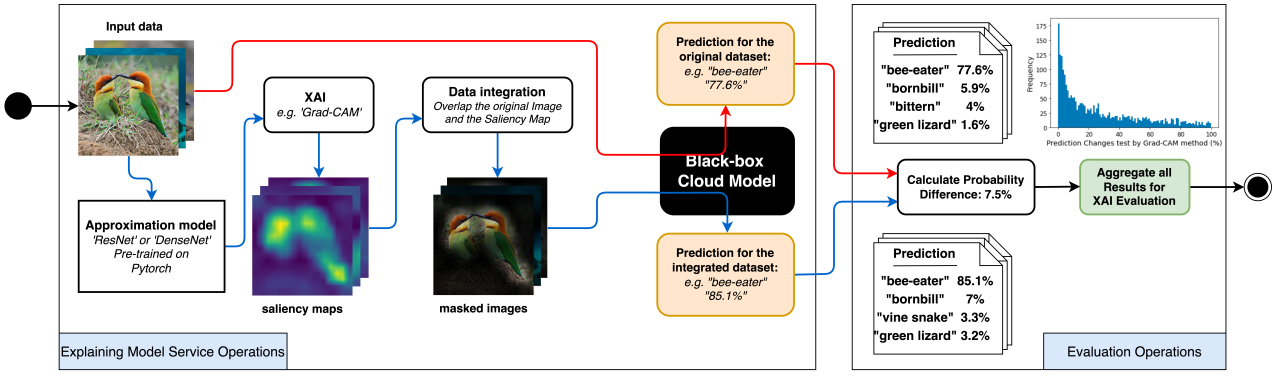


Figure 3: Workflow for Integrating Cloud AI Services with State-of-the-Art Deep Learning Models to Approximate Feature Contribution Values Using Post-Hoc XAI Methods

We describe the process and data flow in Figure 3 at the run-time. The initial step involves the preparation of the input dataset. Given that cloud model services typically function as a black box, with internal parameters and model structure concealed from the user, we employ approximation models. The original dataset is then used in three distinct steps: 1) Input images are directly submitted to the cloud-based AI service, which returns their prediction outcomes. 2) The images are processed by an approximation model, employing model-specific XAI methods [30, 48, 62, 92, 140, 168, 176] to generate salience maps. 3) The original images, overlaid with the salience maps, are used to create masked images highlighting only the important features. Subsequently, both the original and masked images are submitted to the cloud AI service for inference, yielding model predictions and confidence values. Comparing these two sets of results allows us to quantify the impact of features. The minimal variance between the two sets indicates similar feature importance values. We also evaluate the XAI consistency metrics, offering insights into the XAI evaluations within cloud AI services.

In summary, the exploration showed that post hoc XAI methods, which elucidate the correlation between inputs and outputs, can approximate feature contributions for cloud AI services. As illustrated in Figure 3, the workflow produces explanations for cloud model predictions.

XAI Service Architecture and API Design

We design the XAI operations as well as a comprehensive API architecture. This approach led us to define a reference architecture comprising four distinct layers from top to bottom. They are the user interface, the coordination center, the core microservice, and the data persistence. The overview of this reference architecture is illustrated in Figure 4.

API Design for Microservices

One of the key features of this proposed architecture is its non-intrusive application

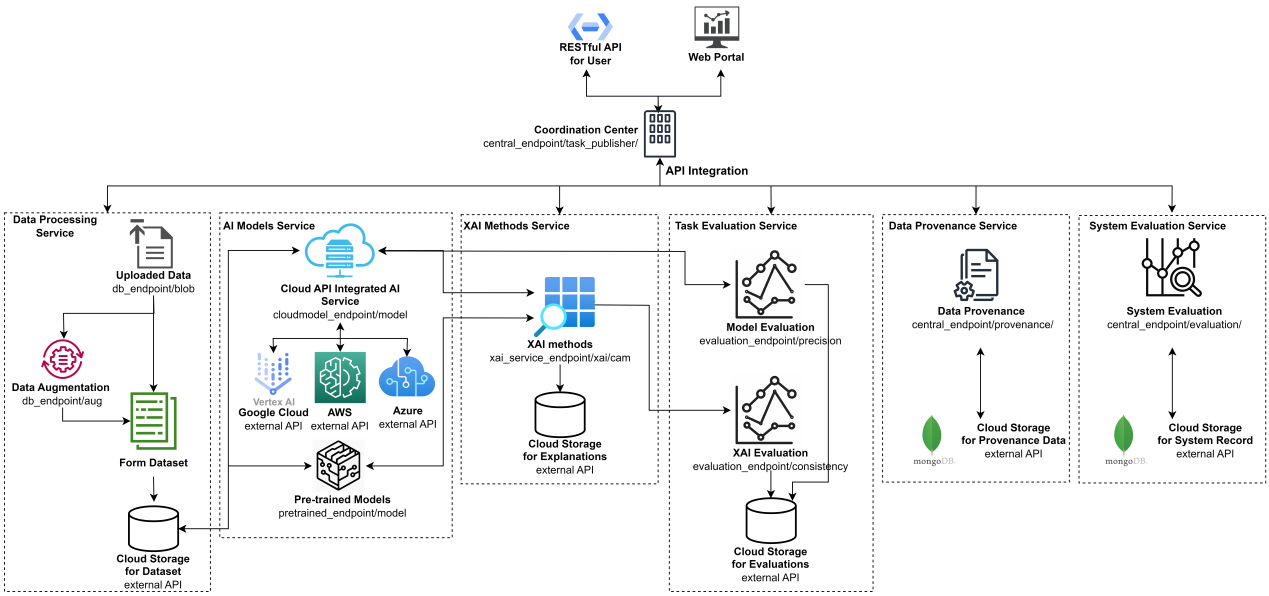


Figure 4: Microservice-Based Reference Architecture for XAI Service

of XAI methods. This approach allows us to apply XAI techniques without significantly disrupting or modifying existing system operations. We have completed the API design for this microservice system, with these APIs systematically arranged and hosted on Swagger Hub, adhering to OpenAPI 3.0 standards. Detailed API documentation featuring an array of functional interfaces is available in our published Swagger document.[1].

API-based Coordination Center

The coordination center processes requests from the user interface layer and activates the necessary components for task execution. The core functions of the coordination center are managing microservices, overseeing XAI tasks, and retrieving data.

To manage microservices, the coordination center works with the user interface to handle service registration and modification. It logs endpoints and functional types and initiates the corresponding microservice upon request.

In XAI task management, users submit a task sheet outlining the task type (either XAI

[1] https://app.swaggerhub.com/apis-docs/WANGZERUI418/xaiservice/1.0/

or evaluation), associated microservices, dataset, and specific parameters. The center oversees the entire task lifecycle, using the submitted task sheet as a blueprint. During execution, the center accesses data, activates the necessary service APIs, and ensures task completion. Upon task completion, the executor updates the center with the task status, runtime metrics, and provenance data. Additionally, the coordination center offers a pipeline configuration that sequences task sheets for standard execution of XAI and evaluation tasks.

Data Processing Microservice

Data Processing serves as a critical component in the system architecture. It is primarily responsible for managing data inflow into the system and orchestrating data processing tasks. This microservice takes into account the need for streamlined data management, providing interfaces for users to upload their datasets in specified formats, organize data, and maintain the consistency and integrity of the data.

Data Processing allows users to not only upload their data but also assemble them into datasets. Furthermore, it provides the option to apply data augmentation techniques to enhance the variability and volume of the original data. Data augmentation is instrumental in enhancing the robustness of the AI models and improving the performance of XAI methods. The Data Processing Microservice stores data in the user-specified cloud database to ensure data security and ownership.

This design addresses multiple aspects of data management, including data import, processing, augmentation, and secure storage. The incorporation of these features ensures that the Data Processing Microservice not only maintains the data lifecycle within the system but also provides users with flexibility and control over their own data.

AI Model Microservice

The AI Model Microservice is specifically designed to handle prediction tasks. This

microservice is flexible in integrating various cloud AI services from cloud platforms including Amazon Web Services, Google Cloud Platform, and Microsoft Azure. It communicates with these platforms using their specific APIs, ensuring seamless connectivity and usage of the cloud AI services. By connecting with these platforms, users are given access to a broad spectrum of AI models and tools readily available on the cloud.

Another notable feature of the AI Model Microservice is the ability to integrate pretrained models provided by the users. The integration of user-provided models is facilitated through a well-defined RESTful API specification. Users need to ensure their models adhere to this specification, and once that is done, their models can be easily plugged into the system.

XAI Method Microservice

The XAI Method Microservice in the system's architecture provides explanations for AI model predictions. These XAI methods work based on the contributions of individual features to the final prediction. Once an XAI method is integrated into the system, it can be repeatedly used for various similar types of XAI tasks without the need to consider the development steps again. This approach not only simplifies the procedure of applying XAI methods but also enhances the efficiency of the XAI operations.

Furthermore, this microservice provides a seamless connection with the Data Processing Microservice and the AI Model Microservice through RESTful APIs. This enables it to access data and models, generate explanations, and provide users with comprehensible insights into the AI models.

Evaluation Microservice

The Evaluation in the system brings forward a reliable approach to assess both the AI models and the deployed XAI methods. It is crucial to guide AI practitioners in their decision-making process and enhance trust in the AI models and their explanations. This

service is equipped to evaluate the performance of AI models and XAI methods by examining the explanation results. It employs the consistency evaluation method referenced in [85]. This method provides a comprehensive evaluation of the stability and reliability of the XAI methods, thus ensuring that the selected XAI methods yield consistent explanations.

The results derived from this evaluation are efficiently stored and maintained for future reference. This also allows for continuous monitoring and comparison of different models and methods, paving the way for continuous improvement and development in the system.

In summary, the proposed architecture provides a comprehensive solution for implementing XAI as a service. It consists of several key components: the User Interface, Coordination Center, Data Processing Microservice, AI Model Microservice, XAI Method Microservice, and Evaluation Microservices. These components work synergistically to enable the effective execution and evaluation of XAI tasks.

Design Provenance Meta Data of XAI Operations

The purpose of provenance data is to trace and reproduce XAI operations. The design of microservices brings another benefit in addition to the integration of heterogeneous components involved in XAI operations. The design helps us understand the relations of these components, focusing on their characteristics with abstraction. For example, we can retrieve the provenance data of two XAI pipelines and identify the difference in the configuration. This helps trace and observe how a certain explanation is produced, improving transparency and trustworthiness.

To this purpose, the provenance data needs to capture the properties of XAI services, the configuration of XAI tasks and pipelines, an explanation summary from the XAI method and the data set, and XAI cross-validation results. We apply graph format data to organize, visualize, and understand the relations between XAI provenance data.

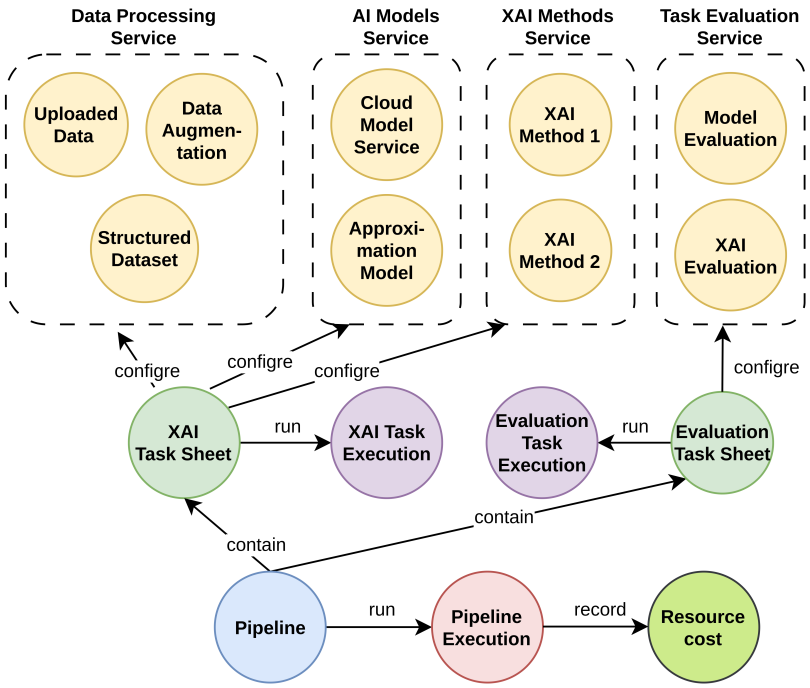


Figure 5: Directed Graph Model of XAI Provenance Data Capturing Microservices, Tasks, and Pipeline Relationships

We model the provenance metadata shown in Figure 5. The nodes represent types of XAI provenance data, and the edges represent the relationships between them. There are four types of microservices as the yellow nodes in the graph-format provenance data: (1) database, (2) AI, (3) XAI, and (4) evaluation. These nodes contain the entities that describe the metadata of the microservices, for example, the unique ID, the type, the name, and other note information of the microservice.

The task sheets are of two types: (1) XAI task sheets and (2) evaluation task sheets. The two task sheet nodes are independent because they deliver different final results, even though they share certain similar components. They record the metadata attributes, including the unique ID, the task type, the task name, the setting parameter, and other detailed configuration information. When users execute a task sheet, it generates a unique metadata profile represented as the violet color nodes. These execution nodes capture details including the execution ticket, associated task sheet ID, time of execution, status, results data,

and resource consumption.

The pipeline node at the bottom of the graph data typically includes both XAI and evaluation tasks, facilitating comprehensive XAI operations for specific scenarios and settings. The metadata in the pipeline node comprises the pipeline ID, name, and associated task sheet ID. Initiating the pipeline generates a pipeline execution node, represented as a distinct node in the graph-structured provenance data. The pipeline execution node includes log details. Computational costs are recorded following the execution.

This section presents configuration settings of each microservice in the framework. The database microservice uses the ImageNet dataset [43], ensuring efficient data storage and access during testing phases. The model microservice integrates cloud AI services via Restful APIs. XAI microservices are responsible for generating explanations in saliency maps. The coordination center effectively manages these diverse microservices, ensuring a streamlined execution of tasks leading to the creation of saliency maps. Our evaluation task involves validating explanations produced by approximation models. Additionally, we apply data augmentation techniques to explore potential improvements in model and XAI performance. The source code for implementation of the XAI service discovery scenarios is available in the GitHub Repository [2].

Database Microservice. We implement the database microservice using Azure Blob, a cloud database service that manages data uploading and retrieval for our case studies.

Integrate Cloud AI Service. Incorporating the APIs of cloud AI models into a microservices infrastructure is crucial for completing XAI service tasks. Post-hoc XAI methods need access to AI model results in response to various inputs. This integration provides a unified interface and pipeline for executing XAI experiments.

Integrate Pre-trained Models. We employ ResNet [79] and DenseNet [84] as approximation models due to their exceptional performance on ImageNet image classification

[2] `https://github.com/ZeruiW/XAI-Service`

benchmarks [43]. The pre-trained models are encapsulated within microservices, and their output is standardized to ensure compatibility with other services.

XAI Methods Microservice. We interpret image classification models using CAM-based methods such as Grad-CAM [168], Grad-CAM++ [30], EigenCAM [140], Layer-CAM [92], and XGrad-CAM [62]. CAM-based methods generate saliency maps, offering insights into the attention of the convolutional neural network during prediction. These maps highlight key pixels or areas in an image that significantly influence the prediction results of the model. These methods are encapsulated into microservices, allowing them to download model parameters and dispatch explanations through the coordination center. We establish an API that facilitates the retrieval of model parameters. This step is critical as it bridges the gap between the XAI and AI services, which are standalone applications.

Evaluation Microservice. We have implemented an evaluation microservice to assess explanations generated by XAI methods, using the previously defined evaluation metrics [85]. The `stability` forms the backbone of our evaluative procedures. The evaluation microservice executes the algorithm and forwards the results to the coordination center, offering a holistic view of XAI service performance. This structure enables a comprehensive assessment of explanations, thereby enhancing the quality assurance of XAI methods in our framework.

3.1.2 Framework Validation Through Scenario Studies

This section validates the proposed XAI service framework through four comprehensive scenarios that demonstrate its practical effectiveness across different deployment contexts and evaluation dimensions.

Scenario One: XAI Consistency Evaluation

Attribute-based post-hoc XAI methods produce explanations by applying feature masking and mutation, noting changes in predictions. Evaluating the variation in explanations across different XAI methods is essential to ensure their trustworthiness. Trustworthiness and transparency are critical technical requirements when applying XAI methods to AI models [123]. Improved flow by rephrasing

As shown in Figure 3, the evaluation consists of three stages, beginning with the preparation of feature changes. cloud AI services cannot directly derive saliency maps. To measure changes from feature masking and mutation, we generate saliency maps using approximation models, applying an XAI method. Next, we overlay the saliency maps onto the original images to create marked images for input into the cloud AI services. Finally, processing both masked and original images through cloud AI services generates two sets of predictions. The differences in predictions are analyzed to explain the importance of the features identified by the saliency maps, using XAI. Figure 3 demonstrates the workflow to approximate feature contribution exploration with evaluation. It should be noted that the above inputs are from the test datasets, not from the training dataset. Then, we use the XAI methods and model predictions to compute prediction changes. Finally, we apply consistency metrics to provide the summarized evaluation.

In the initial stage, we focus on employing a valid approximation model, as outlined in Figure 3. We consider two candidate approximation models, ResNet [79] and DenseNet [84]. As a result, Figure 6 illustrates the comparative analysis of prediction change values, highlighting the statistical differences between approximation models. Although the differences in prediction change values between ResNet and DenseNet were subtle, ResNet exhibited lower mean change values during the assessment, indicating superior XAI metrics. Lower computed prediction change values correlate with better XAI consistency metrics. Therefore, we selected the better option, ResNet, as the approximation model for our

subsequent analyses.

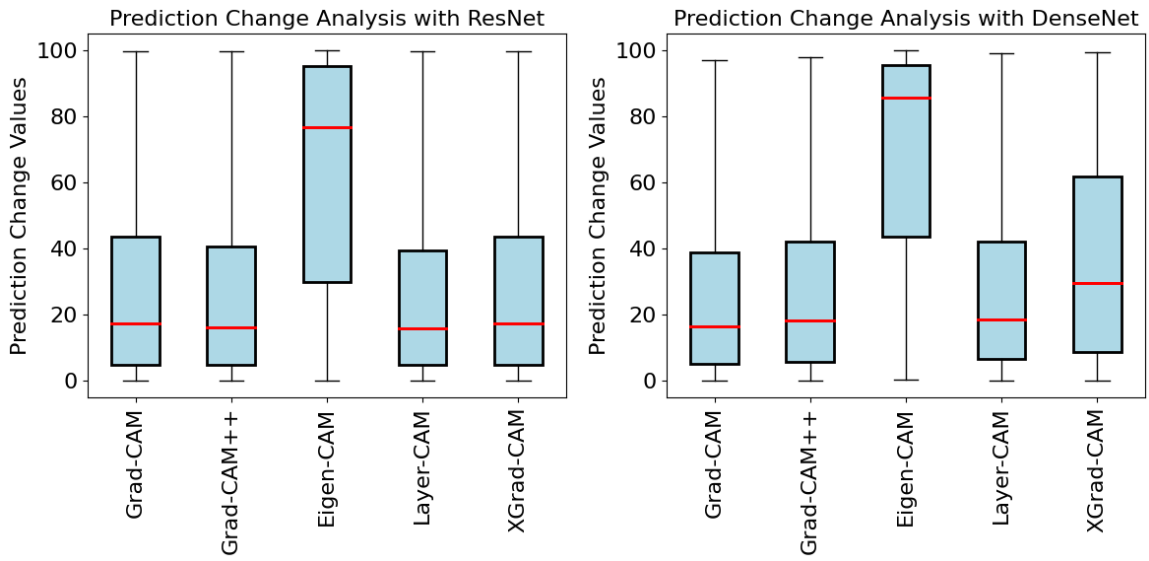


Figure 6: Comparison of Prediction Change Values for ResNet and DenseNet Approximation Models in CAM-based XAI Methods

We analyze XAI Consistency Evaluation for Azure Cognitive Services [132], Google Vertex AI [69], and Amazon Rekognition [7], use CAM-based XAI methods and measure the explainability. Subsequently, we aggregate the explanation outcomes across the dataset.

The detailed statistical analysis is presented in the first row of Figure 7, where Eigen-CAM displays notably high prediction change values. The analysis reveals a significant disparity in the performance of EigenCAM compared to other CAM-based methods. For EigenCAM specifically, 64.7% of the data showed prediction changes exceeding half, indicating that the explanations provided by EigenCAM do not accurately reflect the model's predictions. Among other XAI methods, Layer-CAM performs the best, causing the fewest prediction changes across the test set. The overall performance of these four XAI methods remains comparable.

For a precise comparison of the prediction change values across various XAI methods, we employed the consistency evaluation Equation 2. The results of this assessment are displayed only in the top row of Figure 8. This violin plot presents the consistency metrics

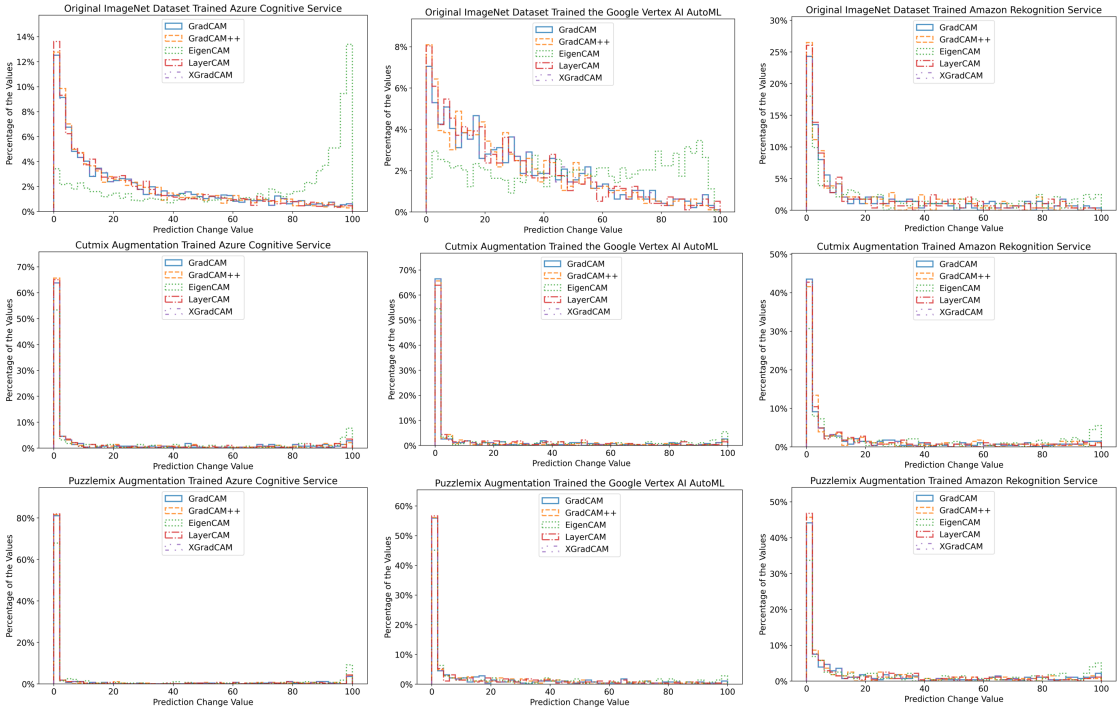


Figure 7: Distribution of Prediction Change Percentages Across Cloud AI Services with Multiple CAM-based XAI Methods

for explanations derived using five CAM-based XAI methods across leading cloud platforms A smaller value on the y-axis indicates better consistency evaluation results. Among these five XAI techniques, both GradCAM++ and LayerCAM show superior consistency metrics. Concurrently, as previously observed, EigenCAM exhibits notably higher values, signifying the worst consistency.

Summary - This scenario demonstrates the practical application of the open API architecture, outlined in Section 3.1.1. The findings demonstrate that XAI methods exhibit variability when subjected to the evaluations. Notably, GradCAM++ and LayerCAM display superior results, recording the lower values of consistency metrics.

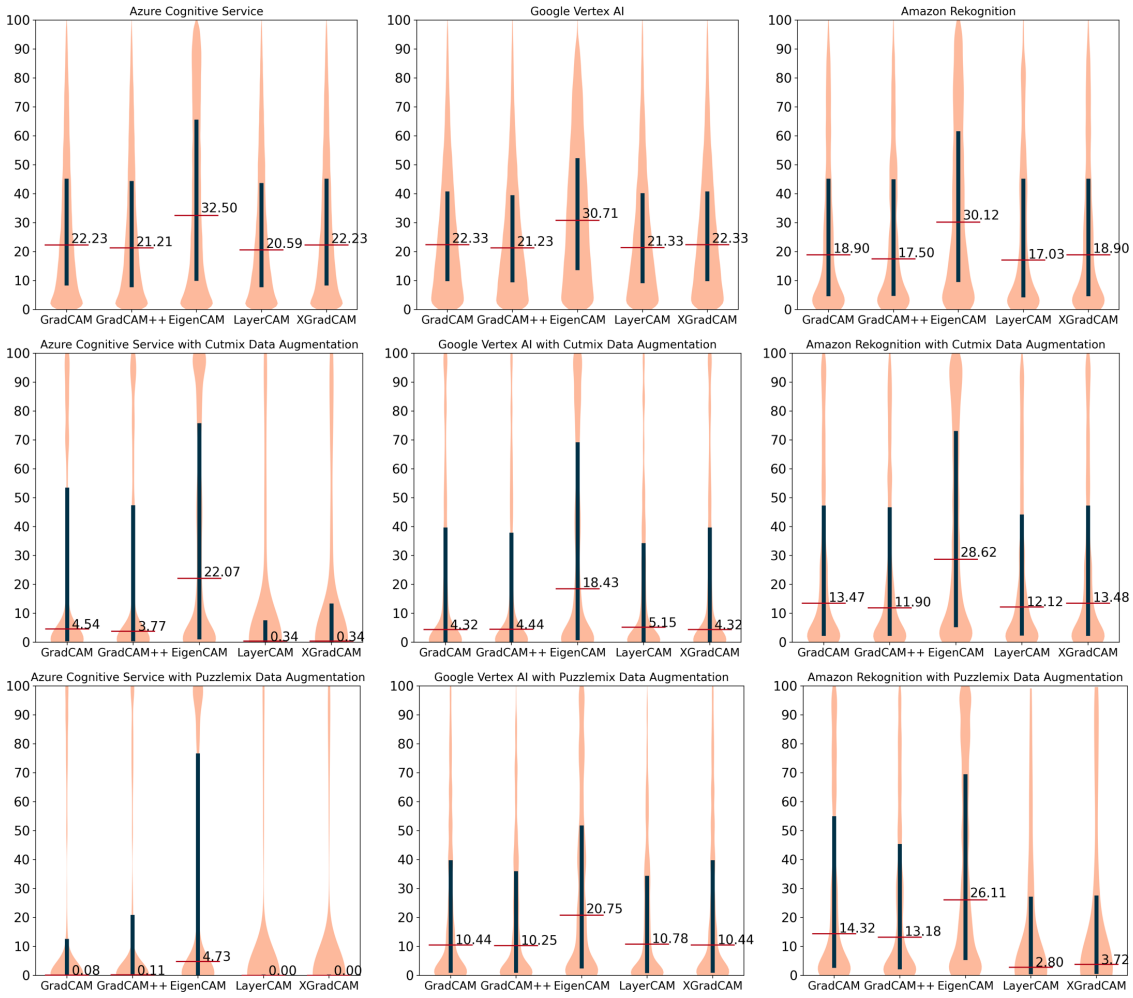


Figure 8: Evaluation of XAI Consistency Across Cloud AI Services Using CAM-based XAI Methods: Lower Mean Values Signify Better Consistency Metric

Scenario Two: Data Augmentation Effects on Cloud AI Service Performance

Cloud service platforms such as Microsoft's Azure Cognitive Services [132], Google's Vertex AI [69], and Amazon's Rekognition [7] offer online AI services, including image classification models that can be trained with user-provided datasets. Specifically, Azure Cognitive Services limits the number of distinct labels to fifty for image classification.

Cloud AI services primarily provide evaluation metrics such as Precision, Recall, and F1-Score. We devise three cases. One case uses the original ImageNet dataset without data augmentation, while the other two apply data augmentation to the ImageNet dataset. Cloud AI services divide the augmented data into training and test sets. These services remain as black boxes.

Table 2: Performance Metrics for Cloud-Based AI Services with Different Data Augmentation Strategies

	Azure			**Google**			**AWS**		
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
Original	**96.0%**	74.6%	0.839	90.5%	41.1%	0.565	82.8%	78.7%	0.807
Cutmix	88.3%	**87.0%**	**0.864**	**94.1%**	**82.0%**	**0.876**	**84.7%**	**82.5%**	**0.818**
Puzzlemix	**92.6%**	**91.0%**	**0.905**	**93.5%**	**81.1%**	**0.869**	**85.3%**	**84.0%**	**0.828**

This scenario is to evaluate the effect of the learning performance of these cloud AI services with data augmentation. We analyze the Precision, Recall, and F1-score metrics, as detailed in Table 2. In particular, data augmentation only reduces precision in Azure AI service since the precision is already high without data augmentation. However, in all the other cases, both CutMix [219] and PuzzleMix [97] augmentation algorithms improve the Precision, Recall, and F1-score. The data augmentation improves the overall cloud model performance F1 score by 0.11 and 0.13.

Summary - The use of data augmentation without the expansion of the training dataset reduces the rate of false negatives and improves the F1-score across three cloud AI services on the ImageNet dataset.

Scenario Three: Data Augmentation Effects on Explanation

Data augmentation is a tool for boosting the performance of deep learning models by generating a diverse set of synthetic data from existing samples. To further investigate the performance of XAI methods in more complex scenarios, two more advanced data augmentation methods, CutMix [219] and PuzzleMix [97], are introduced and tested in this experiment. CutMix and PuzzleMix blend images in a patch-wise manner and rearrange patches from multiple images, respectively. They have been demonstrated to enhance model performance significantly [97]. However, the impact of these augmentation methods on the interpretability of models and the effectiveness of XAI methods remains unexplored.

We commence our exploration by applying the advanced data augmentation techniques, CutMix and PuzzleMix, to our test dataset. The effect of these augmentations on the AI model's performance is summarized in Table 2. Upon evaluation of the three cloud AI services, we observe a general improvement in Precision, Recall, and F1-score metrics under the application of CutMix and PuzzleMix, echoing the effectiveness of these methods in enhancing model performance as reported in the literature [97].

The histogram in Figure 7 is the visual representation of prediction changes of the test data set for the three cloud AI services. We divide samples into fifty equal bins in the one thousand images test data set to observe their distribution. The histogram, Figure 7, illustrates the XAI prediction change distribution. The middle and bottom rows show that both CutMix and PuzzleMix data augmentation techniques significantly enhance the performance of the model compared to the original dataset. The chart shows that the prediction change values for both of these augmentation techniques are smaller than the original dataset, which implies that the XAI methods can generate explanations that are more aligned with the original data.

This observation supports the claim that these augmentation techniques not only enhance the model's performance but also improve the explainability of the model. It is

particularly important in practical applications where both the accuracy of the prediction and model explainability are crucial.

Moreover, the consistency of the XAI methods under these augmentation techniques, as indicated by Figure 8, further supports this observation. Consistency here refers to the stability of the XAI methods in providing explanations for more data samples. From the figure, it can be seen that the XAI methods tend to have better consistency with CutMix and PuzzleMix augmentation compared to the original dataset. This result discovered that these augmentation techniques could potentially enhance XAI results.

Combined with the results in Table 2. An analysis of traditional performance metrics such as precision, recall, and F1 score might suggest a minimal effect of the CutMix and PuzzleMix data augmentation techniques on model performance. However, this observation could potentially underestimate the broader influence of these augmentation techniques. When examining the outputs of the XAI methods, these augmentation techniques demonstrate a pronounced enhancement.

Furthermore, through the observation of the histogram Figure 7, we notice a marked improvement in the performance of the EigenCAM method after the application of data augmentation. Initially, EigenCAM seemed to provide less consistent explanations for the original model, as evidenced by the large prediction change values and bad consistency. This suggested a significant discrepancy between the identified salient features and the model's actual decision-making focus.

However, upon applying the data augmentation techniques, similar to other applied XAI methods, EigenCAM demonstrated a marked increase in consistency and accuracy of the provided explanations. This observation underlines the potential of data augmentation techniques to significantly enhance the performance of XAI methods that initially appear suboptimal. This discovery highlights the need to consider data augmentation as a valuable tool not only for enhancing model performance but also for optimizing the explainability

provided by XAI techniques.

The usage of XAI, as illustrated in Figures 7 and 8, unveils an alternative dimension to this assessment. Through the lens of XAI, a considerable enhancement in model explainability is discernible with the data augmentation techniques. This finding makes it clear that even though data augmentation techniques might not significantly shift conventional performance metrics, they notably enhance a model's explainability. This insight enriches our understanding of the model's decision-making process.

Our discovery highlights the importance of comprehensive evaluation strategies. Relying solely on traditional performance metrics might overlook these techniques' crucial benefits. In Figure 8. The violin plot's highlight area represents the frequency distribution of defined distance values, and the vertical bars represent the range of the distance value from the first to the third quartile. The median values are marked and indicated.

Among the specific XAI methods, the Layer-CAM method shows the best consistency since it has the smallest medium and mean values, followed by Grad-CAM++, Grad-CAM, and XGrad-CAM, which are slightly less stable but still comparable with LayerCAM. Eigen-CAM shows poor consistency, and its overall prediction changes value distribution is more elevated than the other XAI methods.

Summary - The scenario concludes that data augmentation techniques can enhance explanations from cloud AI services without altering the model's structure. XAI consistency values improve by 12.79 and 14.10 respectively.

Scenario Four: Reproduction of XAI Explanation

From the scenarios described above, we have experienced the complexity of XAI evaluation. XAI evaluation extends beyond merely running a single XAI algorithm. It involves a complex composition of entities and tasks, including datasets, feature masking and mutation, data augmentation, benchmark model approximation, cloud AI services, and various

XAI methods. The scale of these combinations can rapidly escalate the complexity involved in providing explanations. Tracing XAI pipelines and ensuring transparency in XAI explanations are key technical requirements for XAI evaluations. We devise four cases to demonstrate the XAI pipeline provenance in our framework design aspects. Figure 9 illustrates the provenance graph of XAI variation using ResNet, Azure Cognitive Service, and Grad-CAM XAI method. The metadata defined in section 3.1.1 become the vertices in the graph. Each edge in the provenance graph is an API-based service invocation. A red colored vertex indicates a certain variation introduced to the configuration of a pipeline. The blue vertices indicate related vertices in the paths that are affected by the change. A variation results in a new pipeline graph that represents the adjustments.

Cross-dataset adaptability. Database microservice is designed to serve data via a RESTful API, integrating with cloud data storage services, in the case study, Azure Blob Storage. It ensures the data required by XAI are organized by group and ready for retrieval. By altering the configuration parameters, users can invoke alternative cloud databases, enhancing the system's flexibility. Case one shows a reference pipeline configuration. The dataset can be replaced without changing other configurations in a pipeline.

Configurable model integration Case two illustrates the necessity for users to replace the model currently in use, exemplified by replacing the ResNet model with DenseNet. As indicated by a red node in the graph data, users have the flexibility to configure the model. This modification necessitates the re-execution of the entire pipeline, resulting in changes to the content associated with the blue nodes. The graph provenance data captures and documents user interactions and the resultant outcomes, providing a detailed account of the effects of modifications.

Incorporation of data augmentation techniques. Datasets are labelled with distinct names and group identifiers. This feature is used to generate augmented datasets by specifying the original dataset and the augmentation method. The resulting datasets can then

be employed to retrain AI models, providing enhanced model performance. Case three in Figure 9 shows the pipeline's variation with data augmentation applied. Compared to case one and case two, the vertex *CutMix Augmentation* is connected to a new dataset generated.

Extensibility to XAI methods. For attribute-based XAI methods, the explanation varies by XAI methods for the same dataset and the same model [202]. With well-structured API interfaces, new XAI methods can be integrated by encapsulating the input and output conformable to open API specifications. Case four adopts a different XAI method *Layer-CAM* in the pipeline compared to Case Three.

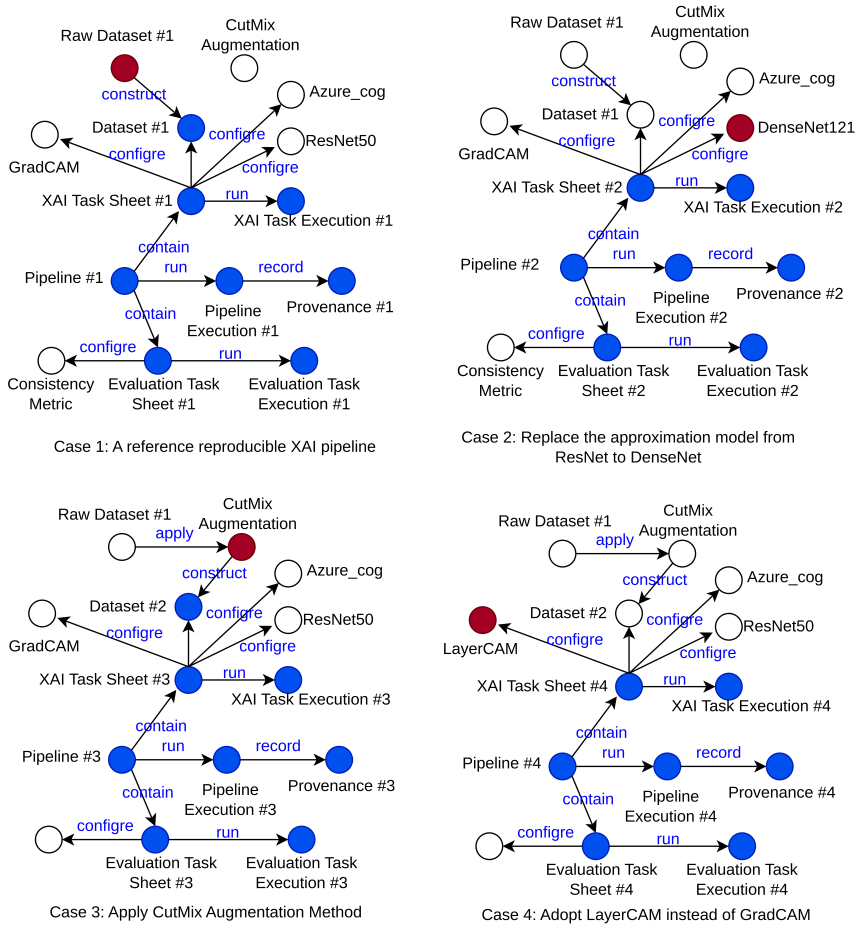


Figure 9: Provenance Graph by XAI Service. Operations using ResNet and Azure Cognitive Service with Grad-CAM, and explanation evaluation

XAI task reproducibility offers significant benefits in the domain of services. First, it

increases the transparency of the service by offering a clear understanding of the steps leading to a specific result. The heightened transparency, in turn, builds user trust as they can independently verify the results. Secondly, the service's adaptability is enhanced. Users can adjust models or configurations based on previously successful operations. Thirdly, reproducibility facilitates auditing, which can lead to further improvements in service reliability and quality. Provenance data is persistently stored in the cloud-based, user-configured database. This ensures that the user-owned provenance data accurately represents the XAI service.

Summary - The XAI service is composition of diverse cloud databases, data augmentation, and XAI methods. The provenance of XAI pipelines enables reproducible XAI processes by re-running the configuration of the graph-format provenance data. Any changes are tractable from the pipeline's configuration which is an encapsulation at the microservice level.

The Discover Scenarios Results

In conclusion, the scenarios we explored manifest the effectiveness and versatility of our proposed XAI microservice architecture. This structure is instrumental in facilitating the evaluation and comparison of various XAI techniques and cloud-based AI services. We summarize the observations:

- Both CutMix and PuzzleMix data augmentation techniques have been found to significantly improve the performance metrics (Precision, Recall, and F1-score) of cloud-based AI services, as demonstrated in Table 2.
- The augmentation techniques have been shown to reduce prediction change values, representing an improvement of XAI performance, as depicted in Figure 7.

- Traditional performance metrics may understate the broader benefits of data augmentation techniques. By considering XAI methods in evaluation, we reveal an enhanced explainability of AI models, indicating the augmentation techniques' potential to create more trustworthy AI systems.

- There is an observable improvement in the consistency of XAI methods under the CutMix and PuzzleMix data augmentation techniques, as indicated by Figure 8. This underscores the potential of these techniques in enhancing XAI results consistency and improving model performance.

- Among the selected XAI methods, the consistency of Layer-CAM outperforms other XAI methods when applied to Azure Cognitive Service using ResNet, as demonstrated in Figure 8. Conversely, Eigen-CAM exhibits poor consistency and a higher distribution of prediction change values, indicating lesser reliability.

These insights reinforce the importance of incorporating XAI methods in the evaluation process to understand the impact of data augmentation techniques, thereby broadening our perspective on AI model performance.

3.1.3 Evaluation of XAI Service Performance

Upon completion of the experiments, we delve further into the evaluation of our XAI service, examining implementation complexity, computational resource usage, and the reproducibility of results. These elements play a crucial role in determining the effectiveness of any service system.

The Effort Required for Microservice Development

To evaluate the effort involved in developing the added microservices, we quantified the Lines of Code (LoC) needed for integration. Specifically, we focus on the LoC necessary

to incorporate a given model or algorithm into a microservice architecture using a provided template. Table 3 presents the statistical breakdown of this effort.

Table 3: Quantitative Analysis of LoC for Algorithm Integration into Different Microservices

	The Template	**Azure Cognitive Service**	**Google Vertex AI AutoML**	**Amazon Rekognition**
LoC	76	144	198	148

To perform the migration of pre-existing code into the microservice architecture, the development process involves several key steps:

1. Initialize an HTTP server framework as the microservice's base.
2. Incorporate the pre-existing code into this server framework as a modular function.
3. Standardize the input: Convert HTTP request data into a format suitable for the model or function.
4. Standardize the output: Transform the model or function output into a proper HTTP response and, if necessary, update the database with the result.

By outlining the development process in this way, we provide a roadmap for developers who seek to transition from monolithic or less modular architectures into a microservices framework.

Integration of Methods from Other XAI Frameworks

We perform a comparative analysis to showcase the external XAI tool's integration capabilities within our XAI service, with a focus on incorporating the GradCAM [168] method. In Figure 10, the comparison utilizes ten thousand images from the ImageNet dataset [43]

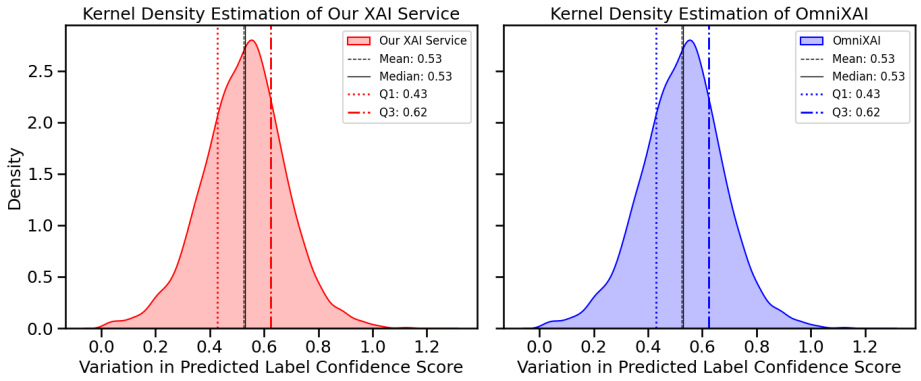


Figure 10: Kernel Density Estimation of prediction confidence changes for 'Our XAI Service' vs. 'OmniXAI'.

input to the ResNet model [79]. The Kernel Density Estimation (KDE) plots show a consistent prediction confidence distribution between our service and the standalone algorithm from OmniXAI. Our XAI service is effective in streamlining complex XAI methods via automated pipelines, simplifying the user experience with the same algorithm executed. Our service, by offering integration and RESTful API, makes advanced XAI techniques accessible for explaining model predictions with less operations.

Evaluation of Computational Resources

The concept of computational overhead extends to the resources required for computation on both the server and client sides. Cloud-based AI services abstract away resource management complexities, charging users based on API request count or usage duration. To provide a comprehensive analysis, we measured computational resources utilized by different XAI methods. This is particularly pertinent for applications that demand real-time responsiveness and for those processing large datasets. We maintained a consistent infrastructure across all XAI methods to ensure a fair comparison. Data related to microservice response time and computational resource utilization were collected and analyzed. Through code audits and local tests, we found that resource usage scales linearly with the workload. Table 4 presents a comparative view of the average energy utilization for different XAI

methods and their corresponding microservices on a per-data-sample basis.

Table 4: Comparative Energy Consumption of Microservices (Average per Data Sample, in 10^{-6} kWh)

Microservices	***CPU Energy***	***GPU Energy***	***RAM Energy***	***Energy Consumed***
Data process	3.60	N/A	0.08	3.68
Resnet	1.61	3.77	0.02	5.39
Densenet	0.38	0.89	0.02	1.28
GradCAM	8.32	19.50	0.10	27.91
GradCAM++	7.09	15.93	0.09	23.11
EigenCAM	121.48	107.54	1.71	230.74
LayerCAM	7.21	10.08	0.09	17.37
XGradCAM	6.77	17.35	0.08	24.20
Evaluation	16.32	N/A	0.17	16.49

From Table 4, we observe that Layer-CAM has the lowest computational overhead. This suggests that Layer-CAM might be better suited for systems with limited compute resource availability. On the other hand, Eigen-CAM shows the highest computational resource utilization and execution time, in line with its lower performance in the prediction change and stability tests.

Overall, these results imply the cost for the quality of the explanations, computational efficiency, and time efficiency. This indicates that the choice of the XAI method should consider the specific requirements and constraints of the application.

3.1.4 Summary

The growth of cloud-based AI brings the urgent need for explanation in these services. Our study responds to this necessity by introducing an XAI service architecture. Utilizing a microservices design, we facilitate the flexible deployment and evaluation of XAI methods for both pre-trained and cloud-based AI models. We present a workflow that retrieves

and evaluates feature contribution explanations for cloud model service. We rigorously evaluated our service in computer vision scenarios. Importantly, the incorporation of data augmentation techniques such as CutMix and PuzzleMix not only increases the performance of the underlying models but also enhances the quality of these explanations. Our detailed assessment of computational efficiency highlights the architecture's broad applicability across different scenarios and cloud platforms. A key feature of our service is its provenance data design, making all XAI operations traceable and reproducible. This work provides a concrete framework that provides automated XAI to cloud AI services. The principles and design of our architecture offer a reference for future developments in the adoption of XAI operations.

3.2 XAIport - A Service Framework for the Early Adoption of XAI in AI Model Development

In this section, we propose the early adoption of Explainable AI (XAI) with a focus on three properties: Quality of explanation, the explanation summaries should be consistent across multiple XAI methods; Architectural Compatibility, for effective integration in XAI, the architecture styles of both the XAI methods and the models to be explained must be compatible with the framework; Configurable operations, XAI explanations are operable, akin to machine learning operations. Thus, an explanation for AI models should be reproducible and tractable to be trustworthy. We present XAIport, a framework of XAI microservices encapsulated into Open APIs to deliver early explanations as observation for learning model quality assurance. XAIport enables configurable XAI operations along with machine learning development. We quantify the operational costs of incorporating XAI with three cloud computer vision services on Microsoft Azure Cognitive Services, Google Cloud Vertex AI, and Amazon Rekognition. Our findings show comparable operational costs between

XAI and traditional machine learning, with XAIport significantly improving both cloud AI model performance and explanation stability.

Machine Learning Operations (MLOps) is a multidisciplinary approach that includes a set of best practices, concepts, and developments [100]. The major tasks of MLOps include automating the ML lifecycle, such as model development, validation, quality assurance, deployment, monitoring, and governance [186].

The quality assurance of MLOps involves several key components, such as data validation, feature engineering assessment, model training evaluation, cross-validation, performance metrics analysis, fairness, bias evaluation, and model explainability [180]. The explainability of models is necessary in sensitive domains. The lack of model explanation leads to distrust in the AI models [83, 170].

Post-hoc XAI methods provide explanations for complex, already-trained models. The Post-hoc XAI techniques, such as SHAP [126], provide feature attribution as explanations. XAI operations are often considered a post-hoc activity [159], implemented after the model has been trained and verified. The quality assurance [180] of complex software development has shown that incremental development and iterative quality control are efficient and cost-effective. Inspired by this principle, we argue that early adoption of XAI operations enhances the quality assurance of AI models with probing observations at the feature representation level and summarized explanations across datasets and AI models.

We present `XAIport`, an XAI service architecture that allows XAI early adoption across cloud platforms and offers unified open API access. Inconsistent explanations can be misleading to evaluate the AI models. The probing results and derived explanations should be quantified for their stability and consistency across datasets and AI models. The efficiency of applying XAI operations should be measured quantitatively so that the runtime overhead and cost are well balanced in evaluating the benefits of adopting XAIs. We summarize the key considerations for the early adoption of XAI operations as follows:

- **Quality of Explanation.** Explanations generated by XAI methods should adhere to the evaluation metrics, specifically the explanation consistency metrics, defined in an established XAI process [85].

- **Architecture Compatibility.** The XAI service flexibly integrates the AI models and XAI methods within the microservice architecture. This ensures incorporation into existing cloud services via open APIs.

- **Cost-Efficiency in CI/CD.** The adoption of XAI into MLOps should result in proportional cost-efficient operational overhead during the Continuous Integration and Continuous Deployment (CI/CD) phases. In ideal scenarios, the additional complexity XAI introduces is approximately proportional to existing MLOps.

This approach mirrors best practices in software engineering, where early integration of unit tests and quality assurance solidifies the software. Additionally, `XAIport` provides a unified measurement of resource consumption and XAI operation overhead.

3.2.1 XAI Methods and Evaluation in the Practice

We explore the increasing importance of XAI in AI domains, such as healthcare and finance. Subsequently, we review the MLOps workflow. Then, we review XAI methods in the field of computer vision and metrics for the assessment.

Alongside the complexity of AI models, the need for explainability has concurrently risen [2]. Explainability fosters a better understanding of model behavior, facilitates trust and encourages responsible AI usage [13]. In the critical sectors of healthcare, finance, and legal systems, XAI is essential in comprehending the model's decisions and implications and ensuring compliance with legal and ethical protocols [13, 146, 150]. The healthcare domain witnesses an especially pronounced need for XAI due to the growing reliance on AI technologies [109].

MLOps integrate machine learning and operations, emphasizing the importance of explainability or XAI [100]. The process begins with defining system requirements [77]. During data collection, potential biases in the data are often overlooked [208]. Data preprocessing techniques, such as cutmix [219] and puzzlemix [97], aim to improve dataset quality. Feature engineering is central to ML models, and incorporating explainability during feature selection simplifies the model [18, 220]. Traditional model quality assurance metrics are expanded to include explainability, especially in cloud AI services [132, 232]. Consistency evaluations in XAI ensure trustworthy explanations [85]. Deployment in MLOps emphasizes the use of visualization tools for better model understanding [85].

Class Activation Mapping (CAM) [228] emerged as a pioneering approach leveraging the global average pooling layer to localize features within Convolutional Neural Network models. A limitation, however, was its need to adjust the model's fully connected layer. In contrast, Grad-CAM [168] refined this by determining the localization weight through the layer's average gradient, eliminating the need for replacements. Advancing this further, Grad-CAM++ [30] incorporated second-order gradients for enhanced precision. Eigen-CAM [140] uniquely uses the primary component of activations without class-specific considerations. LayerCAM [92] assigns spatial weights to activations considering only positive gradients, while XGrad-CAM [62] adjusts gradients based on normalized activations. Representing the forefront of XAI methodologies, these techniques have proven their prowess in generating saliency maps for visual-based XAI tasks.

The metrics for evaluating explainability in XAI are essential and gain considerable attention [35]. However, the field is still grappling with several challenges. Vilone et al. [192] list scientific papers for approaches to evaluate the XAI method and point out the lack of consensus in defining unified evaluation metrics. Instead of qualitative metrics, we prefer quantitative metrics to assess XAI methods concretely. A systematic assessment [85] provides clear consistency metrics to XAI feature contributions.

Summary - The XAI operations have been particularly explored in domains such as healthcare and finance. As AI services become available through pre-trained models bundled with elastic cloud computing resources, the operations of XAI in such a context still require thorough architectural-level research.

3.2.2 The Context of Early Adoption of XAI Services

The goal of XAI adoption is compatible with the objectives of model quality assurance. We design the XAI operations function as a probe into AI models with or without the model's intrinsic structure to provide explanations, for instance, on how features may affect the learning results. Hence, we propose the early adoption of XAI operations revolves around three major `core components`: (1) definition of augmented quality assurance metrics for explanation stability and consistency; (2) compatible architecture styles to integrate with cloud AI service development and deployment; and (3) XAI operations are configurable and measurable in the same manner as cloud AI services. In addition, the adoption of XAI should be cloud-independent and allow cross-validation of multiple XAI methods, AI models, and datasets.

Augmented Metrics for Explanation Quality Assurance

Several studies [35, 115, 192] have shown that XAI methods do not always offer consistent explanations, especially in experiments involving Post-hoc XAI methods. Beyond the model performance, the adoption of XAI operations should measure consistency to ensure model explainability quality. XAI consistency metrics [85] comprise both `Explanation Stability` and `Explanation Consistency`.

`Explanation Stability` measures the consistency among explanations from many data samples. To compute this metric, we consider all possible pairs of prediction changes

from data samples. We then average all these values to a metric, f_d^K, representing stability. Theoretically, a smaller value of this metric indicates higher consistency among the explanations generated by the XAI method. We employ the `Explanation Stability` in the pilot evaluation.

`Explanation Consistency` measures the consistency between different XAI methods. We calculate the prediction changes among XAI methods and average all to f_d^X. A smaller value indicates that different XAI methods are producing similar explanations.

Compatible Architecture Styles

The adoption of XAI operations should function in the compatible architecture context of MLOps. As pre-trained ML models on the cloud are available, cloud services become the encapsulation of the models running on elastic computing resources. The communication between XAI methods and AI models thus follows the service orientation. We propose `XAIport`, a service architecture in which the core XAI components are each represented as a microservice with the Open API definition. The APIs for the `XAIport` service are organized according to Open API 3.0 [183] standards and documents on SwaggerHub.

The core architecture includes: `(1) Coordination Center` uses a configuration template to specify pipelines of end-to-end XAI operations from data input, to feature variation, to model inference, to feature contribution explanation, and to evaluation generation; `(2) Data Processing and Storage` is responsible for data preparation and storage intermediate results for explanation generation; `(3) XAI Microservices` encapsulate state-of-the-art XAI methods computing the feature contribution explanation for AI models on a certain dataset; `(4) Evaluation Microservices` computes and visualizes the metrics for XAI explanation. These services produce the answers to questions such as *how is an AI model affected by feature representation?* Figure 11 provides an illustration of integrating XAI operations along with the development of AI models using cloud

AI. We assume the development of AI models adopts the best practices and technology supports from MLOps and DevOps [50, 83, 100, 186]. The XAI operations are encapsulated as microservices and deployed on the cloud as well. The communication between XAI operations and cloud AI models is only through the endpoints defined by Open APIs. Thus, the enhanced explanation metrics from XAI provide extra measurements for AI model quality assurance.

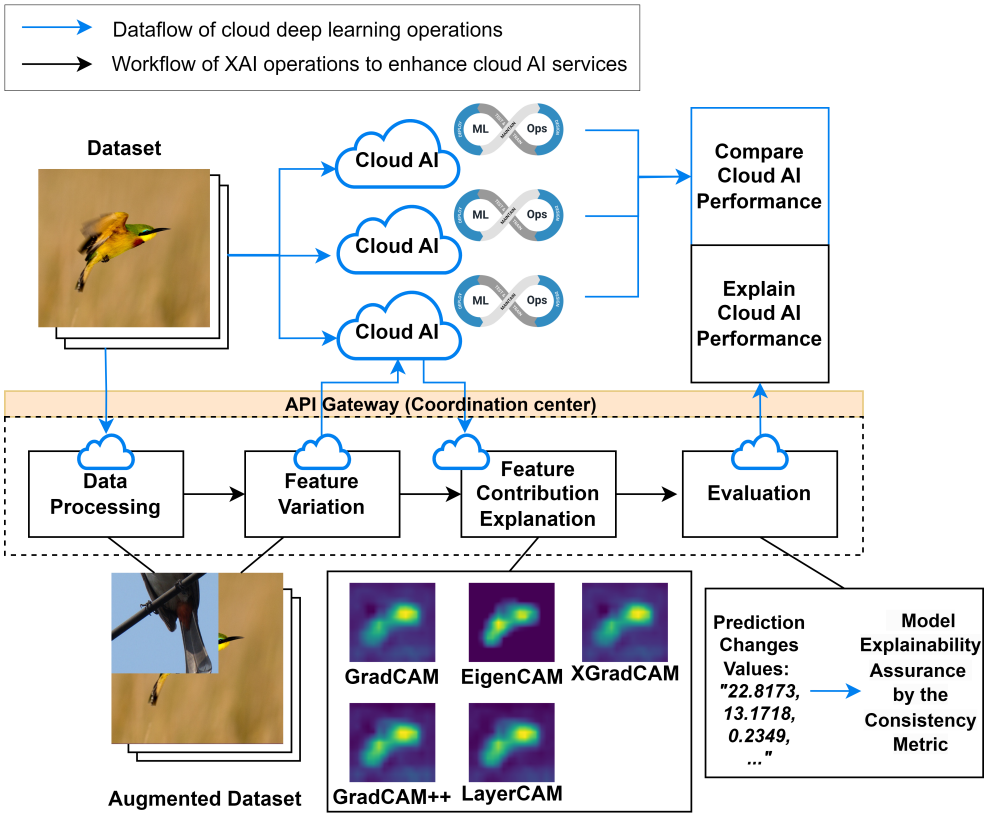


Figure 11: An Illustrating Scenario of Adopting XAI to Multiple Computer Vision Cloud AI Model Development.

Integration with Open Community Pretrained Models. The service-oriented open API architecture `XAIport` is extensible to the AI model development based on open community libraries such as Hugging face [209]. First, the pre-trained models are accessible for trial and testing with open APIs in the same communication model as the cloud AI services

in Figure 11. In addition, when the pre-trained models are downloaded for further retraining and fine-tuning on a domain-specific dataset, the model is deployed in a containerized virtual machine that can be run on a cloud or on a proprietary data center. If we assume such a model is further encapsulated with Open API access, then the `XAIport` architecture illustrated in Figure 11 is applicable without any further changes since `XAIport` decouples XAI operations and AI models through only the endpoint communication through Open APIs.

Extension to Support A/B Testing of AI Services. Performing A/B testing on ML models has been adopted by real-world services [110]. The endpoint of an AI model in the form of Open APIs is the operation unit for automated deployment with multiple production variants for A/B testing. As illustrated in Figure 11, the XAI services with Open API as endpoints are capable of either covering a certain variant (such as the data augmentation) or communicating with these A/B testing variant endpoints and derive the explanation to link the model's learning performance and feature contributions.

Configurable and Measurable XAI Operations

Current XAI methods are in the form of algorithms and disparate library code [2, 13, 85, 126]. We propose the XAI operations should be configurable and measurable. XAI methods, particularly post-hoc techniques, require additional processing power and computational time, thereby increasing the computational overhead [159]. Hardware-wise, the CPU, memory, and GPU requirements may rise to accommodate additional XAI processes. We target to methodically evaluate the operational overhead incurred in the integration and deployment of XAI services across multifarious cloud providers. We focus on measuring the XAI service deployment time and complexity. This exercise entails multiple steps:

Selection of CI/CD Tool. Our methodology commences with the use of cloud pipeline and build tools as the designated continuous integration and continuous deployment (CI/CD)

pipeline tool. For instance, Amazon Web Services CodeBuild furnishes essential building facilities for containerized applications, which is imperative for orchestrating a cloud-agnostic XAI milieu, ensuring uniform deployment across diverse cloud platforms.

Measuring XAIport Computational Overhead. We evaluate the computational overhead of diverse AI models and XAI techniques using `CodeCarbon` [106]. This tool, previously applied to several projects [157,187], is incorporated into `XAIport` to track time, energy, and carbon footprints during XAI activities and AI predictions. Using `CodeCarbon` [106], we differentiate the energy efficiency and time consumption of various XAI operations, guiding the selection of the optimal method for specific use cases.

Measuring XAI Service Deployment Overhead. We assess the effort needed to deploy `XAIport` on multiple cloud providers. We perform Amazon Web Services Elastic Container Service (ECS), Azure Virtual Machines, Azure Container Instances, and Google Kubernetes Engine for their efficiency in deploying AI container applications.

3.2.3 A Pilot Evaluation

We conduct a pilot study using `XAIport` to explore the answer to a data-driven question for cloud-based AI services as follows.

> Can early adoption of XAI improve the learning performance of cloud computer vision services? If any, can the improvement be explained? Can the explanation result be evaluated?

This study uses five visual explanation algorithms, shown in Table 5, applied to three image classification computer vision services, which are Microsoft Azure Cognitive Services [132], Google Cloud Vertex AI [69], and Amazon Rekognition [7]. Cloud platforms offer the following advantages. First, they automate deployment with resource allocation. Second, they offer built-in scalability to manage computational demands efficiently. Third, they deliver monitoring tools for basic model performance. However, These platforms

overlook the explainability of models. In this case, we adopt the `XAIport` service framework. Upon evaluating the three cloud AI services, we identify potential areas for further optimization in both model performance and explanation stability. Then, with the integration of Cutmix [219] and Puzzlemix [97] data augmentation techniques in the XAI operation, we enhance both the cloud model performance and explanation stability on these platforms.

Improving Cloud AI Explanation Metrics with Early XAI Adoption

We apply the ImageNet dataset [43] via the `XAIport` APIs to explore the data-driven question on AI model development. The baseline is the cloud AI service trained only using the original dataset. We adopt five XAI algorithms, which are Grad-CAM [168], Grad-CAM++ [30], EigenCAM [140], LayerCAM [92], and XGrad-CAM [62]. Adoption of these XAI methods takes image data as inputs from the data processing service and generate saliency maps that highlight the focal areas of layers of the model. Then, these data become the inputs for the three cloud AI and return prediction scores. Finally, we use the evaluation service to derive prediction changes and the stability metrics as algorithm 2.

Table 5: Model and XAI Evaluation Results

Service	**F1-score**	**XAI Evaluation**				
		GradCAM	**GradCAM++**	**EigenCAM**	**LayerCAM**	**XGradCAM**
Azure (B)	0.839	22.227	21.211	32.498	20.595	22.229
Google (B)	0.565	22.329	21.233	30.713	21.328	22.327
Amazon (B)	0.807	18.900	17.505	30.119	17.027	18.900
Azure (C)	0.864	4.544	3.773	22.072	0.339	0.339
Google (C)	0.876	4.316	4.437	18.427	5.147	4.316
Amazon (C)	0.818	13.474	11.901	28.623	12.120	13.475
Azure (P)	0.905	0.078	0.107	4.732	0.002	0.002
Google (P)	0.869	10.440	10.246	20.754	10.781	10.440
Amazon (P)	0.828	14.316	13.179	26.105	2.797	3.724

Note: "B" stands for baseline without augmentation, "C" stands for "Cutmix" and "P" stands for "Puzzlemix". The values in the table are XAI stability metrics [85]. (The smaller, the better explanation stability.)

Results and Discussion. Table 5 shows the detailed measurement results. The F1-score

shows a subtle enhancement when both CutMix [219] and PuzzleMix [97] techniques are employed. A pronounced improvement is observed in the model's explanation evaluation. The consensus of XAI explanations has alignment with the performance of cloud AI services. Such a consistent explanation result provides a trustworthy view of the contribution of the data-driven technique to the model performance improvement.

These cloud AI services are entirely black-box. There is a lack of access to the model's parameters, the internal network structure, fine-tuned loss functions, and so on. The adoption of XAI through service orientation and open APIs has enabled us to probe the performance and obtain explanations.

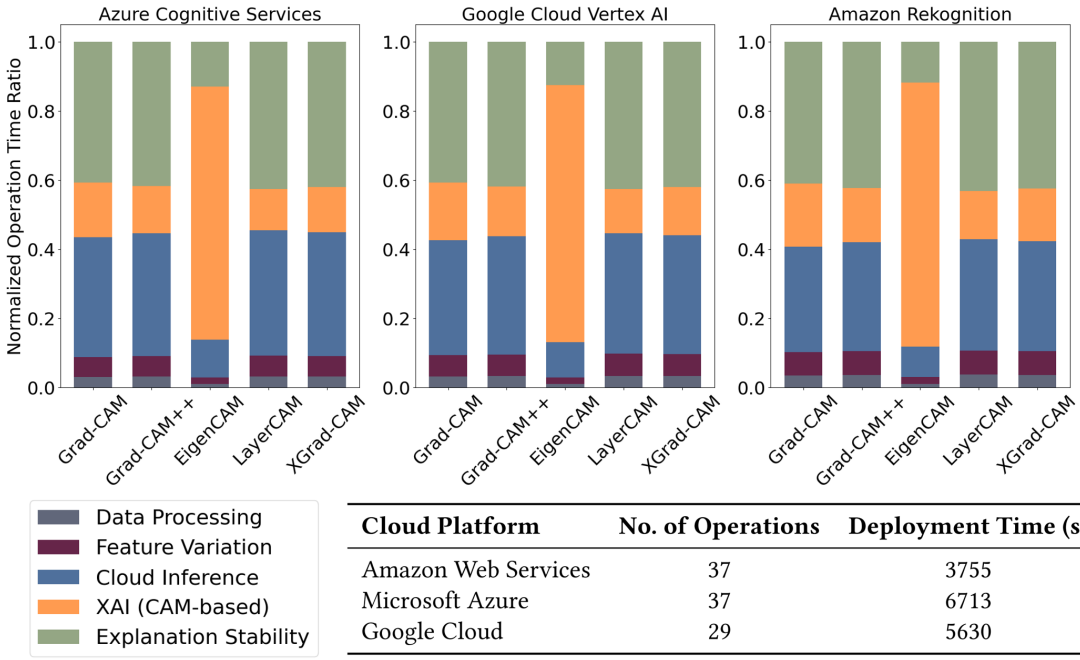
Computational Analysis of XAI Operations and Deployment Across Cloud Services

We record and decompose the time spent on (1) Data Processing, (2) Feature Variation, (3) Cloud Inference, (4) XAI and (5) Explanation Stability. We analyze the operation consumption across the three cloud AI services and the five CAM-based XAI methods.

Figure 12 shows the composition of each unit in XAI operations on average per data sample and the framework deployment duration. The model inference time is relatively stable across cloud services. However, XAI methods take different demands and there is a need for optimization in the evaluation metrics.

3.2.4 Summary

This work outlines the early adoption of XAI operations in the practices of AI model quality assurance. We define the adoption context in three aspects with mature development methods and technology supports. We illustrate a pilot study on adopting XAI operations to answer a data-driven question with regard to improving three cloud AI services. We demonstrate consistent explanation results with measurements of the computation and deployment overhead. We advocate such a context of practice to broad open AI models'



Cloud Platform	No. of Operations	Deployment Time (s)
Amazon Web Services	37	3755
Microsoft Azure	37	6713
Google Cloud	29	5630

Note: The chart displays the decomposition of average XAI execution time per data sample, derived from 1,000 experiments: Data Processing (0.12s $\pm$ 0.03s), Feature Variation (0.23s $\pm$ 0.06s), Cloud Inference (Azure 1.39s $\pm$ 0.42s, Google 1.26s $\pm$ 0.48s, Amazon 1.06s $\pm$ 0.28s), XAI methods: GradCAM (0.63s $\pm$ 0.12s), GradCAM++ (0.53s $\pm$ 0.08s), EigenCAM (9.19s $\pm$ 3.38s), LayerCAM (0.46s $\pm$ 0.06s), XgradCAM (0.51s $\pm$ 0.11s), Explanation Stability (1.60s $\pm$ 0.56s).

Figure 12: XAI Operations and Framework Deployment Time

quality assurance with XAI to gain trustworthiness. In future work, we aim to further develop the software development toolkit (SDK) based on the `XAIport` framework. The SDK provides the tools for automated deployment of XAI operations along the AI service development using open pre-trained models, dynamic A/B testing of AI services, and validation of new XAI methods.

Chapter 4

Automated Orchestration for Evaluating AI Models and Explanations

The growing adoption of AI models in critical software services has intensified the need to address their inherent opacity. To build trust, reliability, and safety, Explainable AI (XAI) techniques are employed to illuminate the relationship between a model's input features and its output predictions. However, the application of XAI introduces its own set of challenges. For instance, different XAI methods, particularly those focused on feature importance, can produce inconsistent explanations for the same model and dataset, necessitating a systematic evaluation of their reliability.

Furthermore, the process of generating explanations is not a single step but a complex workflow involving data preprocessing, model invocation, and results summarization. This complexity is magnified when assessing models against threats like adversarial attacks. Manually managing the numerous combinations of models, datasets, and XAI algorithms is impractical for comprehensive quality assurance.

To address these challenges, we propose an automated, service-oriented framework designed to streamline the evaluation of AI models and their explanations. XAIpipeline encapsulates computing components as microservices and organizes assessment tasks into

automated, parallel pipelines. By integrating diverse open-source XAI libraries and interfacing with cloud AI services, it provides a structured, end-to-end system for enhancing the explainability and quality assurance of AI-based software.

We define concrete metrics to measure the consistency of feature-based explanations across different XAI methods. We present the design and implementation of the framework, which includes a DevOps-style toolkit with APIs, CLIs, and a web portal for user configuration. We demonstrate the framework's utility across various domains—including tabular data, image classification, video action recognition, search ranking, and code vulnerability detection—by assessing key quality attributes such as computational cost, performance, robustness, and explanation consistency. The source code and supplementary materials are available on GitHub.

4.1 A Micro Process for AI Model Explanation

The topic of XAI has drawn increasing attention from AI experts and scientists. XAI focuses on various methods that provide a human understanding of AI models [134–136]. With the explanations, humans can decide if the AI model is trustworthy.

Computer Vision (CV) models are commonly studied in the field of AI. As the depth of the model increases, the vision model performs better but loses its interpretability. The deep learning models could learn biases from the huge data set. Without explanation analysis, people cannot find the model's bias, which cannot make the model trustworthy.

4.1.1 Motivation for the XAI process

XAI-related research is still in the stage of early development. Many XAI methods focus on solving particular questions or attributes of the explainability of a specific type of model. Afterwards, multiple XAI methods may provide diverse explanations for the same cases. It

is hard to judge which explanation is reliable without evaluation With the diversity of XAI methods, the decision to select an XAI method and the subsequent development with the XAI method, including model evaluation, is beyond a single task and involves a complete process.

In this work, we develop an XAI process to provide a general guideline for XAI practitioners. The current development of the XAI process and evaluation metrics mainly focuses on analyzing the XAI methods that generate feature contribution explanations.

To develop the XAI process for feature contribution explanation, we derive three research questions: Given the types of XAI methods and diversities of AI models, what is the basis for the process development to cover the combination? What are the core activities and major entities produced by those activities in a process definition for feature contribution explanation? What metrics for feature contribution explanation are used to evaluate different XAI methods based on the measurement derived for explainability?

The contribution of this work is three-fold as follows:

- We summarize the criterion for feature contribution explanation. We construct a taxonomy for the state-of-the-art XAI methods and guide XAI practitioners in selecting suitable candidate methods for the AI model explanations.

- We present an XAI micro process which provides general guidance to practitioners to achieve AI model explanation, particularly by feature contribution. The process enables practitioners to evaluate and compare their explanation results between multiple approaches. We define two metrics to assess the stability and consistency of the feature contribution explanation methods.

- We present three case studies to demonstrate the usage of the process and the assessment of multiple types of feature contribution explanations across different models. The case studies are a scholarly literature search ranking model, a natural language

processing classification model and an image classification model. We apply the process to the three case studies, get the feature contribution explanations, and follow the evaluation of stability and consistency.

4.1.2 An Analysis of the Current XAI Practice and Criterion

Research on XAI arises as to the growth of AI applications in expanding domains. This section discusses the existing techniques and works regarding the evaluation metrics of XAI methods to understand SOTA research and its relevance to XAI practices.

Methods for Achieving XAI

Generally, there are two main branches of methods for XAI techniques, namely, building an inherent interpretable AI model or explaining the model with post-hoc methods.

Algorithms such as linear regression, logistic regression, and decision tree models are adopted in building an inherent interpretable AI model for a specific domain [159]. With the deep learning models are adopted in many domains, these deep learning models have a paucity of explainability, which motivates the development of post-hoc methods.

Post-hoc methods can be classified into two groups, namely model-specific and model-agnostic. The model-specific methods focus on explaining the specific types of models. A study [228] initially proposed a technique named Class Activation Mapping (CAM) from the global average pooling layer of the convolutional neural networks model. CAM visualizes and highlights the discriminative object parts on any given image to the CNN model. Afterwards, Grad-CAM [168], EigenCAM [140], Grad-CAM++ [30], XGrad-CAM [62], and HiResCAM [48] are a series of methods that solve the drawbacks of the CAM method, optimize the map and provide faithfulness.

The model-agnostic methods explain the model by the model's input and output. They can be applied to any model. Some model-agnostic methods are based on rules or simplified

interpretable models for prediction. The decision set [107] is a framework for independent rules. A model based on rules and bayesian analysis [114] aims to build interpretable predictive models.

Alternatively, researchers are focusing on explaining which feature affects the prediction. Anchor [155] computes the individual explanations with high-precision rules. The partial dependence plot (PDP) [61] provides observations on how the outcome prediction changes with the variation of a single input feature in the scope of the entire data set. The computation complexities of PDP are significantly high because all the data set samples to involve in the calculation for each example. In addition, the mutual offset between related features makes PDP unreliable [9]. An optimized approach of PDP named accumulated local effects (ALE) plot [9] was proposed. It eliminates the unreliable bias cases in the PDP method and reduces the computation complexity, making it possible to apply to large data sets. These methods explain the model by varying the input feature and observing the output prediction results. We call them *mutation-based* methods.

Shapley Value [179] is a reliable feature attribution method with a solid theoretical background [169], which explains a single data sample by calculating the contribution value for each feature. Another novel and well-known method, SHAP [126] gives an individual explanation by predicting the contribution values instead of calculating them. These methods calculate the feature contribution values by removing or masking the features. This work summarizes them as *masking-based* methods.

Assessment of XAI Method

The metrics to assess XAI methods are diverse. A study [103] introduces the definition of *soundness* and *completeness* as XAI metrics. Soundness indicates if the explanation is correct. The measurement of soundness requires access to the ground truth. However, the ground truth of the model may not always be available. Then, completeness describes

the explanation of the entire task model. Completeness is an assessment metric for global explanations but is not necessary for local explanations. A study [153] about an explainable book search system evaluates the *trustworthiness* in terms of the retrieval performance. The system utilizes three approaches: the ranking by user clicks, questionnaires, and the user eye tracker system. The ranking results are compared and evaluated for the trustworthiness of explanations.

XAI practitioners should conduct user studies through interviews or questionnaires to assess those metrics. In our study, we present computational metrics to measure the performance of the XAI method.

The XAI Criterion: Input/Output Relation

To develop an XAI process, we need to establish the criteria to evaluate the process outcome. The criterion is discovering the logic between a model's inputs and outputs. It matches the following XAI goals, including:

Feature Influence. This goal quantifies the correlation between predictive features and class variables in learning. The influence of the input features does not directly reveal the underlying mechanism of an AI model. Instead, it helps quantify the importance of features and further improves the quality of feature selection.

Feature Causality. This goal focuses on the causal relations between predictive features and class variables. For example, the objective in the image classification case presents causality between feature and prediction. The causal effects help to discover the underlying mechanism of an AI model.

4.1.3 Process Development for Feature Contribution Explanation

The XAI process development focuses on the explanation activities and the essential data flow. We present the core activity associated with the second research question for feature

contribution explanation.

We identify the significant activities and structure them as an XAI process shown in Figure 13. In the following part of this section, We present the main functionality of each task and the data flow between tasks.

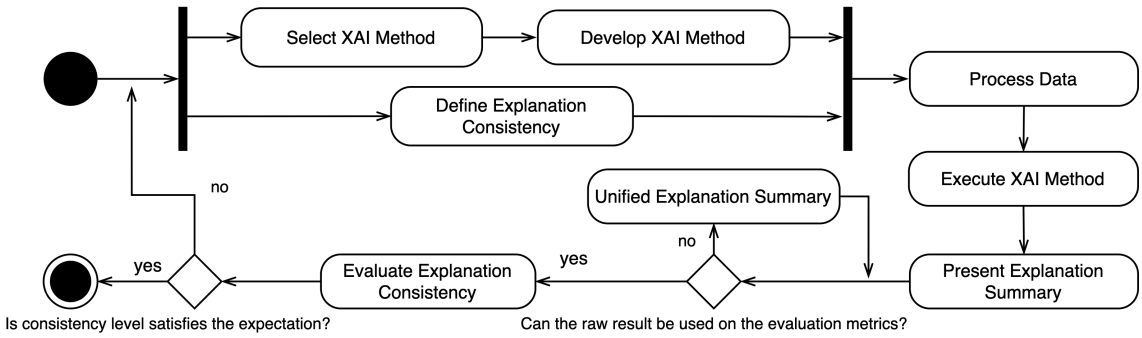


Figure 13: A feature contribution explanation micro process: main activities

Select XAI Method

XAI practitioners explore multiple XAI methods through the taxonomy and identify those to fulfill the XAI criterion. For instance, Accumulated Local Effects (ALE) [9] and Partial Dependence Plot (PDP) [61] both are feature mutation methods by calculate the varied feature values affect the prediction. ALE accumulates the effects of feature value variation within a well-separated interval, while PDP captures the effects of feature value variation within the entire feature space. ALE is a fast and unbiased alternative to the PDP method, but this does not indicate ALE is the default better choice. In the case that the practitioners can not find the proper order of the categorical feature of the target model, then they can not explain this feature with ALE since ALE can only be executed on the ordered feature space. Another example is the method of Model Class Reliance for Feature Importance Estimation [57]. Such a method requires actual outcomes of the data samples to calculate the

expected $loss$ values to generate the explanation. With the selection of multiple XAI methods, XAI practitioners should follow the process and evaluate the explanation consistency for these methods.

Develop XAI Method

Adopting a selected XAI method may further customize the implementation in alignment with the AI model and the data set. A customization is a form of general software configuration. Therefore the practices and tools in software configuration management can be referred to and applied. We omit the details in the discussion. During the execution of XAI methods, hyper-parameter searching and tuning are necessary to gain optimal performance. In addition, cross-validation is essential if the data set may incur unbalanced results for XAI evaluation. One way is to execute the XAI methods with segments of the input data and observe the variation of the XAI results.

Define Explanation Consistency

The same input data set explained by various XAI techniques may yield inconsistent results. Hence we derive explanation consistency as an evaluation metric. The XAI results consistency brings trust to the XAI practitioners. In this work, we present the definition of consistent evaluation in section 4.1.4 and set it as a unit in the process.

Process Data and Execute XAI method

The data processing activity is to perform data cleaning, extraction, segmentation and classification in preparing the data format and quality suitable to an XAI method. In addition to those tasks akin to data preprocessing for model training, the data preparation is also under requirements imposed by the selected criterion. For example, the original data may lead to unbalanced result distribution, which is unsuitable for inputting directly to an XAI method

for evaluation. Hence the data preparation needs to perform extra tasks such as classifying the data sets whose results of XAI methods are more balanced. Then, what follows is the execution of XAI methods. The XAI methods produce explanation results.

Present Explanation Summary and Evaluate Consistency

The process presents the output after XAI runtime. Besides analysis utilizing statistics, tables, or graph plots, the result should be further unified and summarized to match the evaluation metrics. Then, the process validates the results with the defined explanation consistency. Suppose the consistency level does not satisfy the explanation. XAI practitioners may select another XAI method and repeat another round of the process. If the consistency level meets the evaluation, the method can be present for this model and data set, and the explanation results can conclude.

4.1.4 Metric Definition for XAI Method

We focus on the metrics to measure, observe and evaluate the XAI explanation on the same data set across multiple XAI methods and the explanation on multiple data sets generated by the same XAI method. We aim to define the unified metrics that cover the above three scenarios. According to the process definition, the XAI criterion leads to consistency metrics.

Feature Contribution Measurement

The explanation summary generated by an XAI method depends on the model prediction of the input data set.

Consider $\hat{f}(x^{[i]})$ is the model prediction on instance $x^{[i]} < x_1^{[i]}, x_2^{[i]}, ...x_p^{[i]} >$, where p is the number of features. $x^{[i]+}$ represents the instance $x^{[i]}$ under a transformation such as an image with masks. Suppose S is the subset of all the features that we are interested

in their feature importance. j is a feature that belongs to S. R contains the rest of the features, and $P = S \cup R$ makes the whole features P. By marginalizing over features R, the prediction depends on features in S, including the interactions with other features in R. For each instance x of the data set, we define the prediction difference caused by the feature $j \in S$ of interest as absolute difference in Equation 3 and relative difference in Equation 4 respectively.

$$\delta_j^{x^{[i]}}|_{abs} = |\hat{f}_S(x^{[i]}) - \hat{f}_P(x^{[i]})| \tag{3}$$

and

$$\delta_j^{x^{[i]}}|_{rel} = \left| \frac{\hat{f}_S(x^{[i]+}) - \hat{f}_P(x^{[i]})}{\hat{f}_P(x^{[i]})} \right| \tag{4}$$

Furthermore, we define the *feature contribution value* to the payout by the feature j on the prediction of the whole data set X as a function of $\delta_j^{x^{[i]}}$, notated as $\phi_j(\delta_j^X)$. The aggregation of $\delta_j^{x^{[i]}}$ for the whole data set with p number of data set can be different from XAI methods. For example, PDP [61] and Shapley Value [179] and Grad-CAM [168] compute $\phi_j(\delta_j^X)$ as the mean values of the data point $\delta_j^{x^{[i]}}$; while the Mean Centroid Prediff method [115] computes $\phi_j(\delta_j^X)$ as the tangent value of the centroid point of clusters formed by the data points of $\delta_j^{x^{[i]}}$. The higher the feature contribution value, the more important the feature is.

The first case study in subsection A.1 uses the *arXiv* data with thirty-eight data sets. The feature contribution value $\delta_j^{x^{[i]}}$ for a feature j is computed using Equation 3. By ranking features' contribution values in descending order, an explanation is derived as feature importance order on each data set. However, the feature importance orders may vary from different data sets.

For the image classification case study in subsection 4.1.5, multiple XAI methods are applied to the same model prediction on the same data set. The XAI methods produce saliency maps as an explanation, which are used to transform an original image into a

masked image for prediction. The feature contribution value $\phi_j(\delta_j^X)$ is computed by Equation 4.

Feature Explanation Distance

So far, we have set up the one-on-one mapping between an explanation summary and the feature contribution values regardless of the form of an explanation summary. Hence we define the metric *distance* to measure the difference between any pair of explanation summaries. Suppose there are m explanation summaries. The set E contains all the explanation summaries, $E = \{\xi^1, \xi^2, \ldots, \xi^m\}$. Then the distance between any two pairs of summaries is defined as $f_d(\xi^i, \xi^j)$. The choice of selecting any pair of summaries in $E = \{\xi^1, \xi^2, \ldots, \xi^m\}$ has the combination of $K = \binom{m}{2}$. Each choice k produces a distance value as $f_d^{[k]}(\xi^i, \xi^j), (i \neq j, i \leq m, j \leq m)$. The shorter the distance value, the more consistent the XAI explanation summaries are on two data sets. We can now observe and evaluate XAI explanation summaries at two levels, namely *explanation stability* and *explanation consistency* as follows.

Explanation Stability Explanation stability represents the intra-XAI method explanation consistency. When an XAI method is applied to the model prediction with multiple data sets, each data set has an explanation summary produced by the same XAI method.

A violin plot can be applied to visualize the distance distribution and probability density of $K = \binom{m}{2}$ number of distance values to observe the explanation stability. We further aggregate the mean distance value for one XAI method across multiple data sets as follows:

$$f_d^K = \frac{1}{K} \sum_{k=1}^{K} f_d^{[k]}(\xi^i, \xi^j), (i \neq j, i \leq m, j \leq m) \tag{5}$$

Explanation Consistency Explanation consistency presents the inter-XAI explanation consistency across multiple XAI methods on the same data set. It compares different XAI

methods' explanation summaries. Here the number m of explanation summaries are generated from m numbers of XAI methods on the same data set, $E = \{\xi^1, \xi^2, \ldots, \xi^m\}$. For each XAI method I, we compute the distance of its explanation ξ^I to other $m-1$ XAI methods, which produces $m-1$ numbers of distance values $f_d^{[k]}(\xi^I, \xi^j), (I \neq j, I \leq m, j \leq m, k = 1, 2, ..., m-1)$.

Likewise, a violin plot can be applied to visualize the distance distribution and probability density of $m-1$ number of distance values for each XAI method to observe the explanation consistency across XAI methods. We further aggregate the mean distance value for one XAI method I, compared with other $m-1$ XAI methods on the same data set as follows:

$$f_d^I = \frac{1}{m-1} \sum_{k=1}^{m-1} f_d^{[k]}(\xi^I, \xi^j), (I \neq j, I \leq m, j \leq m) \tag{6}$$

Consider the complex comparison scenario under L numbers of data sets, we compute the inter-XAI method explanation distance using Equation 6 for each data set. For each XAI method, we finally collect $(m-1) \times L$ distance data plots to visualize using the violin plot and compute the mean distance values of $(m-1) \times L$ data points.

4.1.5 A Case Study for Image Classifier

The third case study explains the image classifier model with tabular inputs. The selected XAI methods for computer vision are all model-specific. We select six pixel-attribution methods, all CAM-based methods and further evaluate them with our defined metrics over 1,000 image data.

Model and Data

We select the public-released pre-trained CNN model called ResNet50 [79] from Pytorch[1] on the data set Imagenet-1000. The pre-trained model is used for image classification on 1,000 classes. We sampled 1,000 image data from the validation set of ImageNet[2].

Explanation Consistency Analysis

We consider the model-specific XAI method for CNN model and also the pixel-attribution methods that can produce explanations reflecting the active region of the image. According to the taxonomy in Figure 1, we select the Grad-CAM method first and we also identify other CAM-based method, namely, EigenCAM [140], Grad-CAM++ [30], Grad-CAMEW, XGrad-CAM [62], and HiResCAM [48]. The implemetation of those methods are public accessible at this GitHub repository[3].

Summary from Prediction Change Aggregation

Suppose the $\hat{f}_P(x^{[i]})$ is the model prediction on the original image $x^{[i]}$ and the $\hat{f}_S(x^{[i]+})$ is the model prediction on the masked image where $x^{[i]+}$ is the masked image transformed from saliency map $\rho(x^{[i]})$ of the original image. Figure 14 shows the example of deriving the prediction change value.

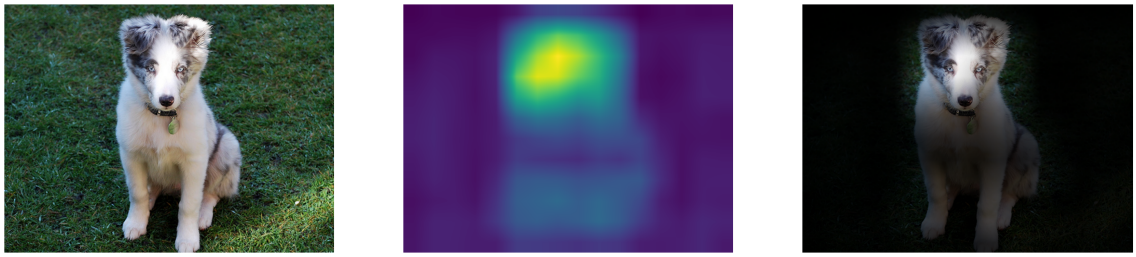


Figure 14: Original image, saliency map generated by the Grad-CAM method, and the masked image. The ground truth label is "collie". The prediction score is 7.5674 on the original image and 5.4336 on the masked one, resulting in a prediction change of 28.17%.

[1] https://pytorch.org/hub/nvidia_deeplearningexamples_resnet50/
[2] https://www.image-net.org/
[3] https://github.com/jacobgil/pytorch-grad-cam

For all the selected CAM-based methods, we define the prediction change on data sample $x^{[i]}$ with Equation 4 as:

$$|\frac{\hat{f}_S(x^{[i]+}) - \hat{f}_P(x^{[i]})}{\hat{f}_P(x^{[i]})}| \times 100, (x^{[i]+} \leftarrow \rho(x^{[i]}))$$

We then get all the prediction changes on data set X for each method.

Stability Observation

To derive the stability metric in this case, we set each prediction change on the data sample as one summary and get the distances of any two pairs of the prediction change with the distance function $\hat{f}_d(\xi_i, \xi_j) = |\xi_i - \xi_j|$. Figure 15 shows the distribution of the distances between the prediction changes. By using Equation 5, we evaluate the stability for a method. As can be observed, the Grad-CAMEW method has the lowest mean value on the stability metric.

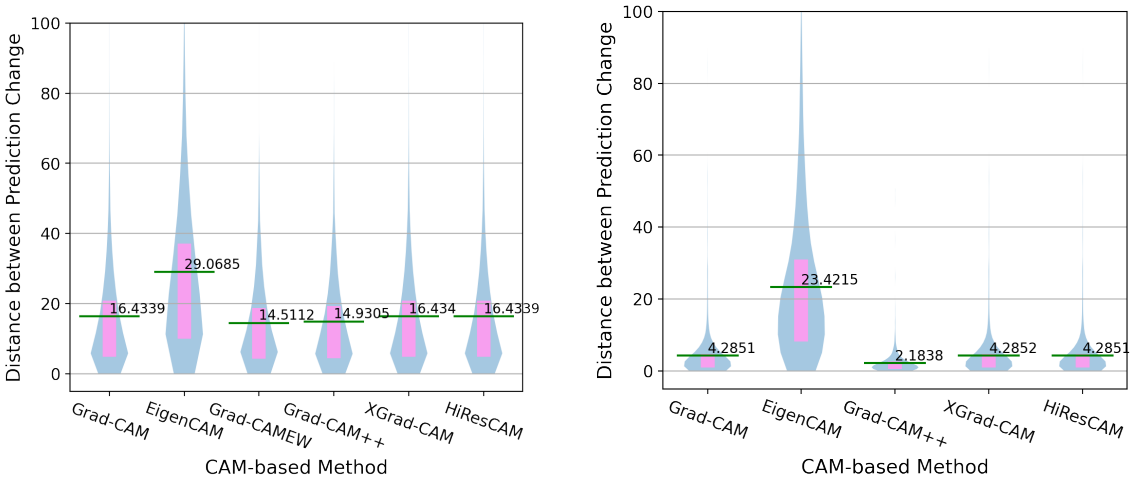


Figure 15: Explanation stability and consistency. Left: Stability across methods; mean value indicates robustness. Right: Consistency of Grad-CAMEW compared to other methods. Green lines denote the mean values.

Consistency Observation

To calculate the consistency between Grad-CAMEW method and the other methods, we use the Equation 6 with the distance function $\hat{f}_d(\xi_i, \xi_j) = |\xi_i - \xi_j|$. Figure 15 shows the

distribution of consistency over 1,000 images.

4.1.6 Summary

This work proposes a micro process for structuring the activities, entities and artifacts essential to an XAI project for feature contribution analysis. We analyze the state-of-the-art XAI goals and derive criteria. We define a taxonomy as the basis of developing the XAI process. The micro process has the benefit of producing instances of XAI according to the data set classes and methods. We define two explanation consistency metrics by analyzing the explanation generation consistency within and across the methods. Thus, it becomes practical for large-scale evaluation of such XAI projects. We provide three use cases for explaining an AI-powered scholar literature ranking model, an NLP code vulnerability detection model, and an image classification model. The first case study performs a total of 152 experiments with four instances of the XAI process. Each instance maps to one XAI method and runs thirty-eight classes of data sets which include 542,877 papers. The second case study performs feature contribution explanation analysis on the NLP classification model with also four instances of the XAI process. The third case study reaches the problem of image classification. Hence, the detailed implementation of the metric evaluation differs from the previous two cases while we conduct six instances of the XAI process. The feature contribution analysis micro-process becomes the composition unit for large-scale XAI projects with large-scale data sets and many target models. Future work could develop the comprehensive XAI process for more kinds of XAI goals and different kinds of XAI methods. More explanation evaluation metrics could be identified to discover different characteristics of the XAI method.

4.2 XAI Assessment for Open Repository Models Under Adversarial Attacks

Artificial intelligence models are increasingly accessible through the open community [209] and facilitate advancements in software applications across various domains. These advancements, represented by innovations in deep-learning models such as ViT, ConvNeXt, CVT, Swin Transformers, SegFormer, and ResNet [47, 79, 120, 121, 210, 212], underscore the rapid evolution of AI capabilities. However, the integration of these models into software applications necessitates a strict evaluation [217] of their quality attributes. A noted gap in current practices is the absence of a comprehensive service framework that facilitates an understanding of model explainability [46, 124].

The importance of explainable AI (XAI) has been increasingly recognized [75], driven by the need to build trust and ensure fairness within AI systems. XAI techniques, which aim to make the decision-making processes of AI models transparent [13], are essential in AI-enabled applications [190]. Additionally, adversarial attacks target vulnerabilities of AI models. They are modifications to input data that are less visible to humans but can make AI models give incorrect inferences or predictions [80, 137]. The rising threats [225] targeting security-sensitive models introduce significant challenges to the deployment of AI in software services. Therefore, effective software quality assurance requires a comparative analysis to guide the development and refinement of AI models, ensuring their robustness and explainability across diverse applications.

The integration of XAI techniques into AI models introduces additional computational layers [94]. The operational complexity of XAI evaluation [85] arises from multiple factors of managing diverse data types, integrating explanation methods with various AI models, evaluating explanations, and summarizing the results. These factors lead to evaluation scenarios that require dozens to hundreds of experiments. In addition, the complexity of

assessment is further scaled by the product of multiple kinds of adversarial attacks. Comparative analysis between XAI and adversarial attacks increases the evaluation workload by necessitating the exploration of interactions between models and explanations under various adversarial conditions.

The goal is to address these complexities by automating the evaluation pipeline. We propose a cloud-based service framework that encapsulates computing components as microservices and organizes assessment tasks into pipelines. This framework also integrates open XAI tool libraries, which are originally not inherently service-oriented, as part of the pipeline composition.

In scenario studies, we assess six vision models against three types of adversarial attacks employing five XAI methods, resulting in a total of ninety distinct combinations. We evaluate three transformer-based tabular models using two XAI methods across three datasets, resulting in eighteen combinations. We set pipelines to assess quality attributes for every combination scenario, including (1) computational cost, (2) performance, (3) robustness, (4) explanation deviation, and (5) explanation resilience. We assessed these attributes across a range of model and XAI method combinations. The evaluation results indicate that higher explanation deviation requires more computational costs. We demonstrate the impact of adversarial attacks on model performance and their explanation.

4.2.1 Related XAI Tools Libraries and Image Adversarial Attacks

In this subsection, we introduce the XAI methods. We review the current XAI tools and frameworks. Then, the taxonomy of adversarial attacks.

Mainstream post-hoc XAI methodologies can be categorized into two distinct types [85]: model-specific and model-agnostic methods. Model-specific methods [**?**, 30, 48, 62,

92] derive feature importance values from the internal parameters of the model itself, offering insights directly linked to the model's internal mechanisms. In contrast, model-agnostic methods [126] establish feature importance by analyzing the relationship between the model's inputs and outputs without requiring the model's internal parameters and layers.

XAI in computer vision often involves generating visual explanations for the decisions made by AI models. The model-specific methods are computationally efficient, rather than the model-agnostic methods [1]. For demonstration, Figure 16 presents selected examples derived from the Vision Transformer model [47] through our XAI service framework. Significant variations in results are observed from the same input data. These results indicate the need for establishing systematic XAI evaluations within the XAI service framework, which is notably absent in previous tools and frameworks.

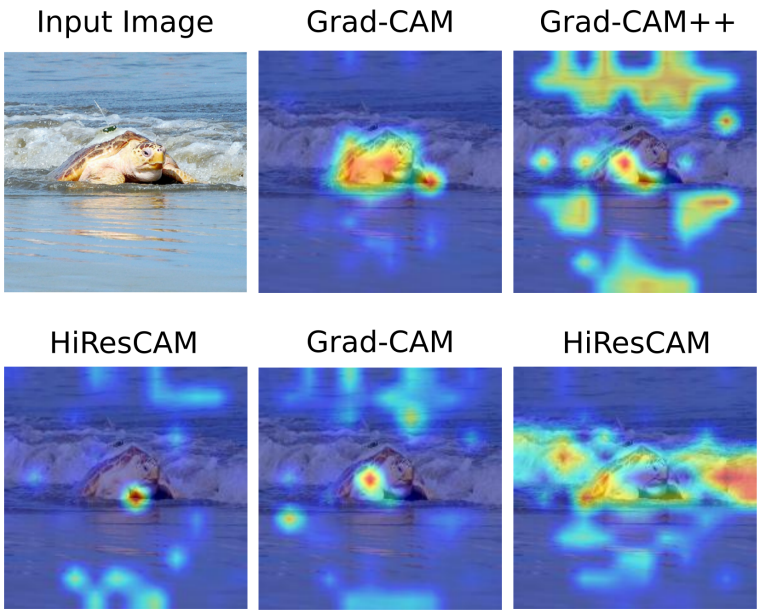


Figure 16: Five CAM-based Visual Explanations from Vision Transformer Model with One Image Example.

Grad-CAM [**?**] utilizes the gradients of the target label flowing into the final convolutional layer to produce a coarse localization map highlighting important regions for prediction. According to an optimization work, Grad-CAM++ [30] extends Grad-CAM by considering the weight of each pixel in the feature maps, allowing better handling of images with multiple occurrences of the same object. HiResCAM [48] generates high-resolution class activation maps, allowing for finer detailed visual explanations with higher computational cost. XGrad-CAM [62] focuses on increasing the linearity of the saliency maps, providing improvements. LayerCAM [92] introduces a layer-wise relevance propagation mechanism, which enables the feature importance across different network layers for more details.

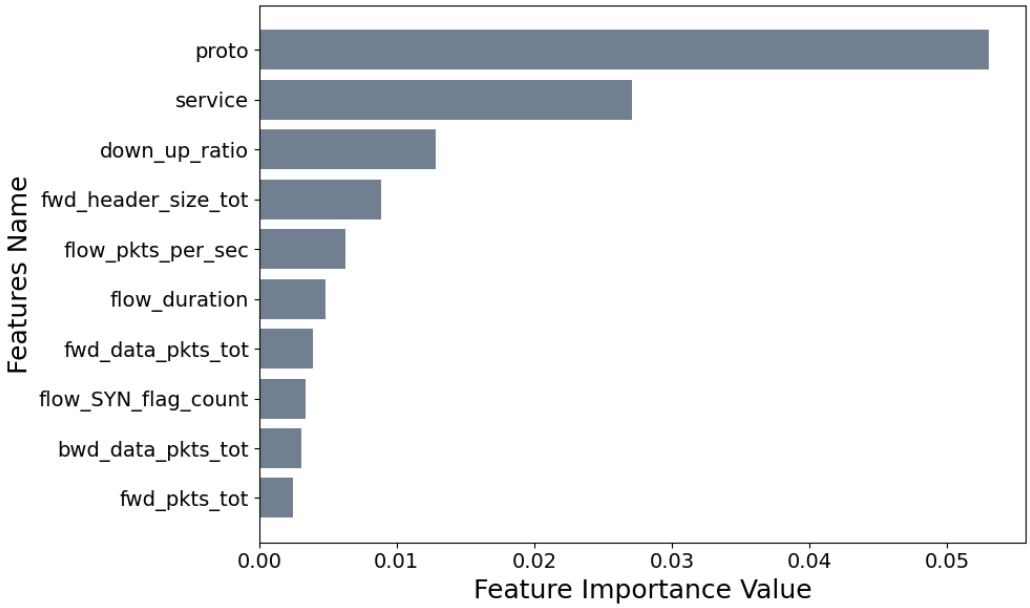


Figure 17: Top 10 out of 83 SHAP Feature Importance Explanations from FT Transformer on RT-IoT Cybersecurity Threats Dataset.

We also apply XAI methods involving structured data within the tabular models. Mean Centroid Prediff [115] and SHAP (SHapley Additive exPlanations) [126] stand out for widespread acceptance as the baseline. These methods quantify feature importance by assessing the impact of masking each feature on model output. The methods provide a score

for each feature, indicating its contribution to model predictions. Figure 17 illustrates an example of the top ten out of eighty-three global feature importance explanation from the RT-IoT2022 cybersecurity classification dataset [160]. We carry out XAI to determine the impact of the network logs features on the classification of threats. We provide data-driven explanation deviation metrics to evaluate XAI methods in the subsection 4.2.5 scenario studies.

XAI Tools Libraries and Frameworks

The role of XAI in providing feature contribution explanations establishes trust in AI models [115]. Before introducing XAI as a service, we review the published tools and libraries. The Explainability 360 toolkit by IBM [14] integrates their explanation techniques within the toolkit package. Microsoft's InterpretML [144] offers support for eight tabular models' XAI approaches. The recent framework OmniXAI [215] offers a broad range of techniques for XAI. However, based on our tests and usages, the existing XAI frameworks have the following limitations:

Expertise-Dependent Usage: The use of these tools often demands specific XAI knowledge expertise, which limits accessibility [147] to software engineers.

Restricted Methods Support: Each library support limited numbers of different XAI methods [14, 144, 215].The disparity in the number of methodologies supported by these tools complicates the selection process and necessitates additional preprocessing.

Evaluation Procedures Deficiency: The comprehensive study [118] that performed quantitative evaluations on saliency methods is relevant. However, the current XAI tools lack standardized procedures for evaluating explanation results, which limits the selection and enhancement of these XAI tools [124, 226].

Cloud Service Support Limitations: Explicit support for cloud AI services is rare among these tools, affecting their applicability across cloud service environments.

These limitations underline the need for a service architecture that addresses these gaps, making XAI accessible and effective for diverse AI-enabled applications.

Adversarial Attacks Types

Adversarial attacks are designed to mislead models, and pose challenges to AI system security [211]. The adversarial attacks are briefly categorized into white-box and black-box attacks, each with distinct methodologies and implications. The taxonomy of adversarial attacks with example works are listed in Figure 18.

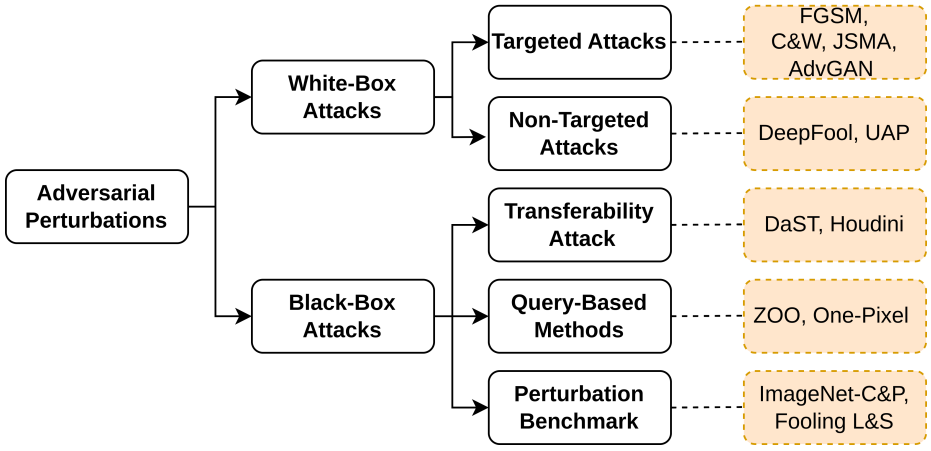


Figure 18: Taxonomy of Adversarial Attacks. References: FGSM [67], C&W [28], JSMA [149], AdvGAN [211], DeepFool [138], UAP [137], DaST [229], Houdini [36], ZOO [33], One-Pixel [181], ImageNet-C [80], ImageNet-P [80], Fooling LIME and SHAP [173].

In white-box attacks, the attacker needs certain knowledge of the model, including its architecture and parameters. Targeted attacks [28,67,149,211] aim to manipulate the model to produce a specific, incorrect output. Non-targeted attacks [137,138] aim to generate any incorrect response from the model.

In black-box attacks, where the attacker has no information about the model's internals, the focus shifts to transferability attacks, query-based methods, and perturbation benchmarks. The perturbation benchmarks refer to algorithmically induced modifications applied to the data with the intent to mislead the model. The benchmark modifications can affect

models and further be employed to measure the model's robustness [80]. The study [80] presents corruption and perturbation as two methodologies for modifying images. Corruption typically refers to modifications that simulate natural or environmental degradation. Perturbation denotes generated changes in sequence. We merge these two groups algorithmically modify methods in the study [80], under the broader category of adversarial perturbations. In the assessing scenarios, we adopt the ImageNet-C benchmark [80]. The benchmark [80] contains fifteen types of algorithmically generated corruptions from categories such as noise, blur, weather, and digital. The weather category contains corruptions that demonstrate an excessive diversity. In the assessment scenarios, we select a representative algorithm from each of the three main categories. We apply nine distinct corruption perturbations across three levels of severity. The Fooling LIME and SHAP [173] method is a recent study using data perturbation to attack LIME [154] and SHAP [126]. We also develop and launch multiple levels of perturbation attacks to the numerical features in selected tabular datasets for scenarios.

4.2.2 Overview of AI System's Quality Attributes

AI-based system refers to a software system that adopts the AI models as components [55]. Software quality is the ability of a software product to meet requirements during its operation in designated conditions [105]. Software quality assurance involves an evaluation process of how well a software product satisfies its stated needs [141]. However, the AI models are distinct from conventional software components. Their core characteristic is being data-centric [8]. Additionally, these components are dynamic and continually evolving as they are exposed to new data over time [8]. Inadequate, inaccurate, or unsuitable training data can result in unreliable models and biased decisions [26]. To address quality assurance, the study [200] presents the concept of early adoption of XAI into model development.

We define the quality attributes of AI models and the explanations produced by XAI analysis in terms of `model performance` and `explanation deviation`. Under adversarial attacks, these quality attributes are further specified as `model robustness` and `explanation resilience`. Additionally, `computation cost` is considered in relation to deployment. The metrics established for these quality attributes encapsulate the combined states of AI models, XAI methods, adversarial attacks, and datasets. The results can be visualized using a radar chart, where each quality attribute is transformed into a normalized value.

Computational Cost. Computational cost measures the resources required to execute an algorithm. The computational cost includes measuring the utilization of CPU, GPU, and Memory. We record the computational cost attribute that encompasses the runtime and energy consumption of AI models and XAI techniques. These metrics are measured in seconds (s) for runtime and watt-hours (Wh) for energy consumption. The CodeCarbon library [106] supports the program to track resource utilization by monitoring hardware-specific parameters. In addition, the CodeCarbon estimates the environmental carbon footprints based on energy consumption.

Model Performance. The evaluation of model performance contains a range of metrics. The typical metrics are Top-N accuracy metrics [38], precision, recall, and F1 score [130]. In the multi-class context, we aggregate True Positives (TP), False Positives (FP), True Negatives (TN), and False Negatives (FN) across all classes. In addition, the Area Under the Receiver Operating Characteristics Curve (AUC-ROC) is also a calibrating metric to assess a model's ability to distinguish between classes [129].

Model Robustness. Robustness evaluation measures a model's performance degradation under specific adversarial conditions. For computer vision models, the use of an image adversarial perturbation benchmark [80] assesses the model under adversarial conditions. Mean Corruption Error (mCE) represents the model robustness in a previous study [80].

This metric aggregates the normalized error rates for a model across corruption types and their respective severity levels. The formulation for mCE [80] is detailed below:

$$mCE = \frac{1}{S_c} \sum_{s=1}^{S_c} \left(\frac{E_s^c(f)}{E_s^c(f_{\text{ref}})} \right) \tag{7}$$

Here, f denotes the model under evaluation. For corruption type c, S_c indicates the severity levels. The variable $E_s^c(f)$ is the error rate of the model f, while $E_s^c(f_{\text{ref}})$ specifies the error rate of a reference model, such as Alexnet according to the study [80].

Instead of comparing with a reference model, we set Kolmogorov-Smirnov (K-S) statistic [49] D_{ks} as a quantifiable metric. It represents the comparison of the distribution of the model outputs between two datasets:

$$D_{ks} = \sup_X |F(X_{\text{orig}}) - F(X_{\text{adv}})| \tag{8}$$

Here, $F(X_{\text{orig}})$ and $F(X_{\text{adv}})$ symbolize the model probabilities distribution of the original and adversarial datasets, respectively. The usage of $\sup_x$ targets the maximum divergence between these two model inference distributions. By incorporating the K-S statistic, we redefine the robustness against adversarial attacks. Equation 9 represents the assessment. A smaller value indicates better model robustness:

$$Robustness = \frac{1}{S_c} \sum_{s=1}^{S_c} D_{ks} \tag{9}$$

For the tabular model, we generate feature perturbations to datasets. Specifically, in the scenario, we make a random perturbation to the feature with the numerical severity factor. Similar to our approach, the adversarial attack [173] introduces a designed biased perturbation, instead of our random perturbation, to attack the SHAP [126] and the LIME [154] methods. The robustness Equation 9 can be used in tabular cases.

Explanation Deviation. Explanation deviation is an assessment of the impact of the

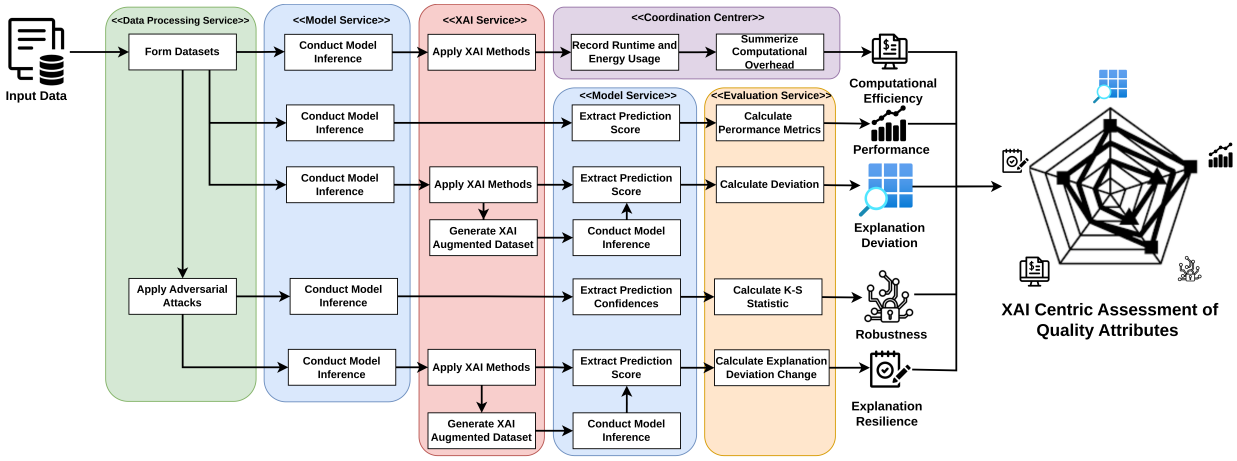


Figure 19: Assessment Pipelines for Open-source AI Model Quality Attributes.

feature importance explanation on the model's outputs. At its core, explanation deviation measures the discrepancy between a model's predictions when all features are considered versus when only those important features are emphasized. Therefore, measuring explanation deviation is a means to validate the actual influence of presumed features on the model's outputs.

In vision scenarios, the saliency map is applied to assess how important each pixel is to a model's output, creating visual importance maps. Grad-CAM [**?**] is commonly accepted in CNN-based vision model [79] explanation. However, whether these XAI perform as effectively for more transformer-based models remains under evaluation [174]. Explanation deviation is determined by the prediction change score, which indicates the impact of the saliency map area on the model's predictions. The masked image is calculated as the element-wise multiplication of the original image X and the normalized, three-dimensional saliency mask $M_{\text{norm 3D}}$. Here, P is defined as the model's prediction class probability values:

$$Deviation = 1 - (P(X_{\text{orig}}) - P(X_{\text{orig}} \cdot M_{\text{norm 3D}})) \tag{10}$$

A smaller change in probability value, then the deviation close to one. Extending to the

whole dataset, the overall explanation deviation is the statistical analysis of their median value. In tabular scenarios, the XAI method computes feature importance values for each data sample. The explanation deviation is represented by evaluating the consistency in the feature importance order, according to the study [85].

Explanation Resilience. Resilience to adversarial attacks is measured by the difference in explanation deviation between non-adversarial and adversarial conditions. Therefore, the explanation resilience is quantified by the Equation 11:

$$\begin{aligned} Resilience =& (P(X_{\text{orig}}) - P(X_{\text{orig}} \cdot M_{\text{norm 3D}})) \\ & - (P(X_{\text{adv}}) - P(X_{\text{adv}} \cdot M_{\text{norm 3D}})) \end{aligned} \tag{11}$$

The resilience attribute indicates that the explanation deviation metric decreases for adversarial reasons. For tabular data, resilience can be calculated by subtracting the explanation deviation of adversarial data from the original data. Comparing the original and perturbed situation allows for resilience assessment: The smaller the resilience, the better the explanation can resist the impact of adversarial attacks.

4.2.3 Pipelines of XAI Centric Assessment of Quality Attributes

In this subsection, we define the pipeline for obtaining quality attributes. Then, we illustrate the cloud-based service architecture. The operational overhead for XAI service is compared with using the existing framework.

Define Pipelines for Each Quality Attribute

This study integrates five quality attribute evaluations. Figure 19 shows the multiple parallel processes for accessing quality attributes. This multifaceted assessment evaluates quality attributions under both original and adversarial conditions.

Computational Cost Pipeline: This pipeline records the computational resources used during model inferences and XAI method executions. The average resource consumption is logged by the `Coordination Center`, when processing a significant volume of data inputs. The record covers AI models, XAI methods and evaluations.

Model Performance Pipeline: To assess model performance, the pipeline begins with the datasets provided by the `Data Processing Microservice`. These datasets are sequentially fed into the `Models Microservice`, which encapsulates the pre-trained model. The outcomes are extracted and persisted. The `Evaluation Microservice` derives performance metrics.

Explanation Deviation Pipeline: This pipeline calculates explanation deviation. `XAI Methods Microservice` are applied to generate explanations. For computer vision tasks, the explanation results are fed back into the model to calculate the prediction confidence drop, thereby calculating explanation deviation. The prediction changes value within consistency metric [85] can also be directly derived from tabular models.

Robustness Pipeline: To assess model performance decline under adversarial attacks, the pipeline begins with the `Data Processing Microservice` preparing perturbed data. The `Models Microservice` then processes both the original and perturbed datasets, yielding two sets of results. The `Evaluation Microservice` measures shifts in the model's performance between the original and perturbed datasets.

Explanation Resilience Pipeline: Similar to the robustness pipeline but with a focus on XAI methods, this pipeline starts with the `Data Processing Microservice` preparing perturbed datasets. These datasets are then subjected to selected models and XAI methods. The `Evaluation Microservice` calculates the changes in explanation deviation.

The individual pipeline is configurable for AI models, XAI methods, and adversarial datasets. They can be composed into singular or combined quality attributes assessment

scenarios. Their deployment and execution need a runtime service-oriented architecture, which is defined in the next subsection.

Define Services Architecture for the Pipeline Configuration

We introduce a cloud service architecture designed to develop XAI assessment pipelines. This architecture enables analysis and comparison of various combinations of AI models, XAI methods, datasets, and adversarial attack approaches, as shown in Figure 20.

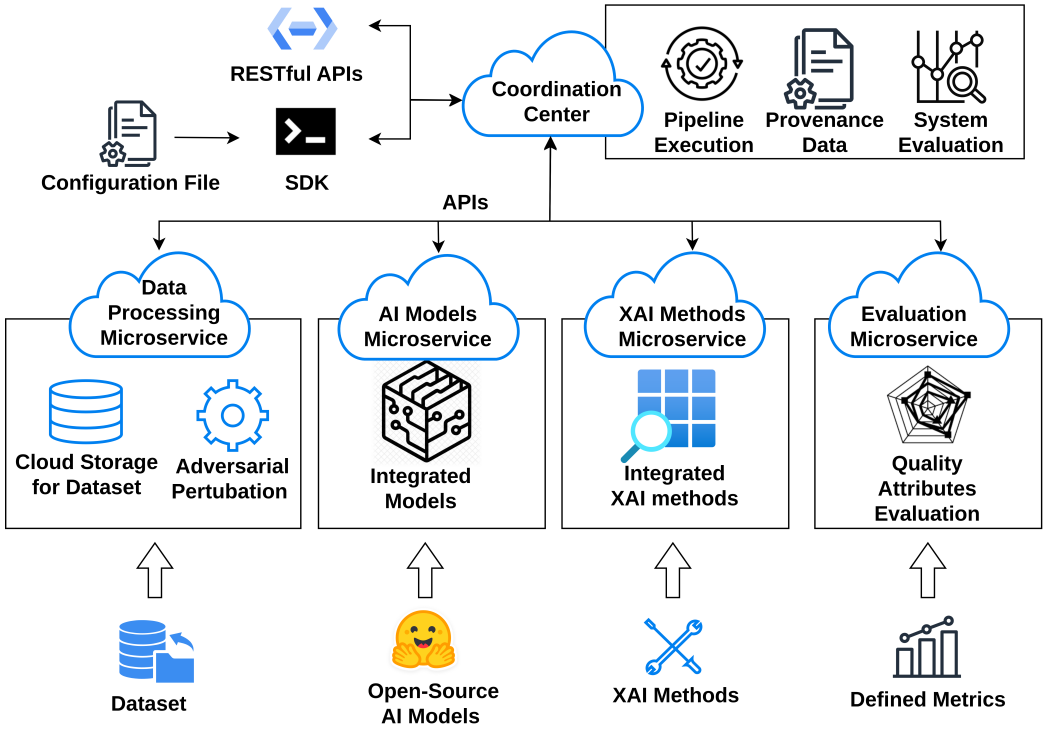


Figure 20: Cloud-based XAI Service Architecture.

Coordination Center: The microservice executes unit operations based on configuration, communicates with other microservices as per task setup, and records provenance data for transparency and reproducibility.

Data Processing Microservice: This microservice ensures data is correctly formatted and meets XAI algorithm requirements. It also applies adversarial attack conditions.

AI Model Microservice: This service encapsulates and deploys pre-trained AI models,

including open-source models from the HuggingFace Community, within its framework.

XAI Method Microservice: This service offers XAI that can generate explanations. XAI algorithms [115] and tool libraries [215] can be encapsulated into the service.

Evaluation Microservice: This service aggregates results and systematically evaluates defined quality attributes.

Collectively, these components create a cloud-based architecture supporting the complete evaluation process. The architecture's flexibility allows for switching services to test various AI models or XAI methods, facilitating extensive investigation into numerous combinations. Additionally, the deployed services are reusable across multiple pipelines.

```
"xai_config": {
  "base_url": "xaiport.ddns.net:8003", //"address"
  "datasets": {
    "t1024_gaussian_2": {     //"datasets id"
      "model_name": "resnet", //"model name"
      "algorithms": [
        "GradCAM", //"XAI methods name"
        "HiResCAM",
        "GradCAMPlusPlus",
        "XgradCAM",
        "LayerCAM"
      ]
    }
  }
}
```

Figure 21: Sample JSON Template for XAI Methods Configuration

In addition, the JSON-based configuration template, as present in Figure 27, specifies how the service executes the pipeline according to the user's inputs. The template allows users to define the interaction between different services and customize the evaluation process to suit specific requirements. The execution of pipelines is through the use of coordination centers, each configured with its JSON-based configuration file. Upon receiving a configuration file, a coordination center systematically accesses the specified microservices to execute each pipeline step.

4.2.4 Comparative Evaluation and Service Integration of Existing XAI Frameworks

The architecture enables the encapsulation and customisation of the external XAI algorithms and frameworks. XAI microservices can encapsulate not only XAI methods but also published libraries and frameworks. For instance, OmniXAI [215] aggregates enriched XAI methods for various data types. We import the OmniXAI package and employ the related functionalities to compute explanations in the XAI methods microservices. The rest of the required units can continue to adopt our service architecture. To verify the integration of the external tool framework, We compare our XAI service and the recently published OmniXAI framework [215]. We focus on differences in task reproducibility, metrics, and ease of use.

Reproducible: Reproducible ensures that results can be reliably validated. Other frameworks offer explanation methods but lack detailed provenance data recording. Validating and reproducing an XAI process requires obtaining the original dataset and editing the source code. This demands effort to reproduce the complex XAI process manually. Our service framework makes the evaluation pipeline reproducible using provenance data. Users can execute the pipeline with a single command using provenance data as the configuration file.

Evaluation Metrics: Other XAI frameworks do not provide evaluations for their methods. Without rigorous evaluation, the effectiveness of the explanations provided by different XAI remains undetermined. Our framework introduces the calculation of XAI consistency metrics [85]. We also define the XAI-centric assessment quality attributes.

Ease of Use: Our framework provides a streamlined SDK interface that simplifies pipeline execution. Table 6 provides a described comparison of the operational steps. The JSON-based configuration template allows users to select datasets, models, and XAI methods in a structured manner. By automating several steps such as data preparation and

Table 6: Operational Overhead Comparison Between XAI Service and Tool Framework

Steps	Our Service	OmniXAI [215] Framework
Data Preparation	Automate data upload with formatted templates.	Script data transformation and insertion.
Environment Setup	Automate environment setup through Docker container.	Manage dependency installation, library configuration, and compatibility.
Configuration	Utilize JSON for configuration with defined rules.	Configure interactions and dataset linkages manually.
Pipeline Execution	Execute pipeline with a single SDK command.	Load datasets, inferences, and compute XAI manually.
Results Analysis	Provide built-in metrics and visualization.	Summarize results with custom coding.
Adjustments	Support adjustments via JSON editing.	Modify code and restart processes manually.

environment setup, our service minimizes the operational overhead for systematic XAI processes.

Result Values Comparison: We compared the explanation deviation results of our service framework with those of existing tools using the ImageNet dataset [43, 80] test set, containing ten thousand images.

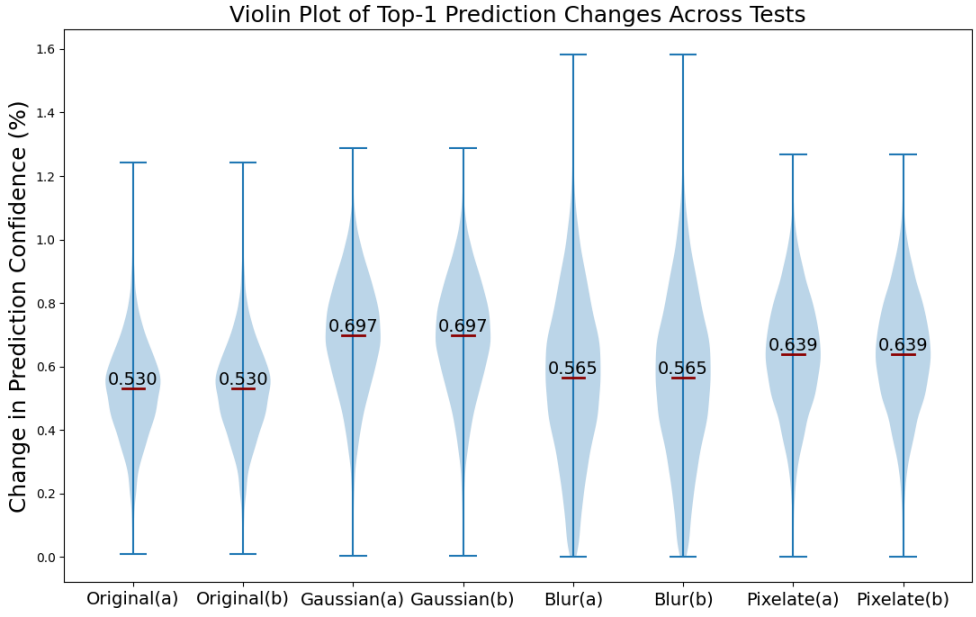


Figure 22: Explanation Deviation Analysis for XAI Service (a) Versus OmniXAI (b) Using GradCAM on ResNet

Our XAI service includes multiple CAM-based methods [**?**, 30, 48, 62, 92]. The recent tool framework OmniXAI [215] employs only Grad-CAM [**?**] as the XAI method for vision models. Figure 22 presents a comparative analysis of the results from implementing Grad-CAM algorithms in our service framework versus OmniXAI. The results show the same prediction confidence distribution. However, our XAI service framework significantly streamlines operations.

Additionally, our framework is containerized and suitable for cloud platform environments. The RESTful API design enables easy integration with external cloud model services.

4.2.5 Assessment Scenarios and Quality Attribute Analysis

We introduce experimental scenarios to demonstrate the service's functionality in scenarios. By running the pipelines defined by the service framework, we aim to investigate the research questions as follows:

- Are the explanation deviation generated by XAI methods variable across models with different structures?
- What is the relationship between computational cost and explanation deviation in model-XAI combinations?
- Considering the known impacts of adversarial perturbations on model performance metrics, how do these perturbations influence the explanation deviation?

The related source code and experimental results can be found in the GitHub repository [4]. Experiments are conducted in a controlled environment to ensure consistency. The experimental evaluation used a local setup with an Nvidia RTX 4090 GPU based on the AD102 graphics processor. This GPU features a core clock speed of 2.52 GHz and 24 GB

[4]https://github.com/ZeruiW/XAIport

GDDR6X VRAM. All experiments were conducted on a Linux-6.2.0 system, Ubuntu 22.04 LTS, Python 3.8.18 with CUDA 12.1, and PyTorch 2.1.0 to leverage the GPU's capabilities.

The Assessment Scenario of Vision Models

Process Configuration: Our dataset comprised 10,000 images from ImageNet [43], covering one thousand classes. We introduced three types of adversarial perturbations in the ImageNet-C [80]: Gaussian Noise, Defocus Blur, and Pixelate, with each type occurring at three severity levels.

Table 7: Selected State-of-the-Art Vision Models from the Huggingface Repository

Vision Model	Publisher	Huggingface Repository
ViT	Google	google/vit-large-patch32
ConvNeXt	Meta	facebook/convnext-tiny
CVT	Microsoft	microsoft/cvt-13
Swin	Microsoft	microsoft/swin-large-patch4
SegFormer	Nvidia	nvidia/mit-b0
ResNet	Microsoft	microsoft/resnet50

As list in Table 7, the models selected for evaluation are Vision Transformer (ViT) [47], Convolutional Neural Networks Next (ConvNeXt) [121], Convolutions Vision Transformer (CVT) [210], Swin Transformer (Swin) [120], Semantic Segmentation with Transformers (SegFormer) [184], and Residual Networks (ResNet) [79]. These models are top-performance sourced from the Huggingface open repository. The publisher has pre-trained these models. We select the ResNet [79] as a baseline. Additionally, our experiments employ various CAM-based methods [**?**, 30, 48, 62, 92] to generate saliency maps for vision models.

Computational Cost Analysis: Evaluating computational costs is essential to understand the practical implications of deploying XAI techniques in applications. The assessment covers processing time and energy consumption for different AI models using XAI techniques.

Table 8: Analysis of Computational Costs for Vision Models (per 1,000 images)

Model	Inference Time (s)	Mean XAI Time (s)	Pipeline Energy (Wh)
ViT	13.35	62.50	5.58
ConvNext	5.73	41.19	2.67
CVT	16.91	69.76	5.86
ResNet	10.37	45.34	3.32
Swin	**32.38**	**137.24**	**10.07**
SegFormer	13.62	65.08	5.73

Table 8 illustrates each model's combined inference time, XAI time, and total energy in the pipelines. The results indicate significant variation in total processing time and energy consumption across models. The Swin Transformer shows the highest total process time and energy consumption, indicating a significant computational demand. The CVT model consumes the second most energy and runtime. Conversely, ConvNext has the lowest energy consumption and a relatively shorter total processing time. This is reasonable because the employed ConvNext model [121] is a relatively tiny-sized model. This conclusion suggests that CNN-based models, such as ConvNext and ResNet, are still potentially more viable options for applications where energy efficiency and faster processing are crucial.

In summary, these findings indicate significant variations in both time and energy consumption across different models. XAI consumes significantly higher computational costs compared to model inference.

Model Performance Analysis: The objective is to evaluate and compare the performance of leading open-source vision models from the Huggingface repository, applying a range of adversarial perturbations.

Figure 23 displays the F1 scores for six computer vision models under both original and adversarial perturbation conditions, emphasizing their performance. Initially, all models exhibit high performance on the original dataset. However, adversarial perturbations have varying effects on the performance of each model. Models indicate performance degradation under adversarial perturbations. Specifically, CVT and SegFormer show obvious

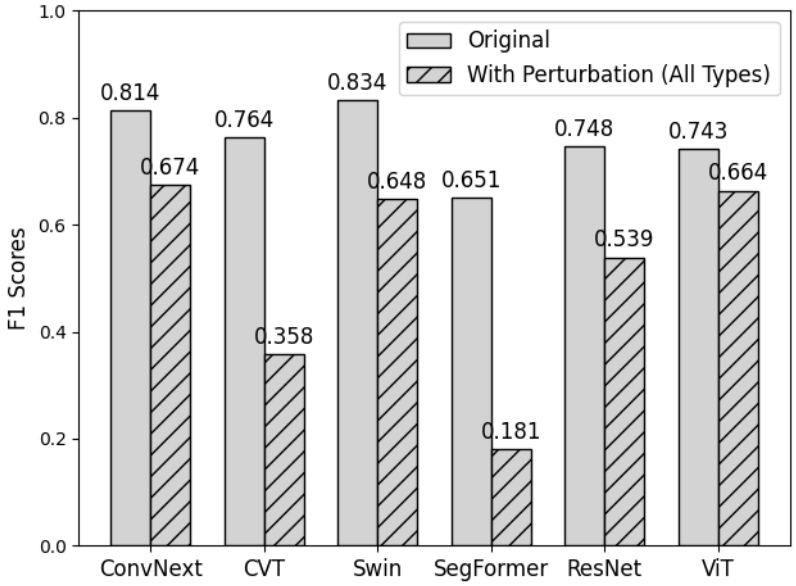


Figure 23: Vision Models F1 Scores: Original Dataset and Adversarial Dataset Averages.

vulnerability against adversarial perturbations.

Model Robustness Analysis: This analysis evaluates model robustness against different levels of adversarial perturbations. The aim is to quantitatively evaluate and compare the robustness by assessing their ability to maintain prediction accuracy. To analyze changes in prediction distributions under adversarial perturbations and further assess model robustness, we employ the introduced Kolmogorov-Smirnov (K-S) statistic. Initially, we extract the probability values from the model inference of the original dataset. Next, we extract prediction probabilities for each perturbed dataset and compute the K-S statistic. A higher K-S value indicates a greater shift in the model's outputs, implying a significant reduction in robustness.

Figure 24 shows the K-S values offer perspectives on model robustness. SegFormer consistently exhibits higher K-S values across all perturbations, indicating significant prediction shifts under adversarial attacks. This suggests vulnerability in maintaining prediction consistency against such perturbations. Conversely, models such as ViT and Swin exhibit lower K-S values, implying more robust performance under adversarial conditions.

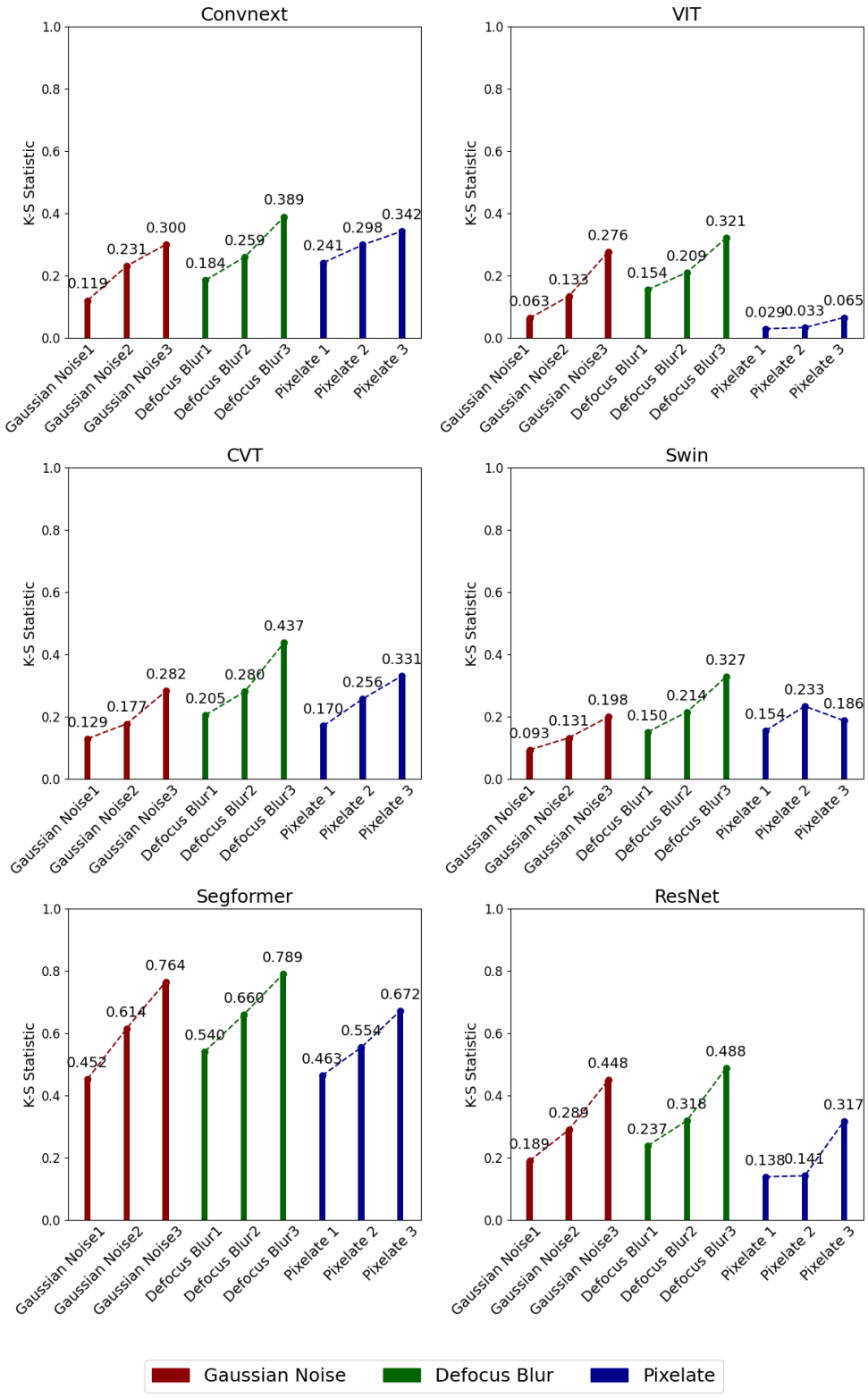


Figure 24: Comparison of K-S Statistics to Assess Model Robustness under Three Levels and Types of Perturbations (0: Identical, 1: Highly Divergent).

The bar plots in Figure 24 also display K-S statistic values for each perturbation type (Gaussian Noise, Defocus Blur, Pixelate) across three levels (1, 2, 3). Each bar's height indicates the deviation of the model's prediction distribution from its original, caused by a specific perturbation. Different colors for each perturbation type enhance visual distinction.

As a result, this quantitatively evaluates the robustness of various models under diverse adversarial perturbations. The Vision Transformer and Swin Transformer models exhibit relatively higher robustness.

Explanation Deviation Analysis: The objective is to evaluate the impact of adversarial perturbations on the explanation deviation. The utilization of saliency maps to annotate images assists developers in identifying the reason for model inaccuracies, thereby enhancing model performance.

Figure 25 shows prediction change percentages on a heatmap. The explanation deviation attributes can be calculated as one minus the value shown in the figure. The figure is 3-dimensional: the left rows label six vision models, while the right rows specify the original dataset and three types of adversarial perturbation. The labels at the bottom represent various XAI methods. The value in each cell reflects the summarized percentile of prediction value changes observed between the original input and the explanation inputs. Lower values mean the model's inference is less affected by irrelevant feature masking, indicating XAI makes a closer approximation of relevant features. Consequently, a lower summarized prediction change percentage values in Figure 25, correlates with better XAI explanation deviation.

Significant variations are observed across different model-XAI combinations. The results reveal that the Swin Transformer model maintains consistent explanation deviation across varied XAI methods and adversarial perturbation types, as confirmed through extensive cross-validation.

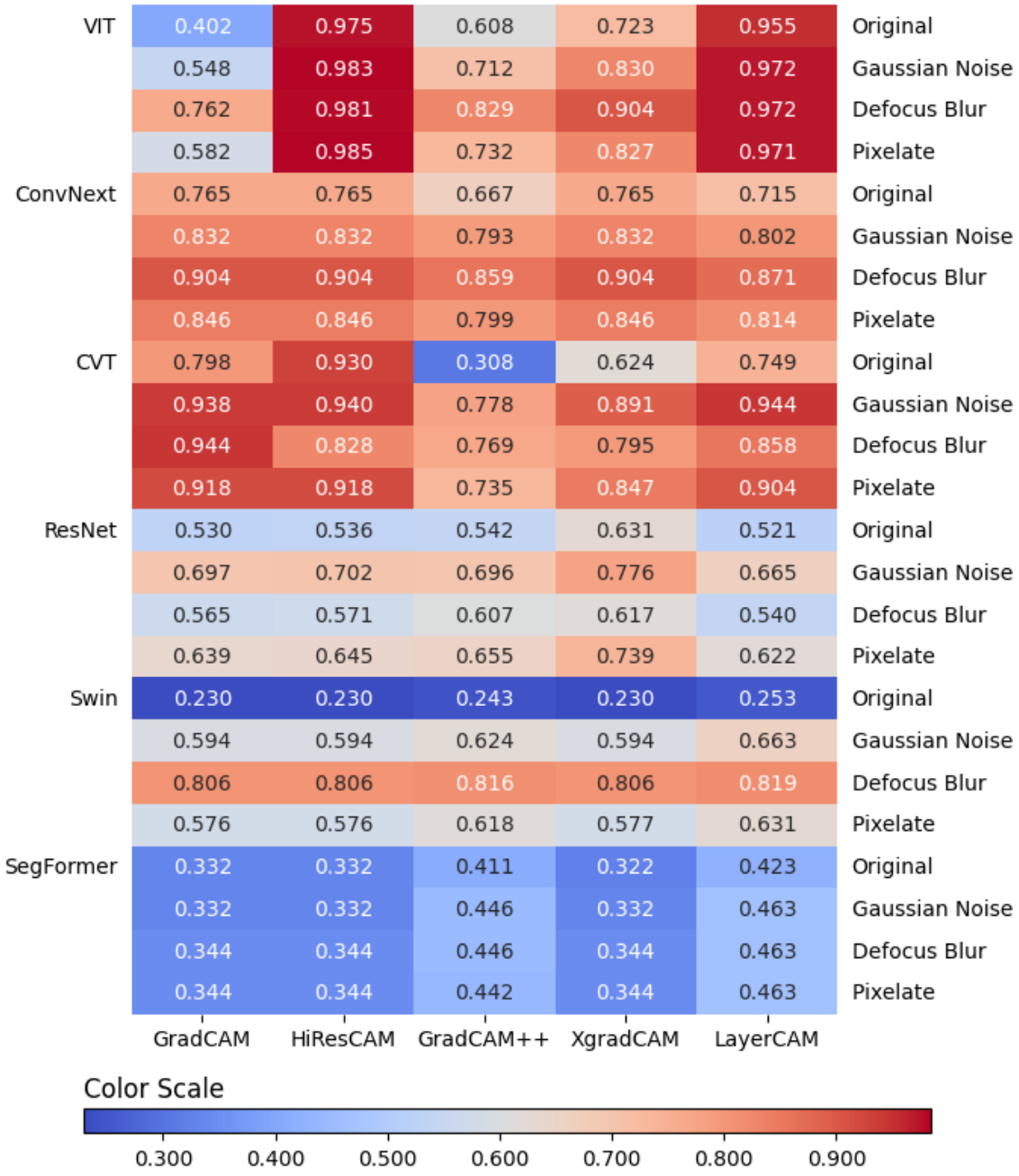


Figure 25: Heatmaps Illustrating Median Prediction Change Percentage for Original and Adversarial Perturbed Images. Lower Values Indicate Better Explanation Deviation.

Explanation Resilience Analysis: The impact of adversarial perturbations in XAI scenarios remains an under-explored question. This analysis seeks to determine if adversarial perturbations contribute to misleading explanations.

Figure 25 shows the prediction change percentages under adversarial perturbations, which are used to calculate explanation resilience attributes. Additionally, Table 9 offers a concise summary of the explanation resilience results. The smaller the resilience value, the better the explanation can resist the impact of adversarial attacks.

Table 9: Summary of Orginal Deviation, Adversarial Deviation and Explanation Resilience for Vision Models

Model	Original	Adversarial	Resilience
VIT	0.267	0.160	0.107
ConvNext	0.265	0.155	0.110
CVT	0.310	0.133	0.177
ResNet	0.448	0.351	0.097
Swin	0.763	0.327	**0.436**
SegFormer	0.636	0.614	0.022

The analysis shows that the explanation resilience attribute does not align with performance and robustness attributes. The findings reveal that models, such as Swin Transforme, retain consistent explanation deviation, whereas show a significant decrease when encountering adversarial perturbations.

The Assessment Scenario of Tabular Models

In the case of structured data, we present the experimental results for all quality attributes.

Process Configuration: Our study evaluates tabular models using datasets from various domains. COMPAS [111] Recidivism Risk Score Data and Analysis dataset is instrumental in assessing the predictive accuracy of recidivism, offering a rich source of sensitive real-world data. RT-IoT2022 [160], the 2022 Real-Time Internet of Things (IoT) dataset [160] for cybersecurity threats case, is a collection of network traffic data. PriceRunner [4]

Product Classification and Clustering dataset provides a scenario in the e-commerce domain.

The TabTransformer [86], TabNet [10], and FT Transformers [70] are employed as the models for tabular data. The TabTransformer [86], a model inspired by the Transformer architecture, is adapted for tabular data by encoding categorical features into embeddings. TabNet [10] utilizes sequential attention to choose features for each decision step. The FT Transformer [70] optimizes the transformer model to handle numerical features. In terms of XAI methods, Mean Centroid Prediff [115] and SHAP [126] are applied.

Computational Cost Analysis: For the computational cost analysis, we test the time and energy consumption for the three model inference, Mean centroid prediff [115], and SHAP [126] method. Table 10 presents the recorded time and energy consumption for every thousand samples using the algorithms.

Table 10: Analysis of Computational Costs for Tabular Models Based on per Thousand Rows

Model and XAI	Time (s)	Energy (Wh)
TabTransformer	16.04	0.49
TabNet	19.39	0.56
FT Transformers	15.62	0.47
Mean Centroid Prediff	1112.28	53.87
SHAP	768.13	26.74

The costs for the three transformer-based tabular models are comparable. However, XAI takes significantly higher computational costs than model inference. It is observed that the Mean Centroid Prediff [115] has higher time and energy consumption compared to SHAP [126].

Model Performance and Robustness Analysis: We evaluated the performance of various transformer-based tabular models [10, 70, 86] against adversarial perturbations. The performance metrics before and after the adversarial perturbations are shown in Table 11.

The adversarial perturbation causes the model performance to decrease. Among the

Table 11: Analysis of Model Performance and XAI Deviation in Tabular Scenarios

Dataset	Model	Model Performance		XAI Deviation	
		Original	Adv. Changes	Original	Adv. Changes
COMPAS	TabTransformer	0.683	-0.118	0.965	-0.105
	TabNet	0.674	-0.114	0.989	-0.087
	FT Transformers	0.690	-0.103	0.965	-0.071
RT-IoT	TabTransformer	0.921	-0.435	0.987	-0.060
	TabNet	0.883	-0.364	0.985	-0.078
	FT Transformers	0.950	-0.603	0.984	-0.088
PriceRunner	TabTransformer	0.994	-0.175	0.977	-0.063
	TabNet	0.991	-0.179	0.973	-0.062
	FT Transformers	0.997	-0.174	0.973	-0.049

tests, the performance results decrease remarkably, especially for the RT-IoT dataset, which primarily consists of numerical data derived from sensors.

Explanation Deviation and Resilience Analysis: We employ the Mean Centroid Prediff [115] and SHAP [126] methodologies to calculate feature importance values. These explanation deviations are also shown in Table 11. The average adversarial decrement in explanation deviation for Mean Centroid Prediff stands at 0.047, in contrast to a 0.099 reduction for SHAP. This reveals that Mean Centroid Prediff exhibits superior resilience compared to SHAP; however, it comes at the cost of increased computational demands.

Conclusions and Insights

The five quality attributes are designed to comprehensively evaluate the open-source models. Figure 26 presents a radar chart visualizing the comparative analysis of selected vision and tabular models across the quality attributes. It is noted that the chart's values have been normalized, where each value closer to one indicates better the attributes. This evaluation aims to support engineers in selecting and developing explainable models in AI-enabled software.

The evaluation results show variability in the explainability metrics of XAI methods

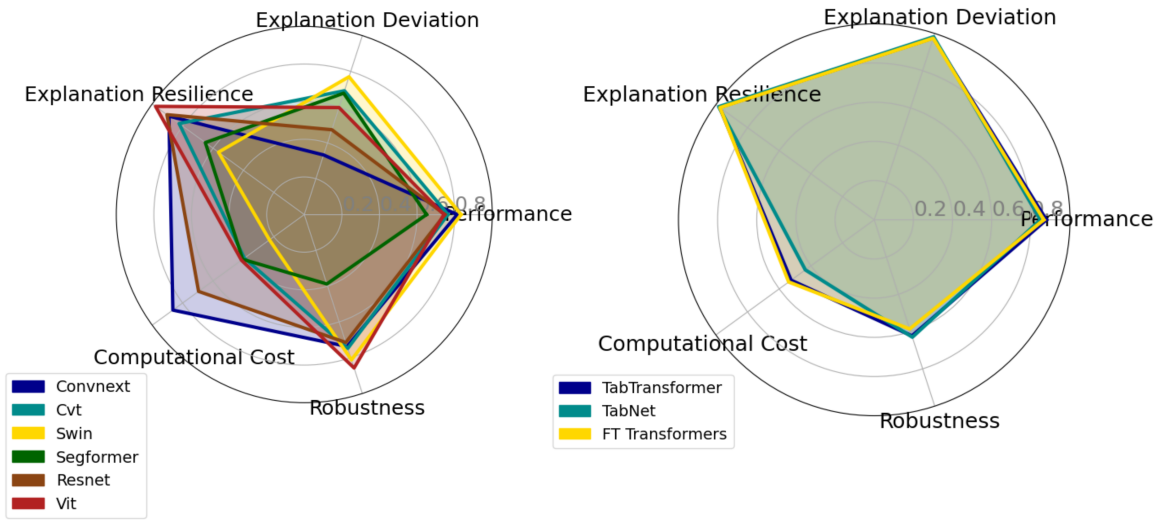


Figure 26: Comprehensive Overview of Multiple Quality Attributes Assessment for the Selected Models.

across various models. The scenario studies encompassed various models, ranging from CNN-based to Transformer-based and from vision to tabular data. When combined with various models, XAI methods offer diverse explanation utilities, as detailed in Figure 25. This variability highlights the importance of assessing compatibility between models and XAI methods. This study introduces a cloud-based XAI service that automates the evaluation pipeline, thereby streamlining the assessment of explainability metrics across diverse models and methods. By providing ease of customization, this service enables microservice to be reusable and the pipeline reproducible.

The investigation into the computational costs of XAI shows a notable burden, especially when compared to the costs for model inferences. The experimental XAI pipelines, as summarized in Table 12, show a correlation between a model's explanation deviation and its computational demands.

The Swin Transformer model, offering the highest model performance and explanation deviation, also incurs the highest computational costs, as indicated by processing times and energy consumption metrics. This correlation showcases a trade-off between achieving

Table 12: Explanation Deviation and Energy Consumption of the Selected Models

Models(Vision/Tabular)	**Explanation Deviation ↓**	**Energy (Wh) ↓**
Swin	0.770	10.07
CVT	0.692	5.86
SegFormer	0.678	5.73
ViT	0.598	5.58
ResNet	0.474	3.32
ConvNeXt	0.333	2.67
TabNet	0.983	0.56
TabTransformer	0.977	0.49
FT Transformers	0.974	0.47

desirable levels of explanation deviation and the associated computational costs, suggesting the need to evaluate the model-XAI combination and optimize this balance.

4.2.6 Summary

This study proposes an XAI service framework designed to streamline operational complexities in XAI evaluation. The framework provides API and SDK interfaces for operation and enables deployment on cloud platforms. It facilitates data preprocessing, encompassing the processing of datasets, data transformations, and the application of adversarial perturbations. The service encapsulates AI models and XAI methods, enabling flexible combinations arranged by the task configuration. We develop and implement evaluation pipelines to assess five key quality attributes: computational cost, performance, robustness, XAI deviation, and XAI resilience.

The findings lead us to conclude with three research questions. First, we observed the variability in metrics results across models and XAI methods. This shows the necessity of evaluation before selecting the XAI method for AI models. This ensures effective explanations are provided to stakeholders. Second, our analysis summarizes the results of computational cost and the explanation metric. High explanation deviation often requires

the expense of increased computational resources. Finally, we demonstrate that adversarial perturbations affect both the model and the XAI, thereby emphasizing the importance of incorporating robust model and XAI methods. These multidimensional quality attributes guide researchers and practitioners in making informed decisions in AI-based software development and deployment.

4.3 Automated Orchestration of Explainable AI Services for Cloud AI and Open-source Models

The widespread adoption of cloud-based AI services and open-source models has enhanced software access to advanced AI capabilities. Leading cloud platforms, including Microsoft Azure, Google Cloud Platform, and Amazon Web Services, offer sophisticated AI functionalities without requiring users to manage model training or computational resources. Similarly, open-source repositories such as HuggingFace offer an array of pre-trained models for diverse AI tasks. However, a significant challenge persists: While these services and models provide valuable predictions, they often operate as black boxes, not explaining their decision-making processes [85].

While these AI capabilities offer powerful functionality, the lack of explainability raises significant concerns, particularly when AI systems are deployed in sensitive software service domains. This limitation creates significant barriers to the broader adoption of AI in software applications [65]. Without clear explanations for AI decisions, software engineers struggle to evaluate models and ensure compliance with regulatory requirements [20]. Recent legislative initiatives underscore the growing importance of explainable AI (XAI). The European Union's AI Act [53] proposes strict requirements for AI systems, including the

need for transparent and explainable AI systems. In the United States, the National Institute of Standards and Technology (NIST) has published an AI Risk Management Framework [142], which requires the practitioner to evaluate XAI. Canada's Directive on Automated Decision-Making [71] mandates that government institutions provide meaningful explanations to affected individuals about the decisions of employed AI. As these regulations continue to evolve, software engineers and AI practitioners are motivated to proactively integrate XAI into their AI systems to ensure public trust. Our proposed service addresses this pressing need by implementing XAI methods across diverse cloud platforms, AI services, and pre-trained models.

However, operationalizing XAI in real-world software systems is non-trivial. AI engineers often grapple with a fragmented landscape of XAI tools, each with its own integration requirements and a lack of standardized interfaces for diverse AI models and platforms. This complexity creates a significant barrier to efficiently embedding explainability into the AI development lifecycle.

Problem Statement. In the context of AI-based software development, the core challenges our service addresses are: (1) the lack of an end-to-end solution from data processing to explanation; (2) the complexity in integrating XAI algorithms with diverse online AI platforms, where services are often only accessible through APIs; (3) the absence of unified access to a wide range of XAI methods and algorithms; (4) the need for quantifiable metrics for XAI results to guide further optimization; and (5) the lack of reproducibility in XAI operations, which hinders consistent evaluation and improvement of explainability methods. These challenges collectively limit the integration of explainable AI into software engineering practices, impeding the responsible development and deployment of AI technologies.

Contributions. To address these challenges, we introduce the XAIpipeline service

with key functionalities: (1) Unified Open APIs: The XAIpipeline service provides a standardized approach to apply XAI techniques across diverse AI models and platforms. (2) Multi-Service Integration: The XAIpipeline interfaces with major cloud AI services and open-source model repositories. (3) Quantifiable XAI Results: The XAIpipeline generates evaluations and robustness measurements for model optimization. (4) Parallel Processing: The XAIpipeline supports the parallel execution of XAI methods across different tasks, enhancing efficiency. (5) Automated Execution: The XAIpipeline streamlines the XAI process through configurable, automated pipelines. (6) Reproducibility: We ensure XAI operations are recorded and reproducible through provenance configurations.

The versatility of our solution is evident in its application to different AI models spanning tabular data, text, image, and video domains. Currently, the service includes five XAI methods for tabular models [85, 198]; two methods for text models in classification [115]; five CAM-based methods [198] for image vision models; and three methods for video vision models [199]. The source code and task pipelines are available in a publicly accessible GitHub repository for reproducibility. Our extensive scenario studies demonstrate the operational improvements of our XAI service across various domains. In the experiment section, we showcase three scenarios, which are Tabular data classification AutoML service, Cloud Vision AI service, and Video Action Recognition analysis. We have achieved integration with three major cloud AI services, including Google Vertex AI, Microsoft Azure, and Amazon Rekognition, enabling XAI for both proprietary and open-source AI solutions. Our service streamlines XAI operation processes, supports diverse XAI methods for AI applications, and improves the cloud model's performance and explainability metrics. By providing software engineers with configuration-driven automated XAI capabilities, our service contributes to the responsible deployment of AI technologies across industries.

4.3.1 Limitations in Current XAI Cloud Service and Integration Workflow

XAI methodologies can be broadly categorized into model-agnostic and model-specific approaches [85]. Model-agnostic techniques, such as SHAP [126] and LIME [154], provide feature importance scores and can be applied to any machine learning model. Model-specific methods, particularly in computer vision, include various CAM-based techniques [168] that generate visual heatmaps highlighting regions crucial for model decisions. Recent advancements have also introduced an XAI tools collection [215], expanding the methods available to practitioners. Despite these developments, the integration of these methods into practical applications remains challenging. With diverse XAI methods requiring different data preprocessing steps and parameter configurations, the operation pipeline becomes complex to manage [85]. There is a need for a unified, automated approach to streamline XAI implementation and management.

The integration of these XAI methods into AI services has become increasingly important. Major cloud providers are incorporating XAI features into their offerings. Google Cloud's Vertex AI provides explainability features, including Sampled Shapley and Integrated Gradients. Similarly, Microsoft's Azure Machine Learning Interpretability only offers capabilities within its ecosystem. However, these cloud-based solutions are typically restricted to their specific service tasks, limiting their applicability to external models and services.

In software engineering, practitioners face challenges when integrating XAI into systems. The manual implementation of XAI techniques often requires substantial effort to incorporate XAI algorithms with various AI services, introducing operational complexities [85, 143] that can lead to inconsistencies and inefficiencies in the development process. Furthermore, the field lacks standardized evaluation methodologies for XAI results, making it challenging to compare explanations across different models, methods, and adversarial

conditions [198]. XAI operations should be transparent and reproducible to ensure that explanations can be consistently obtained, verified, and interpreted by various stakeholders.

Recent advances in machine learning operations (MLOps) platforms have attempted to address workflow automation challenges, yet they exhibit significant limitations for XAI integration. MLflow [222] provides experiment tracking and model management but does not offer native XAI integration. TensorFlow Extended (TFX) [16] provides production ML pipelines but primarily supports Google's ecosystem without cross-platform integration, contradicting the need for unified access across cloud services and open-source models demonstrated in our experiments. Weights & Biases [21] offers experiment tracking with limited explainability features, yet these remain pre-training and lack the evaluation metrics such as consistency, fidelity, and robustness that our service provides.

The limitations of current XAI solutions highlight the necessity for an approach that can apply XAI techniques across different platforms, provide standardized evaluation methodologies, and integrate XAI into software development. These gaps limit the responsible development and deployment of AI models in public software. Our XAIpipeline service addresses these fundamental limitations by offering a dedicated, platform-agnostic solution that seamlessly integrates diverse XAI methods with standardized evaluation across cloud services and open-source models.

4.3.2 Configuration Driven XAI Service

XAIpipeline service advances the implementation of explainable AI by automating the orchestration of diverse computing entities. Through a unified interface, it integrates data preprocessing, self-host AI models, cloud AI services, and XAI methods while efficiently managing data flow between components. This approach empowers AI engineers and software architects to focus on pipeline configuration, significantly reducing the implementation's overhead. Moreover, the service's modular architecture facilitates the integration of

new components and methods [201], ensuring adaptability to evolving XAI techniques and software engineering requirements.

API-Centric Architecture

The service's core architecture is API-centric. This approach aligns with current practices and ecosystems in AI services [113]. By promoting modularity, the service allows for independent component updates, which in turn facilitate integration with existing AI services and models. This approach also enables the easy incorporation of new XAI methods without disrupting the current system. Moreover, it maintains a clear separation between AI development and application logic, thereby improving the overall maintainability and flexibility of the system. These features collectively contribute to a robust and adaptable service that can evolve with the rapidly advancing field of explainable AI.

The XAIpipeline service implements a multi-faceted interface layer: a RESTful API that adheres to OpenAPI 3.0 specifications, a Command-Line Interface (CLI) for streamlined configuration-based execution, and a Web Portal for comprehensive service monitoring and visualization of explanation results. We illustrate a sample configuration file in Figure 27 that drives the pipeline execution. This declarative approach encapsulates necessary parameters for end-to-end operation, including data source specifications, model access credentials, XAI method selection, and evaluation criteria. The configuration defines cloud storage locations for datasets and specifies the essential parameters required to authenticate and query cloud AI services. This configuration-driven architecture enables reproducible execution while abstracting the underlying complexity of cross-platform integration.

Components and Workflow

We illustrate the XAIpipeline service's components in Figure 28. To provide explainability for AI-integrated software services, we implement an interface layer for users. Under

```
{"dataprocess_config": {
    "server_url": "http://xaiport.ddns.net:8001",
    "datasets": {
      "t1204": {
        "storage_address": "S3://xai-datasets/imagenet/images10k/",
        "data_type": ["Image"]
      }}},
  "model_config": {
    "base_url": "http://xaiport.ddns.net:8002",
    "models": {
      "t1204": {
        "model_version": "google_vertex_testxai",
        "platform": "GCP",
        "project": "627640618913",
        "endpoint_id": "8089666696966569984",
        "location": "us-central1"
      }}}}
```

Figure 27: Sample JSON template for XAI pipeline configuration

the interface layer, the XAIpipeline service consists of several key components that work together to provide comprehensive explainability.

At its foundation, the Data Processing Service manages dataset upload and organization within cloud storage solutions while also handling data transformation to fulfill the models' input requirements, data augmentation to enhance model performance, and adversarial perturbation to examine model robustness.

The Model Service component implements a unified interface for cloud AI services and open-source models, ensuring consistent interactions to compute explanations [85]. Software services utilize either cloud-based AI services or self-deployed models. This service provides the API requests for the downstream process, including prediction queries and gradient requests. This integration is essential for cloud AI services where direct access to model internals is restricted, as it maintains a consistent interaction pattern across various model types and hosting platforms.

The XAI Methods Service component implements XAI algorithms and external tools as containerized services, allowing the integration of custom algorithms that conform to

the service's API. XAI tools and frameworks, such as Sampled Shapley [126] and OmniXAI [215], can be employed. Users select XAI methods compatible with their specific model type and task requirements. The service supports a range of analysis types, including tabular model [85, 115], image model [198], text model [172], and video model [199]. This component enables users to access diverse methodologies without managing complex switch operations.

Lastly, the Evaluation Service implements comprehensive metrics such as consistency [85], fidelity [199], and robustness [198] to analyze the quality of explanation results. This evaluation capability allows users to compare different XAI techniques and select the most appropriate XAI approach for their specific application context.

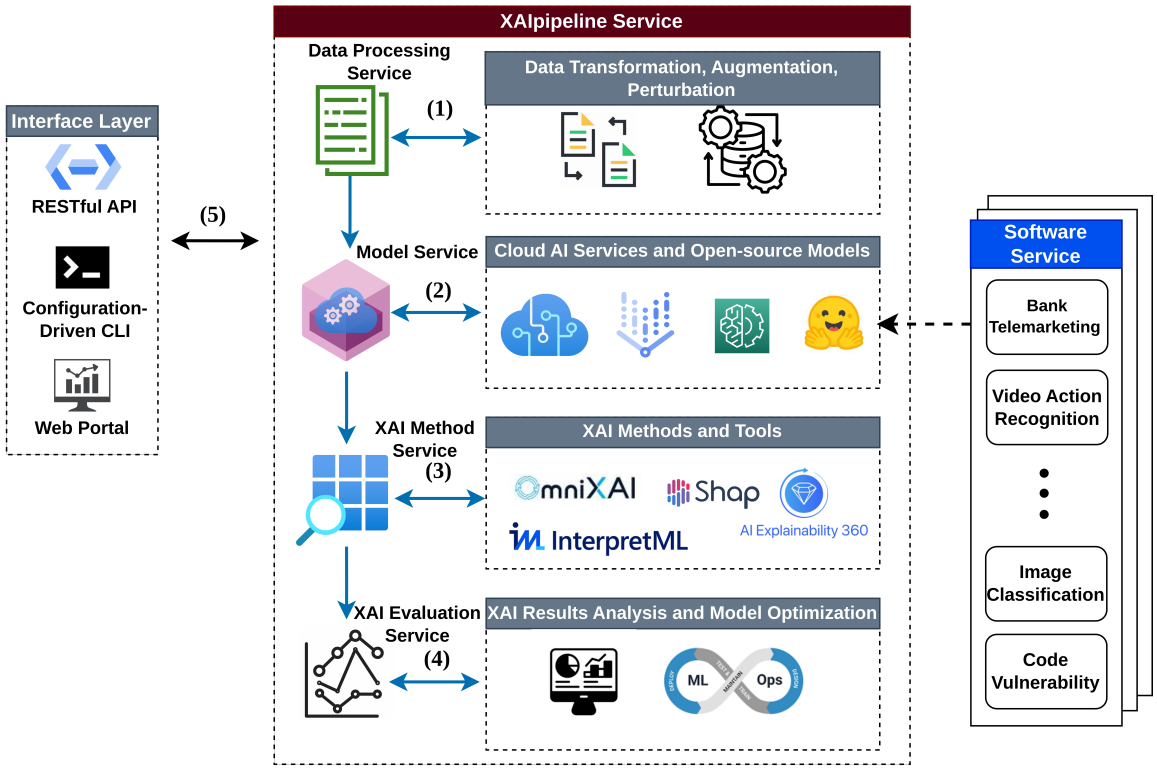


Figure 28: XAIpipeline service components. (1) data processing service, including augmentation and perturbation; (2) model configuration, whether cloud-based or local model is integrated into the server; (3) XAI method and parameter setting, external tools can be configured; (4) specification of evaluation metrics; and (5) the pipeline execution.

The service's workflow is highly configurable through the specification file, enabling users to tune the parameters to their specific needs. Users can select from a range of pre-implemented XAI methods for specific tasks. Additionally, the system supports the integration of custom XAI methods, provided they conform to the service's API specifications. Furthermore, the modular design facilitates continuous improvement, allowing users to update or extend functionalities by modifying the code. This approach ensures that the service remains adaptable to new developments in XAI research and can meet the changing needs in various domains.

Addressing XAI Challenges

The XAIpipeline service addresses five core XAI operational challenges through the components illustrated in Figure 28: (1) XAI Pipeline Management: Automating execution through configurable pipelines that streamline end-to-end workflows; (2) Integration Complexity: Seamlessly connecting with cloud AI services and open-source model repositories through standardized interfaces; (3) XAI Method Adaptability: Implementing a unified interface that enables consistent application of diverse XAI techniques across AI platforms; (4) Result Quantification: Generating both qualitative explanations and quantitative robustness measurements to support comprehensive analysis; and (5) Reproducibility: Recording comprehensive operational data to ensure consistent and verifiable results.

To achieve these capabilities, the service employs containerization technology for enhanced portability and scalability across computing environments. All components are architected as microservices, enabling independent scaling based on computational demands. Extensibility is further supported through documented API specifications that allow researchers and developers to incorporate custom XAI methods.

4.3.3 Service Integration and Application Scenarios

This subsection presents three case studies demonstrating the effectiveness of our XAIpipeline service across diverse domains. These experiments include tabular data analysis, cloud vision AI service, and video recognition, showcasing the framework's ability to integrate and standardize XAI methods across heterogeneous data types and models. Additionally, our GitHub repository contains supplementary studies on text-based code vulnerability detection and a series of open-source vision models, extending the framework's application range. The source code and supplementary experimental results are available in a publicly accessible repository for reproducibility.

Explaining Tabular Data Models

This scenario explains tabular models by the Google Vertex AI service. We also compare Google Cloud Platform's built-in XAI options with our XAIpipeline service.

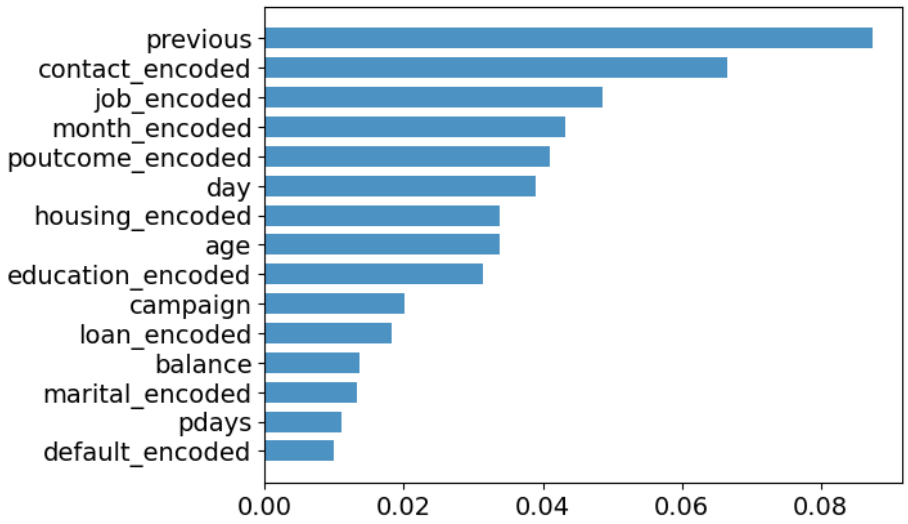


Figure 29: Feature importance results from Google Vertex Explainable AI. This bar chart ranks features by their influence on model predictions. Higher values indicate features with a greater impact on the model's decision-making process.

Dataset and Models. We employed the Bank Marketing dataset [139], which contains

client data from a Portuguese banking institution and includes demographic details, financial information, and previous interactions. The prediction task involves predicting whether clients will subscribe to a term deposit. We upload the dataset to the Google Cloud Bucket. Then, Google Vertex AI's AutoML service automatically optimizes model selection and hyperparameters. The budget is set to one node hour. However, the real training time is 2 hours and 10 minutes. The model precision has achieved 90.6%.

XAI Methods. We compare the Google Cloud Platform's Vertex Explainable AI with our XAIpipeline service. The platform claims to include the XAI function. However, the service only provides an overall feature importance with the Shapley method [126]. In this case, we also select the Shapley method for XAIpipeline services.

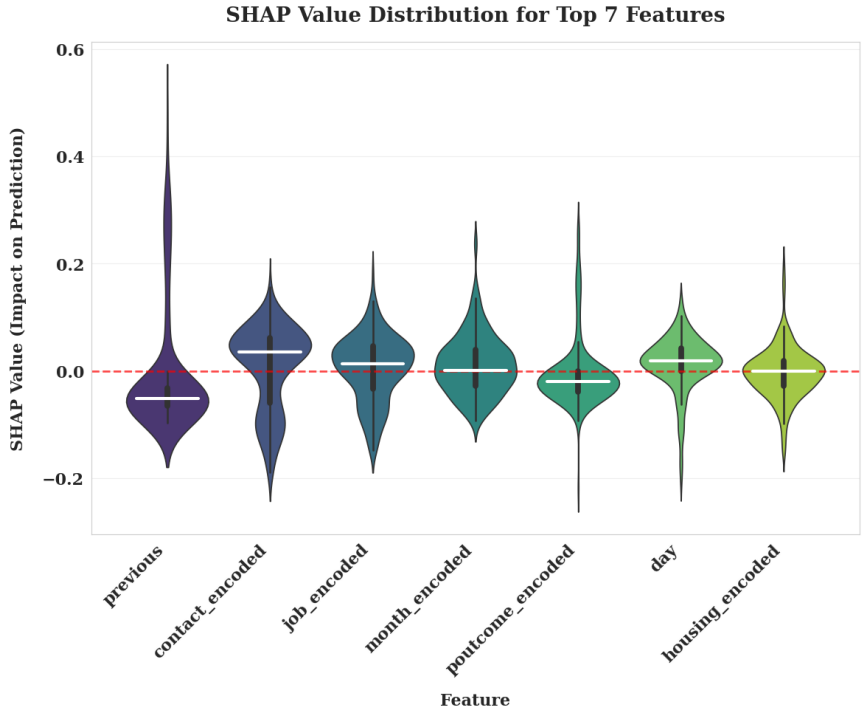


Figure 30: Feature importance results from XAIpipeline's Sampled Shapley method. The violin plots illustrate the distribution and impact magnitude of the top seven selected features across two hundred samples, with median values marked. Positive values indicate features that increase prediction probability, while negative values decrease it.

Experimental Results. As shown in Figure 29, the Vertex Explainable AI only adopts

Sampled Shapley Value [126] as the global explanation method. Further details, local explanations, and evaluation metrics are not provided or accessible. In our XAIpipeline, there are multiple Tabular XAI methods available, including LIME [154], Prediff [115], and so on. Here, we also plot a dataset of the Shapley Value method, as shown in Figure 30. The XAIpipeline service generated feature importance values for each sample, providing insights into local explanation. Compared with Figure 29, which is an overall feature importance ranking, our method, Figure 30, further illustrates the distribution of Shapley values across two hundred individual data samples. The violin plot summarized the feature importance for all of the samples and visualized the distribution.

Our analysis demonstrated that the XAIpipeline service can provide both global feature importance and instance-level feature contributions, enabling users to understand how the model processes tabular data on each data sample. Further statistical analysis is also flexibly accessible; however, the cloud service provider limits the explanation results to only one rough output. Our flexible approach to XAI provides more transparent AI software services and supports quality assurance.

Explaining Temporal Feature in Open-source Model for Video Action Recognition

Our third scenario demonstrates the XAIpipeline's capability to integrate advanced video explanation methods for analyzing temporal feature importance.

Dataset and Models. For video action recognition analysis, we utilized the Kinetics-400 dataset [95], a collection of video clips covering 400 human action classes. The TimeSformer model [19] is an advanced Transformer-architecture video model specifically designed for action recognition.

XAI Methods. To explain video models, we integrate a spatio-temporal XAI method, STAA [199], which provides a video-level saliency map explanation using the attention mechanism. Different from previous saliency map spatial explanation [168], the STAA

[199] method provides a temporal explanation as well. Therefore, the video model's explanation can be a saliency video with the model's spatial attention and temporal attention.

Experimental Results. We illustrate the temporal importance values in the line charts in Figure 31. The XAIpipeline service orchestrated the generation of temporal importance across one hundred video samples, interpreting how the TimeSformer model [19] distributes attention across time for different action categories. The analysis demonstrated clear temporal reasoning patterns. For instance, in jumping videos, the model exhibited minimal attention during preparatory frames but showed significantly heightened activation precisely when the subject became airborne [199]. To the best of our knowledge, the video explanation is not included in the previous XAI framework or services [201].

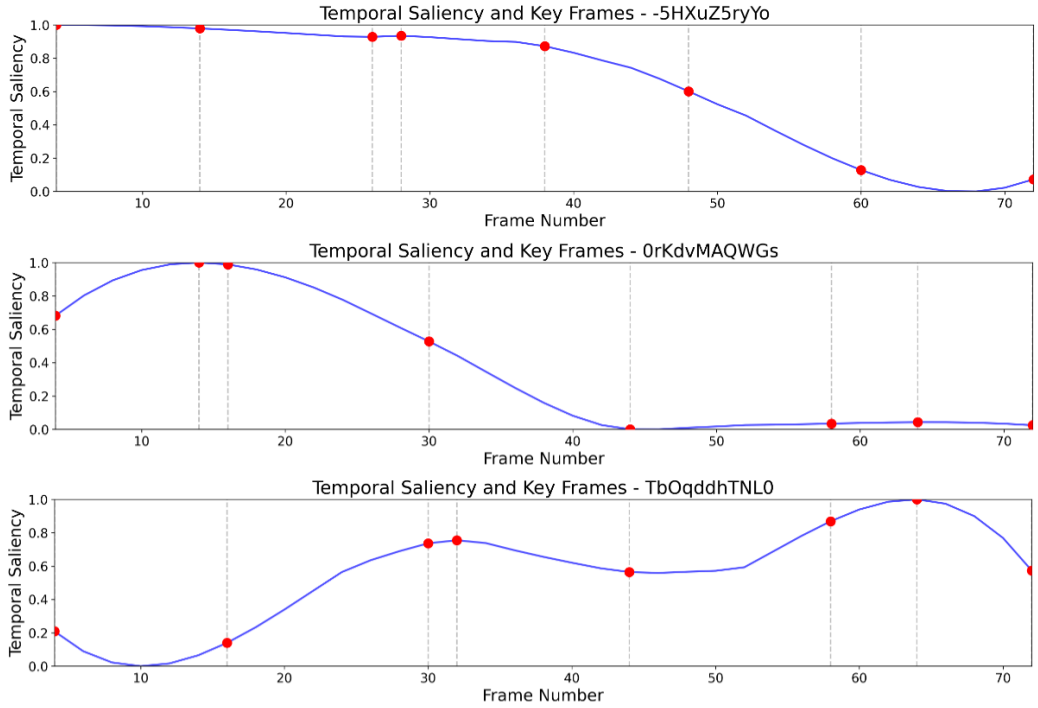


Figure 31: Temporal feature importance across video frames for three different videos. Each subplot displays how the model's attention varies throughout the temporal dimension, with the x-axis representing frame indices and the y-axis indicating importance scores.

Overall, these experiments demonstrate the versatility of our XAIpipeline service. By integrating multiple XAI methods with diverse model architectures, the service provides

explainable insights into model decision-making. Compared with the Google Vertex XAI, our XAI pipeline service provides more detailed local explanations and flexible statistical analysis. For cloud vision services, we use the XAIpipeline to test the data augmentation techniques that improve both model performance and explainability metrics. The recent spatiotemporal XAI method revealed how video models dynamically focus attention based on action-relevant frames, which is not provided in previous services or frameworks, such as Google Vertex XAI or OmniXAI [215]. XAIpipeline works across cloud services and open-source models through a unified, automated approach that eliminates the need for platform-specific implementations.

Service Computational Overhead Analysis

The adoption of XAI introduces additional computational overheads. From Figure 32, we can review the ratio for time consumption from various operation components for three cloud vision AI services.

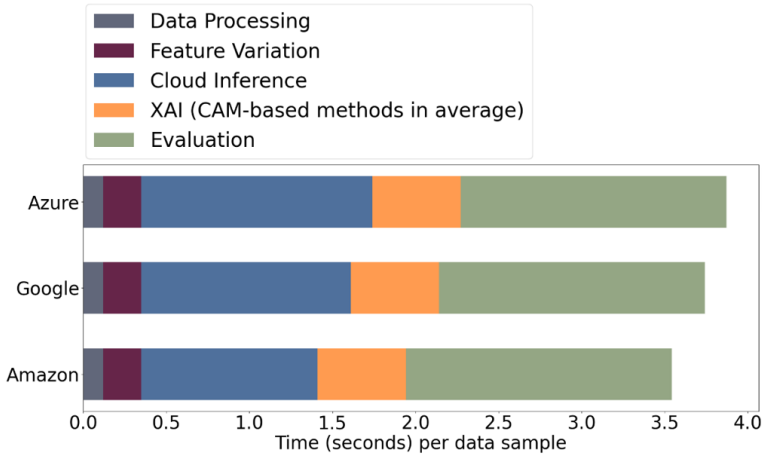


Figure 32: Measured average time consumption from XAIpipeline's service components for each image data sample.

The data processing and XAI times are fairly consistent across cloud services since the algorithm remains the same. XAI evaluations account for a significant portion of the

total operational time. The model inference time varies by cloud providers. Besides, we integrate a tracking function [106, 202] to monitor the use of computer resources, energy consumption, and the carbon footprint of operations. It underscores the necessity of considering trade-offs when selecting XAI methods, especially regarding computational demand, energy consumption, and associated costs.

4.3.4 Summary

This work presents an XAI service that addresses key challenges in applying explainable AI to cloud-based services and open-source models. The service's unified interface layer provides standardized access to XAI techniques across AI platforms, offering consistent explanations for black-box models. Its API-centric architecture integrates with diverse AI services and repositories, while a configurable pipeline system manages XAI methods across various tasks and data types. The service generates both qualitative explanations and quantitative measurements, supporting model improvement and ensuring reproducibility through comprehensive data recording.

This approach reduces the complexity of implementing XAI methods, enabling software engineers to incorporate explainability throughout the AI service lifecycle. Future work will focus on integrating advanced XAI techniques and exploring AutoML pipeline integration. By streamlining explainable AI integration in software engineering workflows, this service aims to accelerate XAI adoption, addressing demands for transparent and trustworthy AI-enabled software.

Chapter 5

XAI Methods and Adversarial Attacks for Spatio-Temporal Video Model

Transformer-based video models have performed SOTA in various action recognition and video understanding tasks. The limitation is the lack of explainability of these complex models. Current Explainable AI (XAI) methods focus on feature importance analysis of only one dimension, either spatial or temporal features. When applied to transformer-based video models, the challenges are two-fold. They fail to capture the integrated spatio-temporal nature of video data and incur prohibitive computational costs for real-time applications. This work presents STAA (Spatio-Temporal Attention Attribution), an XAI method for interpreting video transformer models for action recognition tasks. STAA simultaneously extracts both spatial importance within frames and temporal relevance across the video sequence from attention values in transformers. This unified approach solves the challenge by capturing how transformer models integrate information across both dimensions for decision-making, providing comprehensive explanations that reflect the model's actual reasoning process. We enhance STAA's raw output through post-processing. The experiments on the Kinetics-400 dataset demonstrate superior faithfulness (0.844 ± 0.116) and monotonicity (0.850 ± 0.030). In terms of computational overhead, our STAA method

is applicable for real-time video XAI analysis applications. We implement a cloud-based architecture that enables video explanations with 150 ms latency.

The widespread deployment of video transformer models in action recognition systems necessitates a comprehensive understanding of their vulnerability to adversarial attacks. Unlike traditional CNN-based video models, transformers process spatiotemporal dependencies through self-attention mechanisms, creating a different vulnerability profile to adversarial attacks. This study presents an investigation of adversarial robustness in video transformers. We develop a novel joint spatiotemporal attack method that precisely targets the attention mechanisms of video transformers. By simultaneously perturbing both spatial and temporal features, our method achieves a 76.30% in ASR on the Kinetics-400 dataset, outperforming frame-wise attacks and state-of-the-art query-based attacks. To interpret the mechanisms underlying these attacks, we introduce quantitative metrics based on Explainable AI (XAI) analysis. Spatial analysis reveals systematic disruption of attention patterns, with adversarial examples showing median SSIM scores of 0.353. Temporal correlation analysis also demonstrates severe degradation in attention coherence across frame sequences. Through experiments comparing previous attack methods, including common corruptions benchmark, frame-wise attacks, sparse attacks, and recent V-BAD attacks, we demonstrate that our proposed method is more effective in transformer-based video models. This study further examines the adversarial training strategy against the selected attacks. To promote reproducibility and facilitate future research, we provide our methods and analysis tools through a public GitHub repository. These findings underscore the effectiveness of jointly considering spatial and temporal dimensions when developing adversarial attack strategies and defense mechanisms for video AI models.

5.1 Spatio-Temporal Attention Attribution for Video Transformer Models

Video understanding has emerged as a cornerstone of modern artificial intelligence, revolutionizing critical applications from autonomous driving safety systems [218] to precision medical diagnostics [119]. The introduction of Transformer architectures, exemplified by TimeSformer [19] and VideoMAE [188], represents a paradigm shift in video analysis, consistently outperforming traditional convolutional neural networks (CNNs) [29, 95] across diverse benchmarks. However, this leap in performance introduces new challenges: as model architectures grow more sophisticated, their decision-making processes become increasingly opaque, necessitating advanced explainable AI (XAI) approaches [145] to ensure accountability and trust in critical applications.

Video understanding presents unique challenges compared to static image analysis, primarily due to its inherent spatio-temporal complexity. While spatial features capture frame-level information such as object configurations and contextual relationships, temporal features encode crucial dynamic patterns—motion trajectories, state transitions, and temporal dependencies [195]. The significance of temporal modeling is evidenced by breakthroughs such as Temporal Segment Networks (TSN) [195], which demonstrate how precise temporal feature analysis enables the discrimination of visually similar but temporally distinct actions. This spatio-temporal interplay forms the foundation of advanced video understanding tasks, where the precise sequencing and timing of spatial transformations define complex actions [29]. Understanding and explaining these intricate spatio-temporal relationships becomes crucial as video AI systems increasingly influence high-stakes decisions in autonomous systems [218] and healthcare [119].

Research Gaps

While significant advances have been made in developing XAI methods for CNN-based image analysis [168], extending these approaches to Transformer-based video models presents unprecedented challenges. These challenges manifest in two critical dimensions: algorithmic complexity and computational efficiency, both of which fundamentally impact the practical deployment of explainable video AI systems.

Algorithmic Challenges. The distinctive architecture of Transformers, centered on self-attention mechanisms, represents a fundamental departure from traditional CNNs, rendering many established XAI methods ineffective or theoretically incompatible [31, 168]. Additionally, architectural divergence is compounded by the inherent complexity of video data, where temporal dynamics create additional complexity beyond what static image-based XAI methods can address [230]. Current approaches often artificially dichotomize spatial and temporal analysis, producing fragmented explanations that fail to capture the holistic nature of video understanding. This limitation creates a critical gap in our ability to explain how models interpret and process the dynamic interplay between spatial and temporal features in video content.

Computational Efficiency Challenges. The computational demands of existing post-hoc explanation methods present a significant barrier to real-world deployment. These methods typically require multiple model inference passes, creating substantial computational overhead for video analysis tasks. This inefficiency becomes particularly acute in scenarios demanding real-time explanations, such as autonomous systems or live video monitoring applications [42]. The computational burden is especially problematic for long video sequences, where the resource requirements can grow exponentially [85, 126] due to the algorithm of post-hoc methods.

These challenges underscore the need for XAI methods to adapt to transformer architectures, provide spatio-temporal explanations, and achieve high computational efficiency.

Therefore, we focus on the following research questions: How to simultaneously capture spatial and temporal feature explanations for video transformer models? What are the cross-cutting quality attributes of the proposed XAI methods in terms of faithfulness, monotonicity, and computational efficiency? What system architecture design enables real-time explanations with hundreds-of-milliseconds level latency?

There are several critical gaps in the current video AI explainability: (1) Despite the widespread deployment of Transformer-based video [11,19,198], there remains an absence of XAI methods for video Transformer architectures [167]. (2) Previous model-agnostic XAI approaches provide a single type of feature explanation, failing to capture both spatial and temporal features in video models simultaneously. (3) The absence of real-time capabilities for video analysis creates a significant barrier for edge computing applications [63]. Traditional post-hoc approaches fail to address the computational demands of video analysis. These gaps collectively impede practitioners from understanding and trusting video AI systems, highlighting the urgent need for novel XAI approaches that can address both the theoretical and practical challenges of explaining video Transformer models.

Our Contributions

To address these critical gaps, we propose STAA (Spatio-Temporal Attention Attribution), a novel XAI method for video Transformer models. STAA achieves three advantages: (1) simultaneous capture of spatial and temporal feature importance through a single forward pass, (2) improved explanation faithfulness through direct alignment with the model's decision-making process, and (3) significantly reduced computational overhead by utilizing the model's internal attention mechanisms rather than requiring multiple inference passes. In this work, the summarized contributions are:

Novel XAI method. STAA fundamentally addresses the method mismatch between current XAI methods and video Transformer models.

Real-Time XAI Framework. Design and implementation of a cloud-based service architecture that enables real-time video explanations with sub-100ms latency, making STAA practical for edge computing applications.

Empirical Validation. Comprehensive evaluation of STAA against SHAP [126], LIME [154], and Transformer Interpretability [31] on multiple datasets [72, 95], showing major improvements in explanation quality and efficiency.

The remainder of this work is organized as follows: Section II reviews related work in video Transformers and current XAI approaches, highlighting limitations in explaining video models. Section III presents our STAA method and adaptations of the general XAI methods for video analysis. Section IV details experimental comparison with the model-agnostic XAI methods. Section V provides the experimental comparison with the model-specific XAI method. Section VI demonstrates STAA's performance improvements in computational efficiency, including our real-time implementation results. Section VII concludes with future directions.

5.1.1 The Challenges for Video XAI

Deep learning has revolutionized video analysis, with Transformer models leading the way [19]. However, their complexity makes understanding their decisions challenging. This subsection explores Transformer-based video models, current XAI methods, and the challenges of explaining video Transformers.

Transformer Models for Video Classification Tasks

The success of Transformer architectures in natural language processing [191] has inspired researchers to adopt them to computer vision tasks [47]. In contrast to traditional CNNs, Transformer-based models can capture long-range dependencies and have shown remarkable performance in various video understanding tasks [95]. TimeSformer [19] pioneered

the application of pure Transformer architectures to video classification by treating different frame patches and time steps as tokens. This approach demonstrated competitive performance on standard benchmarks while being computationally efficient. Building upon this work, ViViT [11] proposed a tubelet embedding scheme, further slightly improving video classification accuracy.

These models have consistently outperformed CNN-based approaches in their respective papers. However, previous XAI methods for generating saliency maps are not applicable to these Transformer-based video analysis models. While methods such as Grad-CAM [168] have been effective in visualizing important regions in CNN-based models, they do not trivially extend to the self-attention mechanisms and token-based representations employed by Transformers. This necessitates the development of novel XAI techniques specifically for Transformer architectures in video understanding tasks. Recent work has begun to develop XAI methods for Transformer models. For instance, the recent work [31] provides XAI for transformer image models. The XAI pipeline [198] includes multiple Transformer image models. Adapting and extending such approaches to video analysis represents a promising direction for research.

Explainable AI Methods in Computer Vision

XAI has gained significant attention in recent years due to the increasing complexity of deep learning models and the need for transparency in AI systems [85]. Various XAI techniques have been developed and applied in the comprehensive XAI service frameworks [201]. SHAP (SHapley Additive exPlanations) [126] provides feature attribution based on coalitional game theory. LIME (Local Interpretable Model-agnostic Explanations) [154] offers another approach by approximating the behavior of complex models with interpretable surrogate models in the vicinity of a specific input, providing insights into local decision boundaries.

Despite the significant progress in XAI techniques for image analysis. These methods face substantial challenges when applied to video models. The temporal feature inherent in video data poses an additional dimension for most existing XAI methods, which are primarily designed for static images and struggle to capture the temporal feature of video content. Furthermore, the architectural differences between Transformers and traditional CNNs create a mismatch, as many XAI techniques rely on gradient maps [168], rendering them incompatible with Transformer models.

The computational demands of video analysis exacerbate these issues, as post-hoc explanation methods often require multiple inferencing runs, which is prohibitively expensive for deep models and long video sequences. The algorithm inefficiency limits their practical applicability in real-world scenarios where timely explanations are crucial [63]. The current methods also fail to provide unified spatio-temporal explanations [126, 154]. This approach fails to capture the intricate interactions between space and time that are fundamental to video data, resulting in incomplete or misleading interpretations of model behavior [177].

These limitations underscore the pressing need for XAI approaches specifically designed to address the challenges posed by video Transformer models. Such methods must handle both spatial and temporal features while maintaining computational efficiency.

Challenges in XAI for Transformer Architecture

While significant progress has been made in developing XAI techniques for image analysis, the interpretability of video models presents unique challenges due to the additional temporal dimension and the complexity of Spatio-Temporal reasoning [177]. For CNN-based video models, Hiley et al. [82] adapted autograd for 3D CNNs to generate class activation maps for action recognition models. Building on this work, Stergiou et al. [178] proposed Class-Specific Saliency Tubes (CSST), which extend 2D saliency maps to the temporal

domain, providing frame-wise visualizations of important regions for action recognition. However, their technique routes are not applicable to Transformer-based video models due to their fundamentally different architectures. To the best of our knowledge, the field of XAI for video Transformers remains unexplored. With the Transformer achieving the performance and being more commonly deployed, lacking XAI becomes particularly concerning given the increasing adoption of these models in applications [75]. We seek to provide deeper insights into their decision-making processes and pave the way for more transparent and trustworthy video analysis systems.

Application Requirements for Real-Time Video XAI

Software engineers define a real-time response as one that occurs within one hundred milliseconds [42]. We refer to this requirement for latency in real-time video analysis. Considering our STAA method reduces the computation time of XAI for a video clip from minutes to milliseconds, it is possible to apply STAA in real-time video XAI analysis.

Real-time video stream model AI has diverse applications across critical industries. In autonomous driving systems [218], real-time video analysis enables immediate detection of road hazards, pedestrians, and changing traffic conditions, allowing for split-second decision-making essential for passenger safety. In medical diagnostics [119], real-time video AI assists surgeons during procedures and enables rapid analysis of endoscopic footage for early disease detection. In security applications [161], these systems continuously monitor surveillance feeds to detect anomalies and identify potential threats, providing immediate alerts to security personnel. These high-stakes applications underscore the importance of immediate and interpretable explanations of AI decisions.

5.1.2 Dataset Specifications and Characteristics

This section provides comprehensive specifications of the video datasets employed in this research, including detailed information about video duration, frame rates, temporal characteristics, and motion complexity.

Kinetics-400 Dataset

The Kinetics-400 dataset [95] serves as the primary benchmark for evaluating our video XAI methods. This section details the dataset's composition, video characteristics, and our experimental configuration.

Dataset Composition and Structure The Kinetics-400 dataset comprises approximately 240,000 video clips collected from YouTube, covering 400 distinct human action classes. These action categories span a diverse range of human activities including sports, musical performances, human interactions, daily activities, and occupational tasks. The dataset represents real-world video diversity with variations in recording conditions, camera angles, video quality, and background complexity.

The 400 action classes are organized into broad categories such as:

- **Sports and Exercise:** Activities including swimming, running, playing various ball games, gymnastics
- **Music and Performance:** Playing musical instruments, dancing, singing
- **Human Interactions:** Handshaking, hugging, high-fiving, fighting
- **Daily Activities:** Cooking, eating, cleaning, personal care
- **Occupational Activities:** Construction work, medical procedures, teaching

Video Temporal Characteristics A critical aspect of video understanding is the temporal dimension. For Kinetics-400, the temporal properties are:

- **Average Video Duration:** Each video clip has an approximate duration of 10 seconds. This duration is standardized during dataset preparation by trimming YouTube videos to capture the core action segment. The 10-second duration provides sufficient temporal context for action recognition while maintaining computational feasibility for deep learning models.
- **Frame Rate:** Original videos are collected at varying frame rates depending on the source recording. We standardize all videos to 30 frames per second (FPS) during preprocessing. This standardization ensures consistent temporal sampling across the dataset and matches the temporal resolution expected by modern video models.
- **Total Frames per Video:** At 30 FPS over 10 seconds, each standardized video contains approximately 300 frames. However, not all frames are processed by the model; see Model Input Configuration below.
- **Temporal Span:** The 10-second clips are selected to contain the complete execution of the target action, from initiation through completion. This temporal completeness is crucial for action recognition, as many activities require observing the full motion sequence for accurate classification.

Spatial Characteristics The spatial properties of Kinetics-400 videos are:

- **Original Resolution:** Videos are sourced from YouTube with variable resolutions ranging from 360p to 1080p or higher, depending on upload quality and time period.
- **Standardized Resolution:** During preprocessing, we resize all videos to 256×256 pixels as an intermediate step, then apply center cropping or random cropping (for

training) to obtain 224×224 pixel frames. This 224×224 resolution is the standard input size for video transformer models including TimeSformer and VideoMAE.

- **Aspect Ratio Handling:** Videos with non-square aspect ratios are first resized such that the shorter side equals 256 pixels, then center-cropped to obtain square frames.

Model Input Configuration The TimeSformer model [19] employed in our experiments processes video input with the following configuration:

- **Frame Sampling:** From each 10-second video (approximately 300 frames), we uniformly sample 8 frames. This sampling strategy ensures that the selected frames span the entire temporal duration of the video.
- **Sampling Formula:** For a video with T total frames, the 8 sampled frame indices are: $\{1, \lfloor T/8 \rfloor, \lfloor 2T/8 \rfloor, ..., \lfloor 7T/8 \rfloor, T\}$.
- **Temporal Coverage:** The 8 frames cover the full 10-second duration with approximately 1.25-second intervals between consecutive sampled frames. This temporal sampling rate is sufficient to capture the motion dynamics of most human actions in the dataset.
- **Spatial Tokenization:** Each 224×224 frame is divided into non-overlapping patches of size 16×16 pixels, resulting in 14×14 = 196 patches per frame. These patches serve as spatial tokens for the transformer.
- **Total Input Tokens:** The complete video input to TimeSformer consists of 8 frames × 196 patches = 1,568 spatial-temporal tokens, plus 1 classification token, for a total of 1,569 tokens.

Data Split for Our Experiments For our XAI evaluation and adversarial robustness experiments, we employ the following data configuration:

- **Pre-trained Model:** We use the publicly available TimeSformer model pre-trained on the Kinetics-400 training set, which achieves 78.0% top-1 accuracy on the official validation set.

- **XAI Evaluation Set:** To ensure balanced representation across all action classes, we randomly sample 30 videos from each of the 400 classes, resulting in a total of 12,000 videos for XAI analysis. This balanced sampling ensures that explanation quality metrics are not biased by class imbalance.

Something-Something V2 Dataset

For additional validation of STAA's generalizability (Section 5.1.3), we employ the Something-Something V2 dataset [72]. This dataset provides a complementary evaluation perspective as it emphasizes temporal reasoning over spatial scene recognition.

Dataset Characteristics

- **Total Videos:** 220,847 video clips across training, validation, and test splits

- **Action Classes:** 174 fine-grained action categories

- **Average Duration:** Videos range from 2 to 6 seconds, with a median duration of approximately 4 seconds

- **Frame Rate:** Original videos are recorded at 12 FPS

- **Key Characteristic:** Actions are primarily temporal in nature and cannot be recognized from a single frame. For example, "Pushing something from left to right" versus "Pushing something from right to left" requires temporal analysis to distinguish.

Model Input Configuration for Something-Something V2

- **Frame Sampling:** We uniformly sample 16 frames from each video (double the sampling rate of Kinetics-400) to capture the finer temporal details required for these temporally-focused actions
- **Spatial Resolution:** 224×224 pixels, consistent with Kinetics-400
- **Videos Used:** We randomly select 2,000 videos from the validation set for STAA evaluation

Dataset Selection Rationale

The selection of Kinetics-400 as the primary evaluation benchmark and Something-Something V2 as the validation dataset is motivated by three key considerations:

Comprehensive Coverage: Kinetics-400's 400 action classes provide broad coverage of human activities, enabling thorough evaluation of STAA across diverse motion patterns, scene contexts, and action complexities. The dataset's widespread adoption in the video understanding community facilitates comparison with prior work and establishes STAA's performance relative to established baselines.

Temporal Diversity: Something-Something V2's emphasis on temporal reasoning provides complementary evaluation beyond Kinetics-400's scene-centric actions. This validation demonstrates that STAA effectively captures fine-grained temporal dynamics required for actions that are temporally distinctive but spatially similar.

Real-World Relevance: Both datasets use real-world videos with natural variations in recording conditions, camera motion, lighting, and background clutter. This realism ensures that our XAI evaluations reflect practical deployment scenarios rather than idealized laboratory conditions.

5.1.3 Methodologies for explaining Video Transformer

Explaining video transformer models presents challenges due to the spatio-temporal dual dimension features. Existing XAI methods are not offered for video models. In this subsection, we extend the previous XAI methods [126, 154] and provide our novel method for video transformer explainability: (1) SHAP-based temporal analysis identifies important temporal segments across video frames, revealing which moments in a video sequence most influence model decisions. (2) LIME-based spatial analysis generates explanations for individual video frames, highlighting important spatial regions within the frame context. (3) STAA (our proposed method) extracts information directly from transformer attention mechanisms to provide unified spatio-temporal explanations, offering an integrated view of both spatial and temporal feature importance within a single framework.

SHAP-based Temporal Feature Attribution

The SHAP (SHapley Additive exPlanations) [126] analysis aims to identify the importance of different temporal segments in a video for the model's decision-making process. This method provides insights into which moments in a video are influential for classification.

Extending SHAP Algorithm for Adaptation to Video Analysis

SHAP applies concepts from coalitional game theory, particularly Shapley values, to feature attribution. In our video analysis, we map these concepts as follows: The temporal segments of the video serve as the "players" in the game-theoretic sense. The "coalitions" are formed by various subsets of these temporal segments. The "characteristic function" is represented by the model's prediction for any given subset of segments. The SHAP value for a segment quantifies its average marginal contribution to the model's prediction across all possible combinations of segments.

To analyse the temporal importance of different parts of a video, we segment the video stream into equal-length temporal segments. SHAP considers all possible combinations

of these segments. Uniform segmentation ensures that each segment represents an equal duration of the video, providing a fair basis for this combinatorial analysis. Practitioners can increase the number of segments to achieve a more fine-grained analysis or decrease it for a more coarse-grained XAI analysis.

Algorithm 1 SHAP-based Temporal Feature Attribution

Notation: D: Duration of video, N: Number of segments, s_i: The i-th temporal segment containing frames from time interval, $f_x(S)$: Model prediction score when using only the temporal segments in set S, f_t: Video frame at time t, ϕ_i: Shapley value for segment i, M_t: Temporal attribution map with N importance scores

Input: D, N

Output: M_t

$s_i = \left\{ f_t \mid t \in \left[\frac{(i-1)D}{N}, \frac{iD}{N} \right) \right\}, \quad i = 1, 2, ..., N$

Initialize $M_t = \{0\}^N$

for $i = 1$ **to** N **do**

$\phi_i = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(N-|S|-1)!}{N!} [f_x(S \cup \{i\}) - f_x(S)]$

$M_t[i] = \phi_i$

end for

return M_t

Temporal Importance Quantification

Algorithm 1 presents our SHAP-based temporal feature attribution method [126]. The algorithm segments the video into N equal-length segments and calculates a SHAP value for each segment to quantify its importance in the model's decision-making process. We have two methods for calculating SHAP values: exact calculation and Monte Carlo approximation. The exact method considers all possible combinations of segments, providing the most accurate results but at a higher computational cost. The Monte Carlo approximation [126] estimates SHAP values by sampling a specified number of random segment combinations, significantly reducing computation time while maintaining an estimate of the true SHAP values.

To illustrate the process of SHAP-based temporal importance analysis, Figure 33 as a flowchart provides a clear understanding of the algorithm's workflow and how it identifies

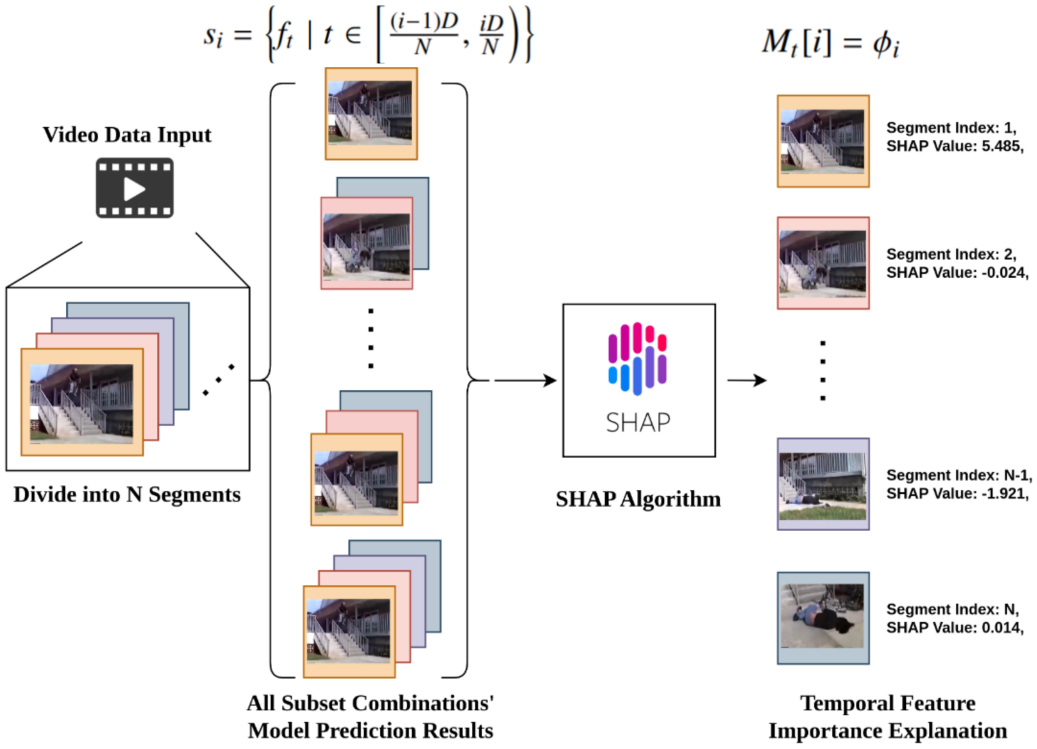


Figure 33: SHAP-based temporal feature attribution method. The process begins with video segmentation into N equal temporal chunks (left), followed by analyzing the contribution of each temporal segment through model predictions with different segment combinations. The method generates temporal importance scores (right) that highlight which time periods in the video are most critical for the model's decision, enabling identification of key moments in actions while ignoring less relevant time segments.

the most influential temporal regions in a video for the model's decision-making process.

Several factors affect the practical implementation and computational efficiency of the SHAP-based method. Segment Size and Temporal Resolution. The choice of N (number of segments) fundamentally determines the granularity of temporal analysis [126]. Smaller segments provide finer temporal resolution but substantially increase computational demands. The computational bottleneck of SHAP implementation lies in the model inference stage. Each segment combination evaluation requires a complete forward pass through the video model, creating a multiplicative relationship between the number of evaluations and

the model's inherent complexity. While SHAP provides valuable temporal feature attributions, its exponential scaling with segment count necessitates considering segmentation parameters.

In summary, SHAP can provide temporal feature attribution through the scores of temporal segments. However, it has several limitations. First, it focuses solely on one feature aspect, according to the segments. Second, the granularity of the analysis is limited by the number of segments N. A small N provides only coarse resolution.

LIME-based Spatial Feature Attribution

LIME (Local Interpretable Model-agnostic Explanations) [154] is employed for spatial analysis of individual video frames. We adapt LIME with the Vision Transformer (ViT) model [47] trained on the Kinetics-400 dataset, enabling it to operate on individual frames while maintaining the model structure in Transformer.

Extending LIME Algorithm for Adaptation to Video Analysis

LIME is grounded in the principle of local approximation. It simplifies complex models by approximating them with interpretable surrogates in the vicinity of specific inputs. For video analysis, we adapt LIME to operate on individual frames extracted at regular intervals from the video sequence. The choice of frame count K involves a trade-off between computational cost and temporal resolution.

For each selected frame, LIME generates perturbed variants by applying random masks to different regions, creating a local neighborhood around the original frame. The interpretable surrogate takes the form of a linear model operating on binary features, representing the presence or absence of interpretable components within the frame. This approach enables LIME to provide localized explanations by highlighting regions that significantly influence the model's frame-level decisions.

Algorithm 2 LIME-based Spatial Feature Attribution

Notation: D: Video duration, K: Number of frames, N_p: Number of perturbations, G: Class of interpretable models, $\mathcal{L}$: Loss function, Ω: Regularization term, $|g_k|$: Absolute weights representing feature importance

Input: D, K, N_p

Output: M_s

$f_k = \mathbf{V}(k \cdot \frac{D}{K}), \quad k = 1, 2, ..., K$ {Extract K frames}

Initialize $M_s = \{0\}^K$ {Array to store importance maps}

for $k = 1$ **to** K **do**

$P_k = \{p_{k1}, \ldots, p_{kN_p}\}$ {Generate perturbations}

$y_{km} = \text{ViT}(p_{km}), \quad m = 1, \ldots, N_p$ {Get predictions}

$g_k = \arg\min_{g \in G} \mathcal{L}(g, \text{ViT}, P_k) + \Omega(g)$ {Fit local model}

$M_s[k] = |g_k|$ {Extract feature importance}

end for

return M_s

Spatial Importance Quantification Algorithm 2 presents the LIME-based spatial feature attribution method for video analysis: The algorithm begins by extracting K equally spaced frames from the input video. For each frame f_k, it generates a set of M perturbed samples P_k by applying random masks. These perturbations help explore the model's behavior in the vicinity of the original frame. Next, the algorithm computes ViT predictions y_{km} for each perturbed sample. An interpretable model g_k is then fitted to approximate the ViT's behavior locally. This process balances between fidelity to the ViT's predictions (represented by the loss function $\mathcal{L}$) and model simplicity (enforced by the regularization term Ω). The algorithm computes importance scores $M_s[k]$ based on the fitted interpretable model for each frame. These scores form a sequence of spatial attributions that can be used to generate the map visualization for each frame, with bluer colors indicating areas of higher importance.

Figure 34 as a flowchart provides a visual representation of the LIME-based spatial feature attribution method. By overlaying these results on the original frames, we can directly associate the importance scores with the visual content of the video, providing an interpretable representation of the model's focus areas. This approach allows us to apply LIME's image analysis capabilities while capturing relevant features for video analysis,

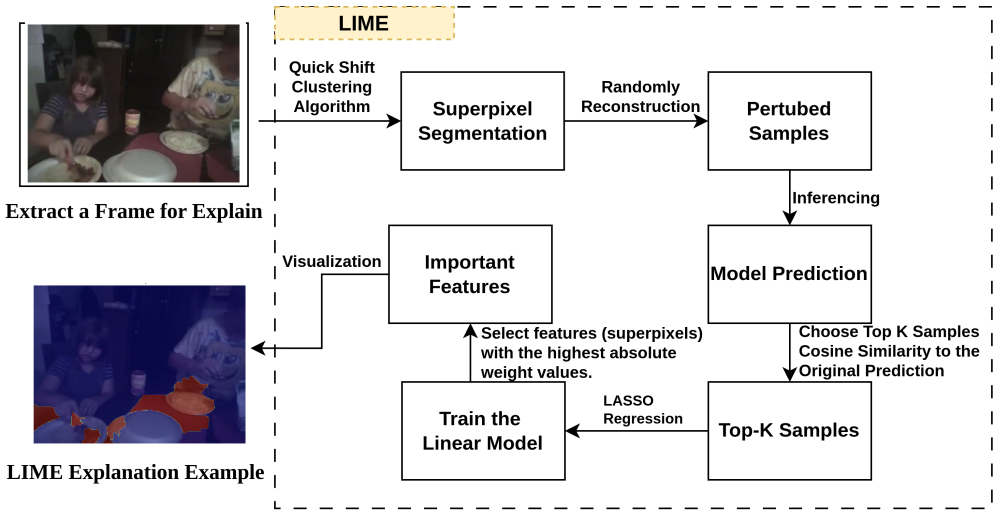


Figure 34: LIME-based spatial feature attribution method: The process extracts K key frames at regular intervals from the video (left-top), then creates perturbed versions of each frame by applying random masks to different spatial regions. The model evaluates these perturbed samples to identify which spatial regions most significantly impact the prediction. The resulting spatial importance maps (left-bottom) highlight action-relevant objects and regions within each frame.

providing insights into which spatial features are most influential in our video Transformer model's decision-making process. This form of explanation is commonly used in vision XAI methods [168, 198, 202] and provides a visual representation of the model's attention mechanisms.

Several factors affect the implementation of LIME in our video analysis context. The number of perturbed samples N_p significantly affects the fidelity of the local approximation. A larger N_p generally leads to more accurate explanations but increases computational cost. In our experiments, we select $N_p = 1000$ for balanced accuracy and efficiency. The number and distribution of frames selected for analysis can affect the understanding of the video. The method of generating perturbations can influence the quality of explanations. We use the default setting in LIME [154], which is random masking. These factors should be considered when applying LIME in video XAI analysis.

LIME provides explanations that are local in nature and may ignore global patterns.

Generating explanations for many frames can be computationally expensive, especially for high-resolution videos. LIME fails in capturing connections between frames.

Spatio-Temporal Attention Attribution (STAA)

This section presents our approach to generating explanations for video Transformer models by analyzing their internal attention mechanisms. By extracting and processing attention values that naturally emerge during model inference, we create interpretable maps that reveal both spatial importance within frames and temporal relevance across the video. This direct method eliminates the need for multiple model passes, significantly reducing computation time while maintaining high faithfulness to the model's actual decision process. We detail the extraction methodology, aggregation techniques, and enhancement methods that together enable meaningful and efficient video explanations. Figure 35 provides a visual overview of the STAA method.

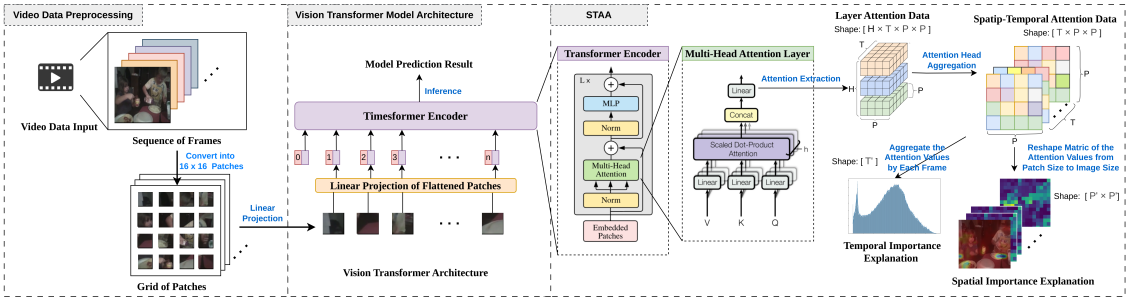


Figure 35: Visualized overview of the Spatio-temporal Attention Attribution (STAA) method for transformer-based video models

Theoretical Basis for Attention as Feature Importance

The use of attention weights as indicators of feature importance is grounded in the fundamental architecture of Transformer models. In these models, attention mechanisms explicitly compute the influence of each input token on output representations through weighted aggregation [191]. When a token receives higher attention weights, it contributes

more significantly to the resulting features used in downstream classification [31]. Differing from gradient-based methods that trace signals backward through the network, attention provides a direct forward measure of how the model distributes its focus across inputs.

Algorithm 3 Spatio-Temporal Attention Attribution (STAA)

Notation:

F: Number of frames in the video clip
H, W: Height and width dimensions of each frame
N: Number of patches per frame (HW/P^2)
P: Patch size (e.g., 16×16 pixels)
A: Number of attention heads in the Transformer model
L: Final layer index of the Transformer model
$\boldsymbol{\alpha}^{(L,a)}$: Attention weights matrix at layer L for head a
$\alpha_{p,t}^{(L,a)}$: Individual attention weight for patch p at frame t
$\mathbf{M}_t$: Temporal attention vector across frames
$\mathbf{M}_s$: Spatial attention matrix for all patches in all frames

Require: Video clip $\mathbf{X} \in \mathbb{R}^{F \times H \times W \times 3}$, Transformer model $\mathcal{M}$
Ensure: Temporal attention map $\mathbf{M}_t \in \mathbb{R}^F$, Spatial attention maps $\mathbf{M}_s \in \mathbb{R}^{N \times F}$

// Extract attention weights from final layer L for all heads
$\{\boldsymbol{\alpha}^{(L,a)}\}_{a=1}^{A} \leftarrow \mathcal{M}(\mathbf{X})$ // $\boldsymbol{\alpha}^{(L,a)} \in \mathbb{R}^{N \times F}$ for each head a
// Initialize temporal attention map
$\mathbf{M}_t \leftarrow \mathbf{0}_F$
// Initialize spatial attention maps for all frames
$\mathbf{M}_s \leftarrow \mathbf{0}_{N \times F}$
// Aggregate temporal attention across all heads and patches
for $t = 1$ **to** F **do**
    $\mathbf{M}_t[t] \leftarrow \frac{1}{AN} \sum_{a=1}^{A} \sum_{p=1}^{N} \alpha_{p,t}^{(L,a)}$
end for
// Aggregate spatial attention for each patch position across all heads
for $t = 1$ **to** F **do**
    for $p = 1$ **to** N **do**
        $\mathbf{M}_s[p, t] \leftarrow \frac{1}{A} \sum_{a=1}^{A} \alpha_{p,t}^{(L,a)}$
    end for
end for
return $\mathbf{M}_t, \mathbf{M}_s$

Attention may capture correlative rather than causal relationships between features and predictions. To address these constraints, our enhanced STAA method employs dynamic thresholding and normalization techniques that mitigate noise and emphasize truly salient

patterns. The justification for our approach is further supported by empirical evidence from our experiments, where STAA's attention-based attributions demonstrate strong faithfulness scores, indicating alignment between identified important features and actual model behavior.

Input and Embedding

Given an input clip video $X \in \mathbb{R}^{F \times H \times W \times 3}$ consisting of F RGB frames, we decompose each frame into non-overlapping patches of size $P \times P$. Each patch $x^{(p,t)} \in \mathbb{R}^{P^2 \times 3}$ is linearly mapped into an embedding:

$$z_{(p,t)}^{(0)} = Ex^{(p,t)} + e_{(p,t)}^{pos} \tag{12}$$

where E is the embedding matrix, and $e_{(p,t)}^{pos} \in \mathbb{R}^D$ represents learnable positional embeddings for spatial position p and temporal position t, as marked 0 to N in Figure 35. Then, z is the overall input to the Transformer, similar to the sequences of words that are used in the language transformer model [191].

Attention Value Extraction

For each transformer layer l and attention head a, we compute query and key vectors for each patch as the TimeSformer architecture [19]:

$$q_{(p,t)}^{(l,a)} = W_Q^{(l,a)} \text{LN}(z_{(p,t)}^{(l-1)}) \in \mathbb{R}^{D_h} \tag{13}$$

$$k_{(p,t)}^{(l,a)} = W_K^{(l,a)} \text{LN}(z_{(p,t)}^{(l-1)}) \in \mathbb{R}^{D_h} \tag{14}$$

where $W_Q^{(l,a)}, W_K^{(l,a)} \in \mathbb{R}^{D_h \times D}$ are learnable weight matrices. $\text{LN}()$ denotes the Layer Normalization. D_h is the dimension per attention head.

The attention weights for Spatio-Temporal attention are extracted as:

$$\alpha_{(p,t)}^{(l,a)} = \text{Softmax}\left(\frac{{q_{(p,t)}^{(l,a)}}^{\top}}{\sqrt{D_h}} \cdot [k_{(0,0)}^{(l,a)} \{k_{(p',t')}^{(l,a)}\}_{p'=1,\ldots,N}^{t'=1,\ldots,F}]\right) \in \mathbb{R}^{NF+1} \tag{15}$$

where l is the index of transformer layer, a is the index of attention head, p, p' are spatial position indices of patches, t, t' are temporal frame indices, $q_{(p,t)}^{(l,a)}, k_{(p,t)}^{(l,a)} \in \mathbb{R}^{D_h}$ are query and key vectors, $D_h = D/A$ is the dimension per attention head, N is the number of patches per frame, F is the number of frames, $(0, 0)$ represent the position index for classification token.

Temporal and Spatial Attention Aggregation

To attribute attention to both temporal and spatial dimensions, we aggregate the attention weights across patches and attention heads. The attributions can be calculated by processing the attention weights α with Algorithm 3.

The temporal attention map $\mathbf{M}_t \in \mathbb{R}^F$ provides a frame-level importance score across the video sequence. This is achieved by averaging attention weights across all attention heads and spatial patches within each frame:

$$\mathbf{M}_t(t) = \frac{1}{AN} \sum_{a=1}^{A} \sum_{p=1}^{N} \alpha_{(p,t)}^{(L,a)} \tag{16}$$

where A is the number of attention heads, N is the number of patches per frame, L is the final layer of the transformer, and $\alpha_{(p,t)}^{(L,a)}$ represents the attention weight at the final layer L, for attention head a, spatial position p, and temporal position (frame) t.

The spatial attention map $\mathbf{M}_s \in \mathbb{R}^{N \times F}$ highlights key regions within each frame by focusing on patch-level attention:

$$\mathbf{M}_s(p, t) = \frac{1}{A} \sum_{a=1}^{A} \alpha_{(p,t)}^{(L,a)} \tag{17}$$

This process allows us to determine which frames and spatial regions within each frame are most influential for the model's classification decision.

STAA Computational Analysis and Limitations

STAA achieves significant computational efficiency through its direct utilization of model attention mechanisms. STAA extracts feature importance directly from the self-attention weights computed in the Transformer model, requiring only $O(N)$ operations where N is the input video clip size. Specifically, STAA accesses the attention values from the layer of the Transformer, which are naturally computed during video processing. This fundamentally differs from post-hoc methods. For example, SHAP requires $O(2^s N)$ operations, where s is the number of temporal segments due to its combinatorial sampling approach. LIME's complexity is $O(KN_pN)$, where K is the number of selected frames and N_p is the number of perturbation samples. STAA achieves efficiency by directly utilizing the model's internal attention mechanism rather than requiring additional sampling or perturbations.

Despite its computational advantages, STAA faces several key limitations. When directly visualising raw attention values as heatmaps overlaid on video frames, we observe significant frame-to-frame fluctuations ("flickering") and diffused attention patterns. The raw attention values often permeate entire frames, making it difficult to identify truly salient regions. This visualisation issue indicates the need for refinement. Different from model-agnostic methods, STAA relies on its self-attention mechanisms. This architectural dependency means STAA cannot be directly applied to other types of video models, such as traditional CNNs. The video Transformer (using 16×16 pixel patches) enables fine-grained spatial analysis and frame-level temporal analysis. This is a significant improvement over post-hoc methods. For a standard 224×224 video frame, this creates 196 patches per frame, allowing detailed spatial attention mapping. Each patch can independently receive attention weights. This is sufficient for action recognition tasks. However, applications requiring

pixel-precise explanations may need further analysis. To address the limitation and further enhance the effectiveness of STAA, we developed a refinement procedure in the following subsection.

STAA Enhancement

We observed that vanilla STAA resulting heatmaps video exhibits significant frame-to-frame fluctuations with background noise. To address these drawbacks of the vanilla method, we developed an enhancement approach focusing on dynamic thresholding and attention focusing.

Dynamic Thresholding for Noise Reduction We introduce dynamic threshold θ_t to frames, calculated as:

$$\theta_t = \mu_t + \lambda\sigma_t \tag{18}$$

where μ_t and σ_t are the mean and standard deviation of attention values for frame t, respectively. The hyperparameter λ controls the threshold's strictness and can be empirically set to between $[0, 1]$. In our experiments, it is set to one to allow more attributes to be indicated. If it is set to zero, that means we filter half of the non-important attention attributes as noise.

Attention Focusing Mechanism We apply this threshold to the STAA results:

$$S'[t,p] = \begin{cases} S[t,p] & \text{if } S[t,p] \geq \theta_t \\ 0 & \text{otherwise} \end{cases} \tag{19}$$

This operation effectively focuses attention on the most salient spatio-temporal regions, reducing noise and improving the clarity of our explanations.

Attention Visualization To enable intuitive human interpretation of the model's focus and facilitate qualitative analysis of attention patterns, we transform the attention values

into visualizable heatmaps. First, we normalize the filtered attention values to the [0,1] range. Then, we blend the colorized heatmap with the original frame:

$$H_t = \frac{S'_t - \min(S'_t)}{\max(S'_t) - \min(S'_t)} \tag{20}$$

where H_t is the normalized attention map. The resulting visualization overlays attention heatmaps on video frames. Here are the result examples shown in Figure 36 with a color spectrum from red to blue.

This visualization highlights the regions and frames that most significantly influence its predictions. Such insights can be used for model evaluation. The complete implementation, including heatmap video explanation samples and source code, is available in our GitHub repository[1].

Cross-Cases Generalizability

STAA demonstrates generalizability across Transformer-based video models. We validated STAA on architectures including TimeSformer [19] and VideoMAE [188]. STAA is applicable to any Transformer-based video model by extracting attention values and mapping them to their proper positions in the video frames. This extraction process requires correctly identifying the Linear attention layers in the specific model architecture. Therefore, manual configuration for attention layer extraction is required for each new architecture.

On the dataset level, our evaluation on Kinetics-400 [95] provides assessment across diverse action categories. Kinetics-400 represents the most comprehensive benchmark for video action recognition, with 400 distinct classes and over 240,000 video clips. It offers greater diversity in human activities, scene settings, and recording conditions than previous datasets such as UCF101 (101 classes) or HMDB51 (51 classes).

[1] `https://github.com/ZeruiW/VideoXAI`

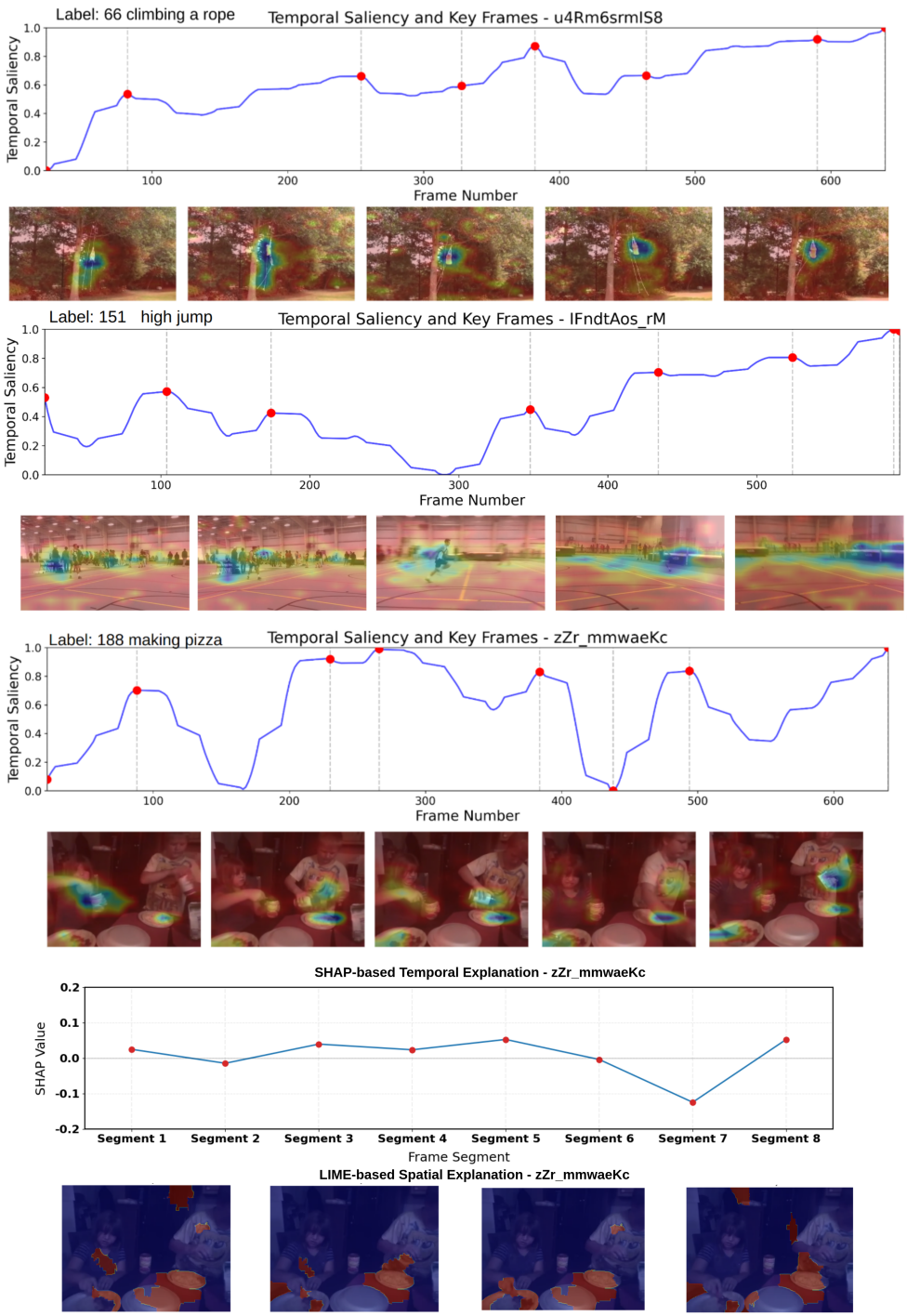


Figure 36: Example of attention visualization: heatmap overlaid on a video frame, where blue colors indicate regions of higher importance, while red indicates lower importance.

In addition, to further validate STAA's cross-dataset capabilities, we conducted experiments on the Something-Something V2 dataset [72], which fundamentally focused on actions without specific objects. This dataset specifically challenges models to understand temporal relationships rather than static scenes. Our analysis confirms that STAA generates coherent explanations on this dataset, with attention maps successfully highlighting interacting objects and tracking their movements through time. The generated results with source code are provided in the project's GitHub Repository. The results are further compared and discussed in Sections IV and V.

These results demonstrate that STAA's effectiveness extends across various Transformer architectures and datasets. The new XAI method addresses the spatio-temporal explanation challenge. STAA provides unified explanations that reflect how these models process video information across both spatial regions within frames and temporal sequences across the video.

5.1.4 Comparison with Recent Video XAI Methods

Comparative Analysis with Model-Agnostic XAI Methods

We conducted comprehensive experiments to evaluate the effectiveness of STAA against established XAI methods using a large-scale video understanding dataset. Our experimental design addresses three key aspects, which are dataset selection and preprocessing, model architecture configuration, and evaluation methodology. We evaluate the Faithfulness and Monotonicity metrics for XAI. Faithfulness indicates how accurately each method's explanations reflect the model's decision-making process. Monotonicity indicates the consistency of feature importance rankings provided by each method.

Experimental Design and Setup

The experiments were conducted using the Kinetics-400 dataset [95], a comprehensive benchmark containing approximately 240,000 video clips across 400 human action classes.

This dataset represents diverse spatio-temporal patterns, from simple actions to complex sequences. Its widespread adoption in the video AI community [11,19,29,122,167] enables meaningful comparisons with existing work while providing sufficient complexity to assess XAI performance across varied scenarios. To address potential class imbalance effects on XAI evaluation, we implemented a balanced sampling strategy across action classes. We randomly selected thirty video files from each of the four hundred action classes, resulting in a total of twelve thousand videos for analysis.

The TimeSformer model [19] was employed as the model architecture, having achieved an accuracy of 78.0% on the Kinetics-400 validation set. This choice was motivated by TimeSformer's state-of-the-art performance and its representative implementation of self-attention mechanisms. The model processes input videos at 224×224 pixel resolution.

The experiments were conducted on a system equipped with an NVIDIA GeForce RTX 4090 GPU, running Ubuntu 22.04.5 LTS. This configuration enables efficient processing of video data. All implementation code and experimental configurations are available in our public repository to facilitate reproducibility and further research in video XAI methods.

XAI Evaluation Metrics

The concept of faithfulness in XAI evaluation is initially developed for image classification tasks [117, 201], measuring how accurately an explanation reflects a model's decision-making process. We extend this concept to video XAI analysis. It quantifies the degree to which the identified important features truly influence the model's output. Monotonicity serves as a more detailed complementary metric. Monotonicity evaluates whether features assigned higher importance scores indeed have a more significant impact on the model's predictions.

Faithfulness quantifies how accurately an explanation represents the model's decision-making process. We define this metric as:

$$\text{faithfulness} = 1 - \frac{1}{N}\sum_{i=1}^{N} |f_{\text{norm}}(x_i) - f_{\text{norm}}(x_i \setminus s_i)| \tag{21}$$

where $f_{\text{norm}}(x_i)$ is the normalized prediction for the original input x_i, $f_{\text{norm}}(x_i \setminus s_i)$ is the normalized prediction after masking the features s_i identified as important, and N is the number of test samples.

We apply min-max normalization to the predictions before calculating the difference. The faithfulness score ranges from 0 to 1, with higher values indicating better XAI performance. A score closer to 1 suggests that masking important features significantly changes the model's normalized prediction, indicating that the explanation has accurately identified crucial features.

Monotonicity assesses whether the importance rankings produced by an explanation method are consistent with their impact on the model's predictions. We quantify this using Kendall's rank correlation coefficient (Kendall's tau) [96], a statistical measure that evaluates the ordinal association between two rankings:

$$\text{monotonicity} = \tau(p_k, d_k) \tag{22}$$

where: p_k represents K numbers in an ascending sequence of masking ratios, for example, [0.1, 0.2, ..., 0.9] in ten ratios, $d_k = f_{\text{softmax}}(x)[c] - f_{\text{softmax}}(x \setminus s_k)[c]$ represents the changes in prediction probability when masking top-K important features s_k represents regions masked according to their importance, τ is the rank correlation coefficient [96].

In our context, a high positive τ value suggests that as we mask more important features (increasing p_k), we observe proportionally larger changes in the model's predictions (d_k), confirming that the XAI method has consistently identified the important features. Conversely, a value close to -1 indicates that the rankings are completely opposite, suggesting that features identified as important by the XAI method actually have a worse impact on

the model's predictions.

This metric can be applied on either temporal feature or spatial feature or even both. For temporal analysis, it evaluates whether masking important time segments leads to consistent prediction changes. For spatial analysis, it assesses the impact of masking important regions within frames. When applied to Spatio-Temporal features, we mask them both. It measures the overall consistency of the selected XAI methods.

Results Analysis and Method Comparison

Table 13 presents the results of our evaluation using faithfulness and monotonicity metrics on the Kinetics-400 dataset.

Table 13: Video Model XAI Evaluation Results

Method	**Faithfulness**	**Monotonicity**	**Computation Time (s)**
SHAP	0.769 ± 0.122	-0.535 ± 0.108	5.59 ± 0.44
LIME	0.525 ± 0.388	-0.411 ± 0.186	46.83 ± 8.68
STAA (Vanilla)	0.190 ± 0.373	0.496 ± 0.437	**0.16 ± 0.01**
STAA (Enhanced)	**0.844 ± 0.116**	**0.850 ± 0.030**	**0.16 ± 0.01**

The results demonstrate varying performance across different XAI methods: SHAP shows moderate faithfulness (0.769 ± 0.122) but poor monotonicity (-0.535 ± 0.108). LIME exhibits low faithfulness (0.525 ± 0.388) and monotonicity (-0.411 ± 0.186), indicating that LIME's explanations may not accurately reflect the model's decision process and lack consistency in feature importance rankings. STAA (Vanilla), using raw attention values, shows poor faithfulness (0.190 ± 0.373) but small positive monotonicity (0.496 ± 0.437). This suggests that the noise in attention values significantly impacts the accuracy of XAI evaluation. STAA (Enhanced), our enhanced method, demonstrates superior performance in both metrics. It achieves the highest faithfulness (0.844 ± 0.116) and monotonicity (0.850 ± 0.030) scores among all methods.

The improvement in STAA's performance from the Vanilla to the Enhanced version

is substantial, indicating the effectiveness of our threshold approach. The high faithfulness score suggests that STAA (Enhanced) provides explanations that accurately reflect the model's decision-making process. The high monotonicity score indicates that the importance rankings produced by our method are highly consistent and reliable across different levels of feature masking.

Both Vanilla and Enhanced STAA methods demonstrate a significant computational advantage over SHAP and LIME, with an average runtime of just 0.16 ± 0.01 seconds. The enhancement algorithm presented in Section 5.1.3 adds negligible overhead to the overall processing time, especially relative to the deep-learning model's complexity, rendering this millisecond-level cost effectively inconsequential. This high efficiency, combined with superior performance metrics, makes STAA (Enhanced) exceptionally well-suited for explaining video analysis tasks. More detailed results, visualization, and reproducible code can be found in our GitHub repository.

Overall, the findings emphasize the strengths of our Spatio-Temporal Attention Attribution approach, particularly in its enhanced form. Combining high faithfulness and monotonicity with minimal computation time, STAA (Enhanced) offers an efficient method for interpreting complex video analysis tasks.

Comparison with Model-specific XAI Methods

This subsection presents a focused comparison between our STAA method and the leading Transformer model-specific XAI method by Chefer et al. [31]. While the previous subsection evaluated STAA against model-agnostic XAI methods, here we directly compare it with a recent XAI method specifically designed for the Transformer vision model. We conduct experiments on both the Kinetics-400 [95] and Something-Something V2 [72] datasets to assess performance across different video understanding tasks.

The recent study by Chefer et al. [31] introduces a gradient-based approach for interpreting Transformer models that extends the principles of Layer-wise Relevance Propagation (LRP) to self-attention architectures. This recent approach represents several different paradigms from our STAA method. Firstly, their approach builds on LRP principles, while our STAA is grounded in the direct interpretation of attention mechanisms. Then, we handle the spatio-temporal token structure in TimeSformer, ensuring appropriate gradient flow through both spatial and temporal attention mechanisms. The previous work only adapts to image models.

XAI Evaluation Metrics

We conducted experiments on two video understanding datasets. Besides the Kinetics-400 [95], which is a large-scale action recognition benchmark dataset. We also employ Something-Something V2 [72], which focuses on actions and object manipulations. This dataset is particularly challenging as it requires models to capture detailed temporal relationships rather than recognizing objects.

We incorporated two additional metrics specifically designed to evaluate explanation quality. **Deletion** [151] measures the decrease in prediction probability as pixels are progressively removed according to their importance. A lower area under the curve (AUC) indicates that removing important regions should cause significant drops in prediction confidence. **Insertion** [151] measures the increase in prediction probability as pixels are progressively added according to their importance. A higher AUC indicates that adding important regions should cause significant increases in prediction confidence.

Results Analysis and Method Comparison

We present a qualitative comparison between explanations generated by Chefer et al.'s method [31] and our STAA method on two representative action videos, as shown in Figure 37. Both methods successfully highlight action-relevant regions, but with notable differences in their focus areas. In the "climbing a rope" example, Chefer's method produces a

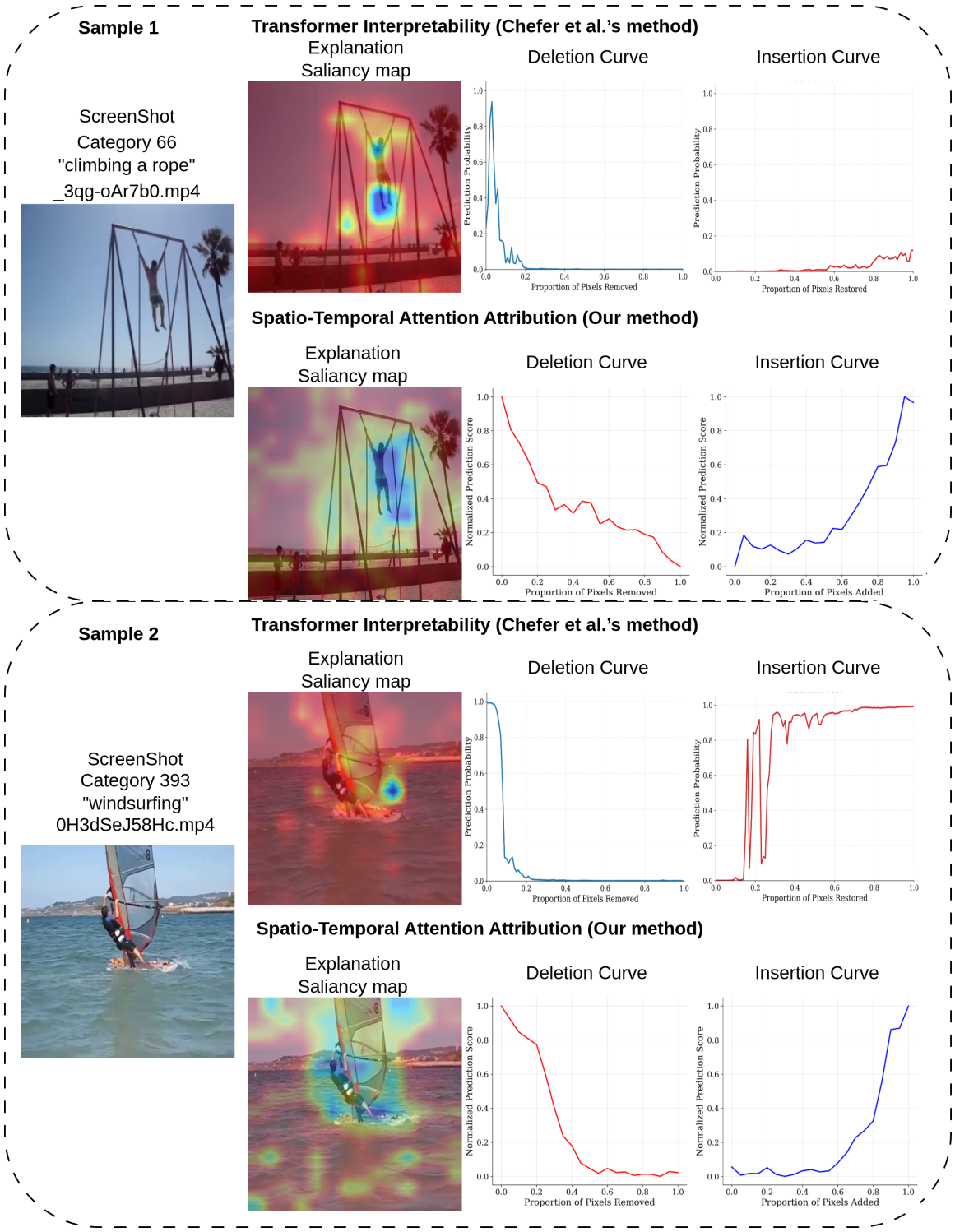


Figure 37: Comparison between the Transformer Interpretability (Chefer et al.'s method) and STAA (Our method). Note: Bluer colors indicate regions of higher importance. For the Deletion/Insertion Curve, the x-axis label is Proportion of Pixels Removed/Added, and the y-axis label is the Prediction Probability Scores

narrower attention map concentrated primarily on the person's legs and head. In contrast, our STAA method captures the climber's entire body along with the relevant rope structure, providing a more comprehensive explanation of action-critical elements. Similarly, for the "windsurfing" sample, Chefer's method highlights a specific focal point near the windsurfer, while STAA provides broader coverage of both the windsurfing equipment and the person, with additional attention to contextually relevant environmental features. These results demonstrate STAA's ability to generate more holistic explanations that incorporate both primary action elements and their supporting contextual features.

The right side of Figure 37 presents the deletion and insertion curves for both methods, providing an assessment of the generated explanations. These curves measure how model confidence changes as regions identified as important are progressively removed (deletion) or exclusively added (insertion). Our STAA method demonstrates consistently smooth curves, indicating that as critical regions are deleted, the model's confidence decreases steadily, and as they are inserted, confidence increases predictably. This behavior confirms that STAA accurately identifies features that truly influence the model's predictions. In contrast, Chefer et al.'s method partially shows inconsistent performance, with the insertion curve for Sample 1 failing to increase as expected, and the insertion curve for Sample 2 exhibiting significant oscillations. These irregularities suggest that adding regions identified as important by their method sometimes introduces noise that interferes with model prediction rather than enhancing it.

Application to Action Recognition

To further evaluate STAA's capability in capturing temporal dynamics across datasets, we applied our method to the Something-Something V2 dataset [72], which presents a fundamentally different challenge compared to Kinetics-400. While Kinetics-400 classifications can often be determined by recognizing objects or scenes in individual frames, Something-Something V2 specifically focuses on action dynamics rather than objects. This

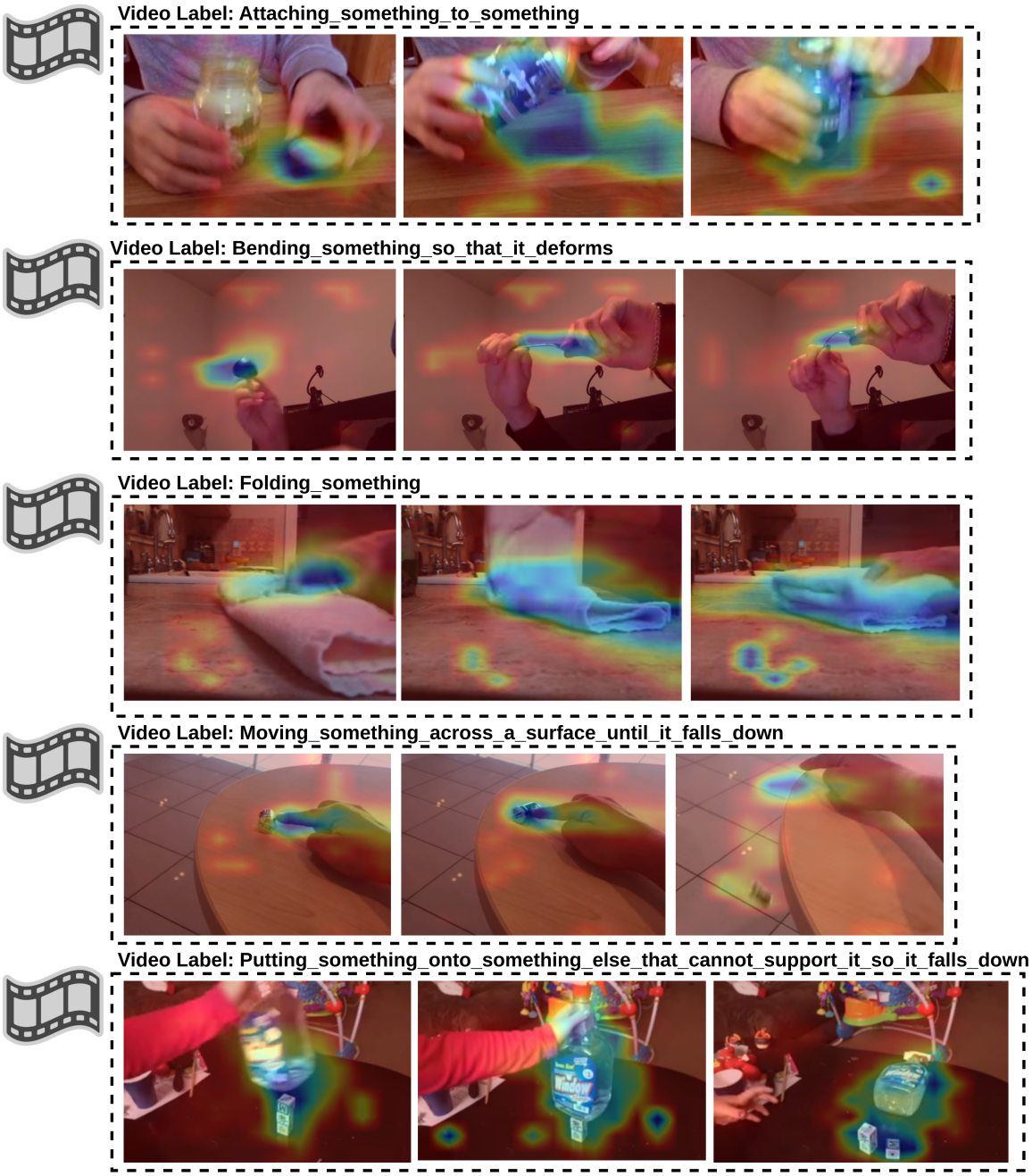


Figure 38: STAA explanations for five representative samples from the Something-Something V2 dataset.

dataset contains 174 fine-grained action categories where temporal relationships and motion patterns are crucial for accurate classification. The objects in the same category can be different.

The distinctive nature of this dataset renders traditional image-based explanation methods ineffective, as they cannot capture the temporal evolution that defines these actions. Current XAI methods have rarely been applied to action-focused datasets due to their limited ability to interpret temporal features. Our STAA method addresses this gap by explicitly capturing temporal features that represent action dynamics. We first trained a TimeSformer model on the Something-Something V2 dataset, achieving competitive performance with state-of-the-art action recognition models. We then applied our STAA explanation method to provide, to the best of our knowledge, the first XAI explanations specifically targeting action-based video recognition.

As shown in Figure 38, STAA explanations for five representative videos from the dataset. For example, in the first video labeled "Attaching something to something," STAA highlights both the bottle cap and bottle opening as the person performs the attachment action, focusing precisely on the cap and bottle during the critical moments of attachment. The temporal attention pattern reveals an increased focus during the actual attachment motion, demonstrating STAA's ability to recognize the decisive moment in the action sequence.

These results demonstrate STAA's unique ability for action recognition tasks, where understanding temporal dynamics is essential.

Our experimental results demonstrate that STAA outperforms traditional model-agnostic methods as well as existing model-specific explanation methods. STAA achieves superior faithfulness (0.844 ± 0.116) and monotonicity (0.850 ± 0.030) scores. Qualitative analysis through deletion/insertion curves shows that STAA provides a more accurate and consistent explanation. Moreover, our experiments on the Something-Something

V2 dataset demonstrate STAA's capability to explain action-focused video models, where temporal dynamics rather than static objects are critical.

5.1.5 Real-Time Video XAI Cloud Architecture

This subsection presents a cloud-based architecture that enables real-time explanation generation for video analysis, addressing the critical challenge of providing hundreds-of-milliseconds latency explanations for edge computing applications [42]. We first establish the architectural framework necessary for real-time video XAI, then evaluate its performance against industry-standard latency requirements.

Architecture Components Overview

To achieve real-time video XAI processing while maintaining system efficiency, we propose a distributed architecture that uses the cloud server's computing resources while minimizing edge device computational overhead. The architecture employs cloud-based GPUs for compute-intensive tasks, including model inference and XAI processing, while edge devices handle data acquisition and visualization, enabling real-time processing within the established hundreds-milliseconds latency threshold [42].

Our proposed architecture integrates five main components to enable efficient real-time video XAI analysis: the Video Sources component serves as the data input, providing various streams such as live camera feeds; the `Video Preprocessing` module performs essential tasks including temporal segmentation, frame extraction, and resizing; the `Transformer Model` implements state-of-the-art video analysis capabilities such as TimeSformer [19]; working in tandem with this model is the `XAI Analysis` component that generates feature attribution explanation through the adopted XAI method; and finally, the `Spatio-Temporal Explanation Visualization` component renders these XAI outputs in human-interpretable formats, for example, heatmap video. These components are distributed between cloud and client devices, as illustrated in Fig. 39. The

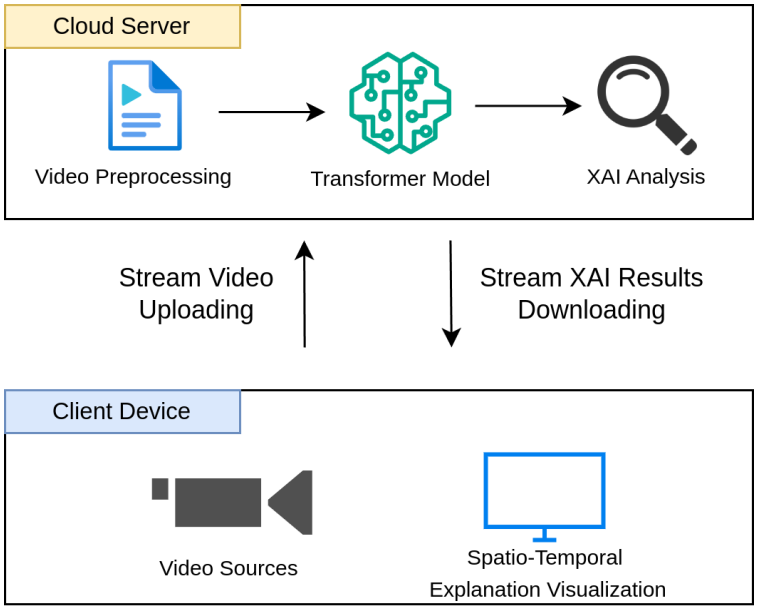


Figure 39: Cloud-based architecture for real-time video XAI

cloud service handles the compute-intensive components, while client-side devices focus on data collection and real-time visualization of the explanations. This design specifically addresses the challenges faced by resource-constrained edge devices for complex XAI tasks [63, 202].

Performance Evaluation for Real-Time Scenarios

To evaluate the applicability of our STAA method, we implemented a test for real-time video XAI. This system demonstrates the practical feasibility of generating explanations for video streams.

Server development Our implementation consists of two main components.

Client. The client reads video files using the PyAV library, which allows for efficient frame extraction. It segments the video into batches that suit model inputs and sends these batches to the server using ZeroMQ, a high-performance asynchronous messaging library. The client side enables real-time streaming of video data to the server side.

Server. The server, built around the AI model and XAI methods, receives frame batches

and processes them in real time. The Video AI model performs inference on the frame batches. The XAI method generates heatmaps and produces explanations for each batch of frames. The server utilizes multi-threading to separate frame reception from processing, ensuring continuous data flow.

Real-time Processing Strategies

Our system achieves real-time processing through several key strategies:

Fine Batch Processing. The server processes video in batches of frames, aligning with the TimeSformer model's input requirements. This approach balances the trade-off between processing granularity and computational efficiency. Smaller batches could provide finer-grained XAI analysis, while larger batches mean processing longer videos in each model inference. In the test, we use an eight-frame batch size that represents the minimal unit that the model can process.

STAA Method. The system employs our STAA method to extract and process attention values from the model's internal mechanisms. It generates spatio-temporal feature importance through the following steps: (1) decomposing frames into non-overlapping patches, (2) computing attention weights $\alpha_{(p,t)}^{(l,a)}$ from the final transformer layer, and (3) aggregating these weights across attention heads to produce temporal importance scores M_t and spatial importance maps M_s. The method employs dynamic thresholding ($\theta_t = \mu_t + \lambda\sigma_t$) to reduce noise and enhance visualization clarity.

Asynchronous Frame Processing. The system implements an asynchronous pipeline where frame extraction, model inference, and explanation generation operate independently. This design prevents bottlenecks and ensures continuous processing flow, which is essential for real-time performance.

Computational Efficiency Evaluation and Analysis

To provide a transparent justification for computational efficiency, we conducted a detailed breakdown of the processing time required by each method at different stages of the

explanation pipeline. Table 14 presents this comprehensive comparison.

Table 14: Computation Time Breakdown by Processing Stage (in seconds)

Module	**SHAP**	**LIME**	**STAA**
Video Preprocessing	0.009	0.009	0.009
Model Inference	5.187	22.173	0.072
Explanation Calculation	0.385	24.639	0.070
Visualization & Storage	0.009	0.009	0.009
Total Time	5.59 ± 0.44	46.83 ± 8.68	**0.16 ± 0.01**

This breakdown reveals the specific advantages of STAA over traditional post-hoc methods. The preprocessing stage requires identical minimal time (0.009s) across all methods to decode video frames. The substantial differences emerge in the model inference stage, where STAA requires only a single inference pass (0.072s), while SHAP necessitates multiple inferences (5.187s) to evaluate different feature combinations, and LIME demands even more rounds to reach an approximate estimation (22.173s) for processing numerous perturbed input versions.

In the explanation calculation stage, STAA achieves significant efficiency (0.070s) by directly utilising attention values already computed during inference. In contrast, SHAP (0.385s) must compute Shapley values across temporal segments, and LIME (24.639s) requires fitting local interpretable models to various input approximations. The final visualisation stage takes minimal time (0.009s) across all methods.

Based on the optimised XAI algorithm, we conducted performance testing of our real-time video XAI system, focusing on the Latency metric. Measured as the end-to-end processing time between receiving a batch of frames and completing feature attribution explanation. The test results are illustrated in a Cumulative Distribution Function diagram in Figure 40.

These results demonstrate that our system can process frames between one hundred milliseconds and one hundred and fifty milliseconds, allowing for near-immediate explanation

generation. Besides, general action classification tasks do not require a frame-by-frame explanation. Therefore, in practical applications, practitioners can adjust the sampling rate according to their specific needs. For instance, since most human actions evolve over multiple frames, sampling eight to thirty-two frames per video clip (around ten seconds) is typically sufficient for accurate action recognition and explanation generation [11, 19].

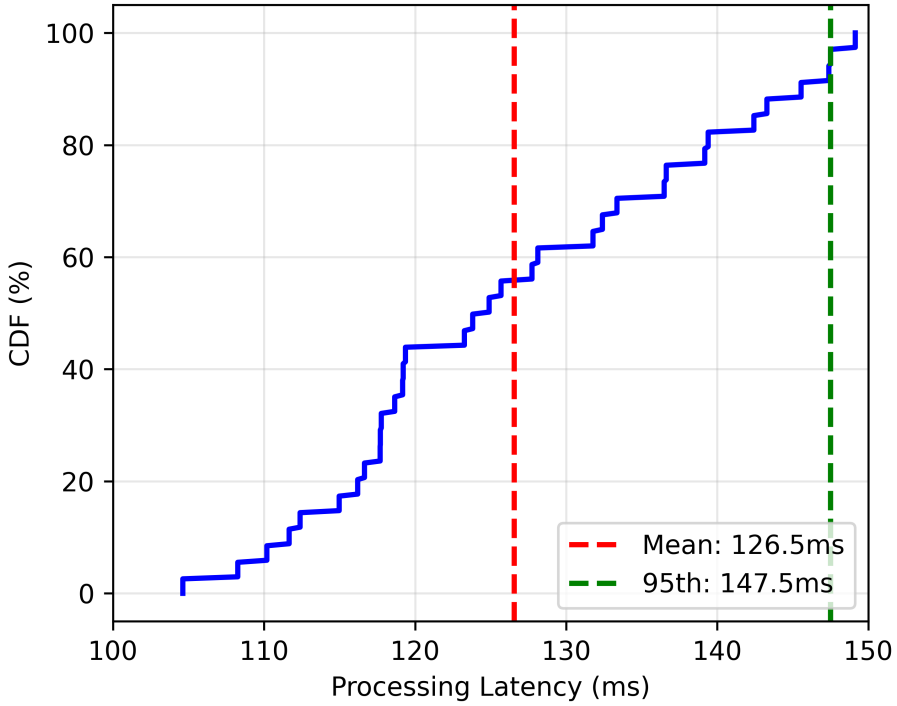


Figure 40: Cumulative Distribution Function (CDF) of the STAA processing latency

Traditional post-hoc XAI methods, such as SHAP or LIME, require multiple passes over the entire video. The latency for these methods (5.59 seconds for SHAP and 46.83 seconds for LIME) far exceeds the requirements for real-time applications. Our STAA-based system generates explanations in a single pass of model inference. This fundamental difference in approaches enables our system to provide explanations with minimal delay, making it suitable for real-time applications.

To facilitate reproducibility and future development, we provide the source code for

real-time video XAI server implementation on GitHub [2]. This codebase serves as a practical demonstration of our method and can be used as a foundation for developing more advanced XAI services in video analysis. The repository provides a guide for setting up the server and client.

In conclusion, our implementation demonstrates the practical feasibility of real-time video XAI, opening up new possibilities for applications requiring immediate insights into AI decision-making processes in video analysis tasks.

> We designed and implemented a cloud-based architecture for real-time video XAI, employing GPUs on the cloud server for efficient model inference and XAI processing. This approach enables client-side devices to achieve an average latency of hundred-millisecond per frame explanation, demonstrating the feasibility of real-time video XAI for edge devices.

5.1.6 Summary

This work presents an advancement in the field of explainable AI for video analysis, addressing the need for XAI in increasing video AI deployment. Our proposed STAA method represents a new approach to generating explanations for video Transformer models, overcoming the limitations of existing post-hoc explanation techniques. The key contributions of this work are threefold: (1) We introduce STAA, a method that provides spatio-temporal explanations simultaneously. The computational cost is reduced by 97% compared to SHAP and LIME. (2) Through extensive experimental evaluation on the Kinetics-400 dataset, we demonstrate that STAA significantly outperforms established methods, leading to higher faithfulness and monotonicity scores. (3) We present a cloud-based architecture for real-time video XAI, showcasing STAA's practical applicability in scenarios requiring immediate explanations. Our implementation measured an average latency of

[2] https://github.com/ZeruiW/RealTimeServer

less than one-hundred-fifty milliseconds per frame, demonstrating the feasibility of real-time explainable video analysis.

In the future, we will further examine the resilience of the STAA method under various adversarial attacks and test its adaptability across different novel Transformer video models and hybrid architectures.

5.2 Adversarial Attacks on Video Transformer Models Through XAI-guided Perturbation

Transformer models have revolutionized computer vision, achieving state-of-the-art performance across image classification [47], segmentation, and video action recognition [19]. In the video domain, transformer architectures offer unique advantages by processing video data as tokenized patches and capturing joint spatiotemporal dependencies through self-attention mechanisms. Video transformer architectures integrate spatial attention within individual frames and temporal attention across frame sequences simultaneously, enabling a more comprehensive understanding of video content than the sequential processing approaches of conventional architectures.

Despite their impressive capabilities, vision models remain vulnerable to adversarial attacks [3], which are carefully designed perturbations that cause models to make incorrect predictions. This vulnerability raises significant concerns about model robustness and security, particularly for applications in safety-critical domains. While adversarial attacks have been studied for CNN-based image models, their impact on video transformer models remains inadequately explored [3, 87, 91]. Ensuring the adversarial robustness of these models is essential for their deployment in safety-critical applications. The fundamental architectural differences between CNN-based models and transformer-based video models create a significant research gap [198]. Video transformers process information through

self-attention mechanisms that jointly model spatial and temporal dependencies, which stand in stark contrast to the local receptive fields of CNNs [19]. This architectural divergence suggests that adversarial vulnerabilities in video transformers may differ from previous studies. While recent work has proposed transferable adversarial attacks for video models [198, 199], these approaches do not fully exploit the joint spatiotemporal nature of video transformer architectures.

Simultaneously, explainable AI (XAI) methods have emerged as crucial tools for understanding model behavior and identifying potential vulnerabilities. For vision models, these techniques typically generate saliency maps highlighting regions that strongly influence predictions. Despite their widespread application to image models, XAI methods remain underutilized in the context of video transformers and adversarial robustness analysis. Our recent studies in vision model robustness analysis [198] and spatiotemporal feature XAI analysis [199] provide foundations for developing new analytical explanations specifically designed for video transformer models. The intersection of video transformer architectures, adversarial robustness, and explainability presents research challenges that have not been adequately addressed in current literature. While recent studies have investigated adversarial attacks on conventional vision models [3,67,81,91,128], their findings cannot be directly applied to transformer-based video architectures due to fundamental differences in feature processing mechanisms. Transferable adversarial attacks [198] represent initial steps toward addressing this gap, but they fail to leverage the integrated spatiotemporal representation capabilities that distinguish video transformers from other architectures. Additionally, the XAI techniques to explain adversarial vulnerability mechanisms in video transformer models remain unexplored, with existing XAI approaches [201] primarily focused on static image tasks rather than the complex spatiotemporal video understanding.

To address these research gaps, we investigate the following three research questions:

How can an adversarial attack method effectively target spatiotemporal dependencies in transformer-based video models? The existing attack methods perturb frames independently, ignoring the temporal relationships that video models learn through self-attention mechanisms. The existing approach fails to exploit the vulnerability of video transformers.

Can XAI methods reveal how adversarial attacks disrupt spatial and temporal attention patterns in video transformers? Understanding the impact of perturbations on model attention can provide insights into vulnerability mechanisms. While XAI methods have been widely used to interpret image model, their application to adversarial attacks on video transformers remains unexplored. We investigate XAI techniques that can reveal the impact of adversarial perturbations on both spatial attention within frames and temporal attention across frames. We quantify the impact of adversarial perturbations on a model's attention.

How effective is our joint spatiotemporal attack compared to existing adversarial methods against video transformer models? While adversarial attacks have been studied, their effectiveness when applied to transformer-based video architectures remains inadequately studied through systematic comparison. This investigation conducts an empirical evaluation comparing recent attack methods, including common corruption benchmarks, frame-wise attacks, sparse attacks, and V-BAD attacks, against our method. Through quantitative performance metrics and controlled experiments across multiple datasets, we demonstrate that our method outperforms previous approaches, providing evidence for the vulnerability of video transformer architectures.

The key contributions of this work include: (1) We develop a novel joint spatiotemporal adversarial attack method specifically designed to target video transformer attention mechanisms, achieving more accuracy degradations and ASR compared to the previous method. (2) We introduce new quantitative metrics based on XAI analysis to evaluate the impact

of adversarial attacks on both spatial and temporal attention distributions. We also provide a novel 3D visualization methodology that unifies spatial and temporal analyses. (3) We provide an experimental study for the effectiveness of common corruptions, frame-wise attacks, sparse attacks, V-BAD attacks, and our proposed joint attack method across multiple video transformer architectures.

To support reproducibility and facilitate future research, we provide the implementation of our methods and analysis tools through our GitHub repository[3].

5.2.1 Recent Studies: Adversarial Attacks on Video Models

This section synthesizes three research domains that inform our understanding of adversarial robustness in video transformers. First, we examine the architectural evolution from CNN-based to transformer-based video models, emphasizing their distinct spatiotemporal processing mechanisms. Second, we analyze recent advances in adversarial attacks on vision models, tracing their progression from image to video domains while identifying gaps in transformer-specific vulnerabilities. Finally, we review explainable AI (XAI) methodologies that facilitate the analysis of model behavior under adversarial conditions.

Transformer-based Vision Models

Convolutional Neural Networks (CNNs) have long served as the foundation of computer vision research, achieving remarkable success in static image classification tasks. However, their inherent limitations in modeling temporal dependencies pose significant challenges for video understanding tasks. To address this, early approaches such as C3D (Convolutional 3D Network) [189] extended 2D convolutions to 3D, enabling simultaneous spatial

[3] https://github.com/ZeruiW/VideoTransformer-Adversarial-XAI

and temporal feature extraction. While C3D demonstrated effectiveness in action recognition tasks, it suffered from high computational costs due to the cubic expansion of convolutional operations. This led to the introduction of I3D (Inflated 3D ConvNet) [29], which adapted ImageNet-pretrained CNNs into 3D by inflating kernels.

Subsequently, the success of self-attention mechanisms in natural language processing inspired a fundamental paradigm shift in vision architectures. The Vision Transformer (ViT) [47] demonstrated that images could be processed as tokenized patch sequences, eliminating the need for convolutions. ViT outperformed CNNs on large datasets, proving that self-attention could serve as an alternative to convolutional feature extraction. Following ViT's success, researchers extended transformers to video modeling. TimeSformer [19] proposed a space-time attention mechanism, factorizing attention computation into separate spatial and temporal branches. TimeSformer reduced computational complexity while achieving state-of-the-art results on video classification benchmarks.

Table 15 summarizes the key characteristics of different video model architectures. The evolution of video models can be categorized into three main types: 3D CNN-based, CNN-RNN hybrid, and Transformer-based approaches. Each architecture type presents distinct characteristics in temporal modeling and computational efficiency.

Despite the success of transformer-based video models in action recognition, their robustness against adversarial attacks remains unexplored [3]. CNNs extract local features, while transformers rely on global self-attention across spatial and temporal tokens. This fundamental difference raises our concern about the ability of accumulated adversarial attack techniques. Our study aims to examine the robustness of transformer-based video models against the previous adversarial attacks.

Table 15: Comparison of Video Models: CNN-based, RNN-based, and Transformer-based Architectures

Type	Model	Temporal Modeling	Computational Complexity
3D CNN	C3D [189]	3D convolution captures spatiotemporal dependencies locally.	High; complexity scales with the temporal dimension and kernel size.
	I3D [29]	Inflated 3D convolution expands 2D CNNs to model temporal dynamics.	High; retains the high cost of 3D convolution.
	S3D [213]	Separable 3D convolutions reduce redundancy in feature extraction.	Moderate; factorized convolutions improve efficiency.
CNN-RNN	Conv-LSTM [171]	Recurrent structure captures temporal relationships over frames.	Moderate; sequential processing limits parallelism.
	LRCN [45]	CNN extracts spatial features; LSTM encodes temporal dependencies.	Moderate; frame-wise feature extraction adds overhead.
	Pred- RNN [196]	Spatiotemporal memory cells enhance sequential learning.	High; increased recurrent computations.
Transformer	TimeSformer [19]	Factorized space-time attention improves efficiency.	Low; linear in spatial and temporal dimensions.
	Video MAE [188]	Masked autoencoder for video modeling.	high; masked token modeling remains expensive.

Adversarial Attacks on Video Action Recognition Models

Computer vision models are susceptible to adversarial attacks, which introduce crafted perturbations to input data, leading to incorrect predictions. The Fast Gradient Sign Method (FGSM) [67], introduced by Goodfellow et al., established a foundational white-box attack paradigm that balances computational efficiency with attack effectiveness in the image domain. FGSM generates adversarial perturbations using the gradient of the loss function:

$$x^{adv} = x + \epsilon \cdot \text{sign}(\nabla_x J(x, y; \theta)) \tag{23}$$

where x is the input, ϵ controls the perturbation magnitude, $J(x, y; \theta)$ is the loss function, and θ represents the model parameters.

Similarly, Projected Gradient Descent (PGD) [128] and other works enhances FGSM [67] through iterative optimization:

$$x_{t+1}^{adv} = \text{Proj}_{\mathcal{B}_\epsilon(x)} \left(x_t^{adv} + \alpha \cdot \text{sign}(\nabla_x J(x_t^{adv}, y; \theta)) \right) \tag{24}$$

where α is the step size and $\text{Proj}_{\mathcal{B}_\epsilon(x)}$ projects the perturbed input onto the ϵ-bounded neighborhood of x.

While the above methods demonstrate computational efficiency and effectiveness in static image classification, their application to video data reveals fundamental architectural mismatches. The core limitation stems from FGSM's single frame gradient computation, which is inherently myopic and fails to account for the complex spatiotemporal dependencies that define video understanding models [34, 87]. Their effectiveness fails in video models [87]. Specifically, the naive extension of FGSM to video data treats a video as an independent "bag of frames". The attack on a frame-by-frame basis disregards the temporal dimension [34], which is the defining characteristic of video data.

Counter to image models, video transformers rely on intricate temporal relationships

learned through self-attention mechanisms across frame sequences. FGSM's frame-wise application treats video as a mere collection of independent images, completely ignoring the temporal coherence that video models leverage for robust classification [175]. Furthermore, the frame-independent perturbation introduces severe perceptual artifacts in the temporal domain [56]. When the attack is applied frame-by-frame sequentially, the uncorrelated noise patterns across adjacent frames manifest as highly visible flickering effects that violate the imperceptibility principle of adversarial attacks [90]. Consequently, the very constraint intended to ensure imperceptibility in static images paradoxically induces highly conspicuous and unnatural artifacts in video [56]. This inconsistency not only makes the attack easily detectable by human observers but also provides a strong signature for automated defense mechanisms. For example, if a defence algorithm just verifies temporal consistency across frames, it can uncover the abnormal temporal inconsistency [90, 175]. FGSM also renders it vulnerable to basic preprocessing in video processing pipelines. The video data operations can effectively neutralize FGSM-generated perturbations due to the frame-based perturbations [175]. These limitations demonstrate why FGSM, despite its success in image domains, fails to provide an effective foundation for video adversarial attacks, necessitating the development of spatial-temporal attack methods for video data.

Building upon these foundational insights, we review recent advancements in video adversarial attacks. Table 16 provides a taxonomy of existing methods, categorizing them by attack type, perturbation scope, and target architectures. The Common Corruption Benchmark [81] has significantly influenced research in robustness evaluation and dataset benchmarking. This work investigates vision model resilience under natural perturbations, demonstrating substantial accuracy degradation when models encounter real-world distortions. As a standardized vision model benchmark, CCB is employed to evaluate model robustness across different architectures and domains, establishing a baseline for comparing adversarial attacks. Frame-wise transfer approaches apply perturbations independently

Table 16: Recent Studies on Adversarial Attacks on Video Models

Method	Type	Perturbation Scope	Knowledge	Model(s) Evaluated	Performance / Observations
CCB (2019) [81]	Black-box	Frame-wise	Transfer-based	ResNet, VGG	Benchmark perturbations from common real-world corruptions.
V-BAD (2019) [91]	Black-box	Frame-wise	Query-based	I3D, LSTM	Transfers perturbations from image models and optimizes through gradient estimation.
Heuristic (2020) [205]	Black-box	Frame-wise	Query-based	C3D, LRCN	Heuristic frame selection algorithm reduces query budget by 28%.
RL-Adv (2021) [203]	Black-box	Sparse	Query-based	C3D, LRCN	Sparse adversarial attack use reinforcement learning to select key frames.
I2V (2022) [206]	White-box	Frame-wise	Transfer-based	NL, SlowFast, TPN	Use adversarial perturbation generated on white-box image models to attack video models.
BSC (2022) [32]	Black-box	Bullet Screen	Query-based	C3D, LRCN	Bullet Screen Attack generated with a reinforcement learning framework.
FMM (2024) [116]	White-box	Video-Text Alignment	Transfer-based	Video-based LLMs	Attacks video-language models by modifying alignment.
SVASTIN (2024) [148]	White-box	Sparse	Model Access	MVIT, SlowFast	Sparse adversarial attack through the guide target video learning module.
This Study	White-box	spatio- temporal	Model Access	TimeSformer	Perturbation depends on Transformer model's self-attention.

to each frame, resulting in high computational costs and query budgets while potentially compromising attack stealthiness. To address these limitations, sparse key-frame transfer methods [148] selectively target important frames identified through optical flow analysis or pre-trained model activations, significantly reducing computational requirements while maintaining attack effectiveness on CNN-based architectures.

Our study addresses the research gap by investigating adversarial attacks specifically designed for transformer-based video models. As highlighted in the bottom line in Table 16, we propose a white-box approach that explicitly targets joint spatiotemporal dependencies, providing an attack method that exploits the spatiotemporal architectural characteristics of video transformers.

Explainable AI Methods for Vision Models

Explainable AI methods aim to enhance the transparency and interpretability of model decisions [85]. Feature attribution analysis, a core component of XAI, quantifies the contribution of individual features to model predictions. The XAI taxonomy [85] distinguishes between two primary categories: model-agnostic and model-specific methods.

Model-agnostic methods are derived from controlling the input and then analyzing the output. For instance, the model-agnostic XAI methods in computer vision, such as Mean Centroid Prediff [115], derive explanations from partial variation of the input. Therefore, a significant amount of interaction is required to obtain the explanation. Model-specific methods derive information from the model's internal architecture to generate explanations. These approaches are commonly adopted in computer vision [198] due to their computational efficiency and architectural compatibility with vision models. The Grad-CAM [**?**] series methods contain numerous refined variants through additional architectural insights.

The architectural transition from CNNs to Transformers necessitated fundamental adaptations to existing XAI methods. CNNs operate through hierarchical local receptive fields,

while Transformers utilize global self-attention mechanisms across tokenized inputs. Our recent work on the Spatiotemporal Attention Attribution (STAA) method [199] addresses this limitation by providing comprehensive explanations for transformer-based video models. STAA simultaneously captures both spatial feature attribution within frames and temporal dependencies across frame sequences, offering a more complete understanding of model behavior. Comparative evaluations demonstrate that STAA significantly outperforms baseline methods in both faithfulness to model behavior and computational efficiency when applied to video transformer architectures.

5.2.2 Spatiotemporal Adversarial Attack on Video Transformers

This section presents an approach for generating adversarial attacks specifically designed to exploit the architectural vulnerabilities of transformer-based video models. Our method integrates spatiotemporal processing capabilities to generate adversarial perturbations.

Spatiotemporal Attack Design

As illustrated in Fig. 41, our attack methodology follows a workflow that preserves spatiotemporal coherence throughout the process. The pipeline begins with video preprocessing (left), where input videos are segmented into frame chunks that align with TimeSformer's temporal processing window. These segments undergo transformation into 5D tensor representations that maintain both spatial dimensions and temporal relationships.

The central component of our approach involves feeding these tensor representations into the video model, such as TimeSformer [19], enabling gradient computation with respect to the cross-entropy loss. By performing backpropagation through the model architecture, we obtain gradient values that indicate how each pixel in each frame influences the model's classification decision. These gradients directly reveal the most vulnerable

aspects of the model's spatiotemporal processing, allowing us to generate minimally perturbed adversarial examples that maximize classification error. The gradient-based attack algorithms, such as FGSM [67], then use the sign of these gradients to create perturbations that are applied uniformly across the video segment. The final stage (bottom of Fig. 41) involves reconstructing the segments into a full-length perturbed video. This step ensures that the adversarial effect persists throughout playback.

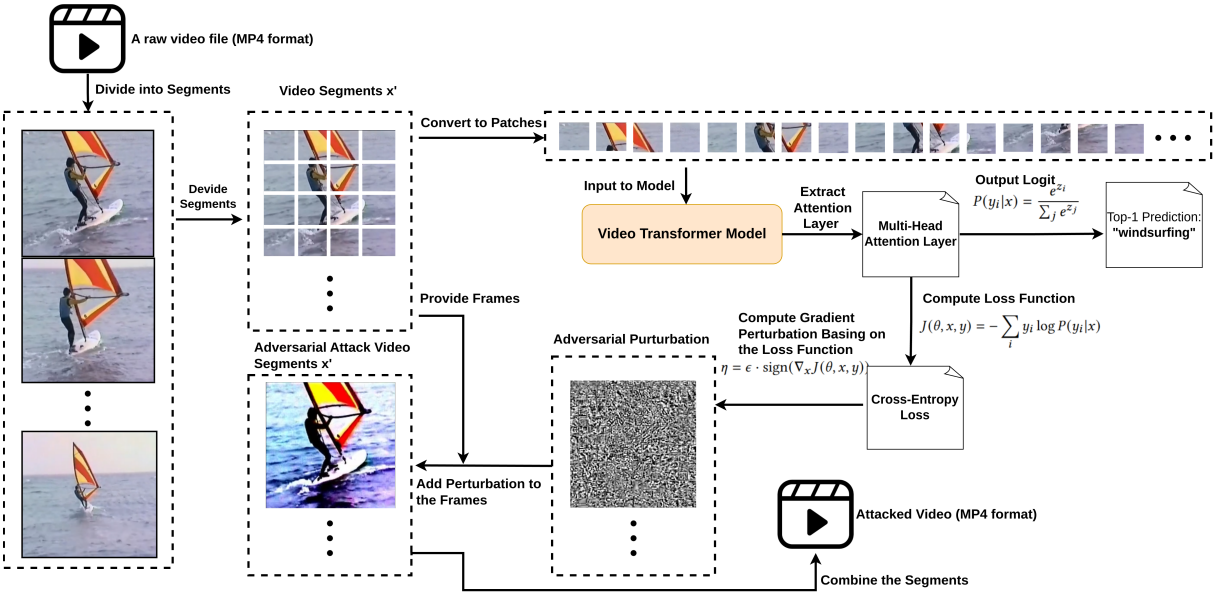


Figure 41: Overview of our spatiotemporal attack pipeline. The process maintains temporal consistency through frame processing and reconstruction while leveraging TimeSformer's attention mechanisms for effective perturbation generation.

The core of our approach lies in its processing of video segments instead of previous frame-wise attacks. We process videos in multi-frame chunks, precisely matching the Transformer's native temporal attention window, such as eight frames for TimeSformer [19]. This architectural alignment enables our attack to manipulate the model's joint spatiotemporal processing mechanisms directly rather than treating frames as independent entities. Each temporal segment undergoes a precise transformation to create a structured 5D tensor representation $\mathbf{X'}_k \in \mathbb{R}^{1\times8\times3\times224\times224}$ in the multi-head self-attention layer. We

extract the values from the multi-head self-attention layer. Then, it can be used to compute adversarial perturbation data through the gradient perturbation.

Throughout our pipeline, we maintain frame rate and encoding quality using specialized video processing techniques. The PyAV library enables precise frame manipulation while preserving the original video properties. Our reconstruction process ensures that the adversarial perturbations remain effective across the entire video sequence without introducing temporal artifacts that could compromise attack fidelity or create unintended detection signals.

Implementation Details

We implemented our spatiotemporal attack using PyTorch with the pre-trained TimeSformerForVideoClassification model [19] from the Hugging Face Transformers library. The implementation processes videos from the Kinetics-400 dataset [95], maintaining their original properties. Our processing pipeline consists of three main stages: preprocessing, perturbation generation, and reconstructed video evaluation. For each video segment $\{x_1, x_2, \ldots, x_8\}$ with corresponding ground-truth label y, we construct a 5D tensor representation $\mathbf{x} \in \mathbb{R}^{1\times 8\times 3\times H\times W}$, where $H = W = 224$ matches the transformer's input requirements. This specific 8-frame segment size aligns with TimeSformer's temporal processing window, enabling direct manipulation of the model's attention mechanisms. For perturbation generation, we compute gradients by tracking the computational graph with `requires_grad=True` in PyTorch. The attack execution computes the cross-entropy loss $J(\mathbf{x}, y; \theta)$ between predicted and true labels, obtaining the input gradient ∇_x through backpropagation. This gradient computation specifically targets the joint spatiotemporal feature space rather than individual frames, allowing our method to directly manipulate the transformer's attention mechanisms.

The cross-entropy loss $J(\cdot)$ serves as our optimization objective, guiding the perturbation generation toward maximally disruptive patterns that specifically target the model's classification decision boundaries. The perturbation magnitude ϵ is empirically optimized to 0.01 based on preliminary experiments that balance attack effectiveness with perturbation imperceptibility. To ensure methodological consistency and facilitate fair comparative evaluation, we maintain this same ϵ value across all experimental conditions, including baseline comparisons. This gradient guides the perturbation generation, followed by post-processing steps, including pixel intensity clamping to maintain visual plausibility and format conversion for video reconstruction. The perturbed frames maintain temporal consistency through precise MP4 encoding that preserves the original video parameters, ensuring that artifacts do not interfere with the experimental validity.

Initial Test Results

In model evaluation, we implement temporal consensus through majority voting across segment predictions, aggregating classification results from all temporal segments to produce the final prediction. This approach aligns with the model's natural processing of video information and provides a comprehensive assessment of attack effectiveness across the entire temporal dimension. When applied to TimeSformer on the Kinetics-400 dataset, our method reduces the model's accuracy from 0.544 to 0.139, representing a 74.4% degradation in performance while maintaining temporal coherence. This significantly outperforms previous approaches - frame-wise attacks only achieved a 13.1% reduction (from 0.544 to 0.473), and V-BAD achieved a 64.7% reduction (from 0.544 to 0.192). The detailed results, as shown in the subsequent section Table 19, demonstrate that our synchronized temporal processing approach effectively exploits transformer attention mechanisms compared to existing methods.

In addition, our method achieves this superior performance while maintaining computational efficiency through direct feature manipulation. The computational complexity scales linearly with model size since we compute gradients directly on the feature space. This represents a significant advantage over the recent SOTA query-based methods, such as V-BAD, which require multiple queries to iteratively refine perturbations. The query-based approaches necessitate repeated model evaluations, leading to computational complexity that grows with each refinement iteration. By eliminating the need for iterative refinement, our method not only achieves better attack performance but also reduces computational overhead, making it more practical for real-world applications and large-scale vulnerability assessments.

Our investigation developed a novel adversarial approach that synchronizes perturbation generation across precise multi-frame temporal windows. These windows are specifically designed to match the video transformer's native temporal attention mechanisms. This approach allows for linear scaling with model size, effectively eliminating the iterative refinement overhead that is typically required by query-based approaches.

5.2.3 Spatiotemporal XAI Analysis on Video Transformers

This section presents our XAI-based analysis framework for understanding how adversarial perturbations affect spatiotemporal attention patterns in TimeSformer models. We employ the STAA method [199] to generate attention maps and analyze them through three complementary perspectives: spatial attention distribution within frames, temporal attention evolution across frames, and unified spatiotemporal attention dynamics.

Our analysis proceeds in three stages. First, we quantify spatial attention disruptions using structural similarity metrics to reveal how adversarial attacks alter focus regions within individual frames. Second, we track temporal attention evolution to understand how the model's focus coherence degrades across frame sequences. Finally, we introduce a novel

3D visualization methodology that unifies spatial and temporal analyses, providing an intuitive representation of attention disruption under adversarial conditions.

Spatial Analysis

XAI saliency maps visualize regions of the video frames that significantly influence the model's predictions. We generate these maps using the STAA method [199], which extracts attention patterns from the self-attention layers of the video transformer. These saliency maps highlight spatial regions where the model focuses when making predictions.

As illustrated in Fig. 42, the XAI-generated saliency maps show the model's focus areas, with brighter regions indicating higher attention values. The combined visualization (right side) overlays these attention values on the video frames, providing an interpretable representation of how the model processes spatial information within the video.



Figure 42: Screenshot of the XAI-generated saliency map example.

We employ the Structural Similarity Index Measure (SSIM) [197] to compare the XAI saliency maps (middle in Fig. 42) between clean and adversarial inputs. For each frame i, we compute SSIM between the clean attention map $S_i^c(u, v)$ and its adversarial counterpart $S_i^{\text{adv}}(u, v)$:

$$\text{SSIM}_i = \frac{(2\,\mu_c\,\mu_a + C_1)\,(2\,\sigma_{ca} + C_2)}{(\mu_c^2 + \mu_a^2 + C_1)\,(\sigma_c^2 + \sigma_a^2 + C_2)}, \tag{25}$$

where μ_c, μ_a represent the mean attention values, σ_c^2, σ_a^2 their variances, and σ_{ca} their covariance. The constants C_1, C_2 ensure numerical stability. This metric captures both

structural and intensity-based differences in attention distribution, with values ranging from 0 (completely different) to 1 (identical).

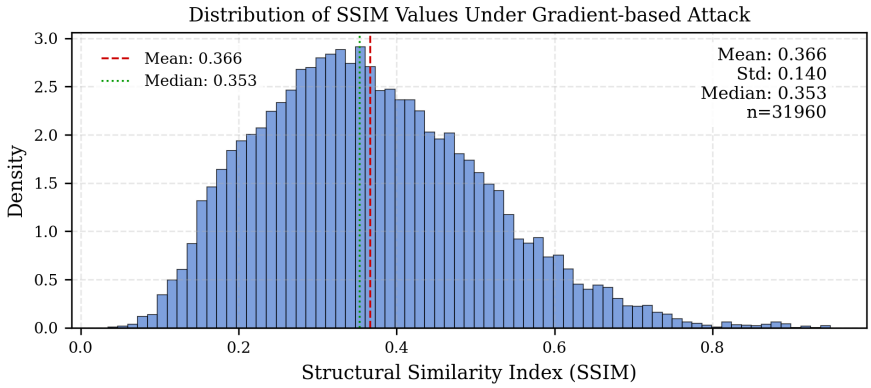


Figure 43: Distribution of SSIM values comparing clean and adversarial attention maps under adversarial attack. The histogram shows the density of SSIM scores across all 20,000 analyzed videos. Lower SSIM values represent greater attention map distortion, demonstrating the impact of adversarial perturbations on spatial attention patterns.

Our analysis of over 20,000 video clips reveals significant attention disruption under adversarial attack. As shown in Fig. 43, the distribution of SSIM values exhibits a clear shift in attention maps across all frames. The median SSIM value of 0.353 is substantially below 1, indicating severe distortion in spatial attention patterns between clean and adversarial inputs. This quantitative measure confirms that adversarial perturbations fundamentally alter how the model attends to visual information within frames.

Temporal Analysis

To quantify how adversarial attacks affect the temporal dynamics of video understanding, we analyze the evolution of attention patterns across frame sequences. Our analysis tracks the temporal attention values using the final attention layer of TimeSformer, providing a direct measure of how the model's focus shifts over time.

We compute a normalized attention score for each frame sequence through:

$$A_t = \text{median}_{h \in H}(a_h^t)$$

where a_h^t represents the attention values from head h at time t. The median operation across attention heads provides a statistically robust measure of the model's temporal focus, reducing the impact of outlier attention heads.

Our temporal analysis reveals that adversarial attacks fundamentally alter the model's ability to track actions through time. The clean video attention pattern, shown by the blue line in Fig. 44, exhibits smooth progression with clear peaks corresponding to key action moments. In contrast, the adversarial example (orange dashed line) shows erratic temporal attention with misaligned peaks and sudden variations, indicating a loss of temporal coherence.

To quantify the similarity between clean and adversarial attention patterns, we employ the Pearson correlation coefficient [89]:

$$\rho = \frac{\text{cov}(A^c, A^{adv})}{\sigma_{A^c}\sigma_{A^{adv}}}$$

Where A^c and A^{adv} represent the attention sequences for clean and adversarial videos, cov denotes covariance, and σ represents standard deviation. This coefficient ranges from -1 to 1, with 1 indicating perfect correlation, 0 indicating no linear correlation, and -1 indicating perfect negative correlation. For example, the observed correlation of 0.071 and 0.133 in Fig. 44 shows that temporal attention is significantly disrupted.

A notable pattern emerges in the temporal attention oscillation characteristics. As shown in Fig. 44, the adversarial video (orange dashed line) exhibits substantially larger amplitude fluctuations compared to the clean video (blue line). This increased oscillation indicates that the adversarial attack causes the model to shift its focus rapidly between different temporal regions rather than maintaining stable attention on action-relevant frames. The frequency of these oscillations increases over time, suggesting that the adversarial perturbations have a cumulative effect on the model's temporal attention mechanism. This

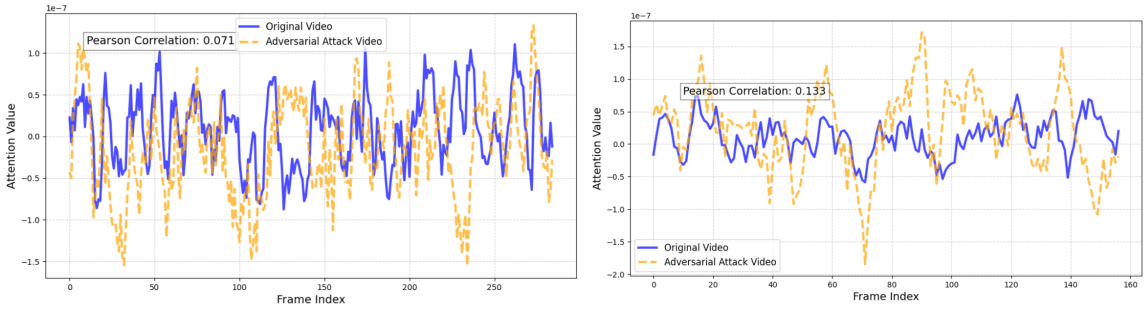


Figure 44: Temporal analysis of attention patterns comparing clean and adversarially perturbed videos. The blue line represents the original video's attention progression. The orange dashed line demonstrates the adversarial video's attention pattern, revealing significant disruption in temporal coherence. The Pearson correlation coefficient quantifies the temporal attention deviation.

progressive destabilization of temporal attention patterns reveals a fundamental vulnerability in how transformer-based video models maintain temporal coherence under adversarial attacks.

Spatiotemporal Analysis and 3D Visualization

To comprehensively analyze the spatiotemporal impact of adversarial attacks, we develop a novel 3D visualization approach that treats attention patterns as a continuous volume across both space and time. For each video sequence, we extract attention maps using TimeSformer's final attention layer:

$$A(t, u, v) = \text{median}_{h \in H}(a_h^t(u, v))$$

where $a_h^t(u, v)$ represents the attention value at spatial position (u, v) for head h at time t. The median operation across attention heads provides a robust measure of the model's focus.

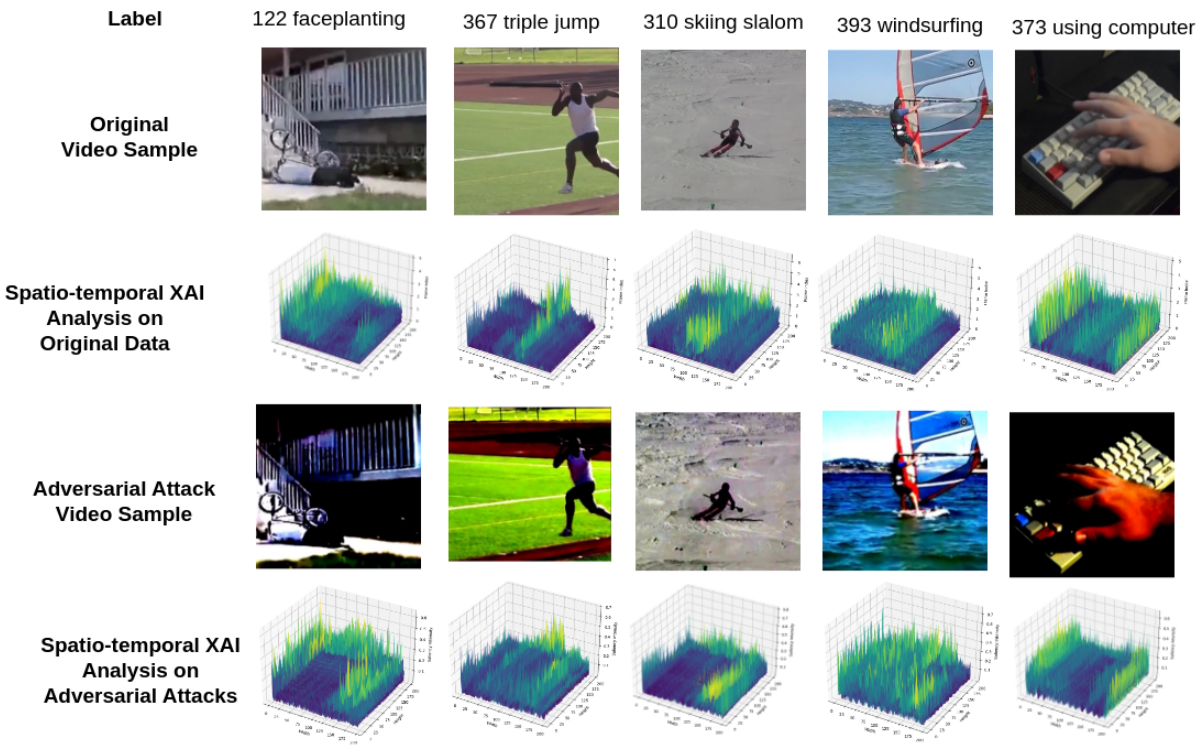


Figure 45: Comparative analysis of spatiotemporal attention patterns through 3D visualization. From top to bottom: (1) Action label from Kinetics-400 dataset, (2) Sample frames from original video showing the action sequence, (3) 3D visualization of clean video attention patterns, (4) Sample frames from adversarially perturbed video, and (5) 3D visualization of attention patterns under adversarial attack. **3D Visualization Axes**: X-axis represents spatial width pixel positions, Y-axis represents spatial height pixel positions, and Z-axis represents accumulated attention intensity. **Color Encoding**: Yellow indicates high temporal attention accumulation (consistent focus across frames), blue indicates low accumulation (sporadic attention), with intermediate colors representing varying degrees of temporal consistency. The comparison reveals how adversarial attacks (row 5) disrupt the temporal consistency of spatial attention, creating scattered attention accumulation patterns compared to the coherent, concentrated attention peaks in clean videos (row 3).

We normalize these attention values to the [0,1] range:

$$A_{\text{norm}}(t, u, v) = \frac{A(t, u, v) - \min(A)}{\max(A) - \min(A)}$$

The 3D visualization methodology constructs a spatiotemporal attention volume where each axis represents a distinct dimension of the video transformer's attention mechanism. Specifically: **X-axis (horizontal)**: Spatial width dimension (u), representing the horizontal pixel positions within each video frame; **Y-axis (vertical)**: Spatial height dimension (v), representing the vertical pixel positions within each video frame; **Z-axis (depth)**: Attention values accumulation, representing the cumulative attention intensity that each spatial location (u, v) receives across the entire temporal sequence. The Z-axis value for each spatial position (u, v) is computed as the temporal accumulation of attention values:

$$Z(u, v) = \sum_{t=1}^{T} A_{\text{norm}}(t, u, v) \tag{26}$$

where $A_{\text{norm}}(t, u, v)$ represents the normalized attention value at spatial position (u, v) in frame t, and T is the video time length.

This accumulation mechanism reveals the temporal consistency of spatial attention. High Z-values represent the spatial locations that consistently receive strong attention across multiple frames, straightforwardly indicating persistent focus regions. Low Z-values represent spatial locations that receive sporadic or weak attention, indicating a consistently irrelevant region. This 3D representation enables analysis of which spatial regions maintain consistent attention across time versus those that show sporadic focus, revealing how adversarial perturbations disrupt the temporal consistency of spatial attention patterns.

Our spatiotemporal analysis reveals systematic disruption patterns in the adversarial model's attention mechanism. The clean attention volume, visualized in the third row of Fig. 45 as a 3D heatmap where yellow indicates high attention values and blue indicates

low values, maintains a coherent focus on action-relevant regions with smooth temporal transitions across consecutive frames. The adversarial attention patterns (the fifth row in Fig. 45) demonstrate scattered distributions with abrupt temporal changes, indicating the disruption of the model's spatiotemporal attention mechanisms. By comparing the attention volumes between clean and adversarial videos, we observe both spatial and temporal attention patterns. The comparison between rows 2 and 4 shows that while the visual differences in video frames may be subtle to human eyes, the resulting changes in attention patterns (third and fifth rows) are substantial, demonstrating how adversarial attacks impact the model's attention.

Our XAI analysis quantifies the video transformer attention under adversarial attacks. Through spatial analysis with SSIM metrics, we measured median scores of 0.353 across 20,000 analyzed videos, revealing significant shifts in attention distribution. The temporal correlation analysis also demonstrated observable disruption, represented by the Pearson correlation coefficient. The novel 3D visualization methodology provides interpretable evidence of how adversarial perturbations simultaneously impact a model's spatial and temporal attention.

5.2.4 Empirical Study on Effectiveness of Existing Adversarial Attacks

Adversarial attacks pose a significant challenge to vision models [198]. While CNN-based video models have been shown to be vulnerable to adversarial perturbations, the resilience of transformer-based video models remains less explored. Unlike image models that process frames independently, video models leverage temporal dependencies, which may also impact adversarial transferability and robustness. This section investigates the effectiveness of adversarial attacks on transformer-based video models, focusing on Common Corruption

Benchmark, Cross-model frame-wise attacks, sparse key-frame attacks, and query-based attacks.

In experiments, we use the Kinetics dataset, the latest widely used benchmark for human action recognition in videos [95]. The Kinetics dataset is more challenging and diverse than the previous common benchmark datasets, such as the 51-class HMDB51 and 101-class UCF101. The Kinetics dataset consists of 400 action classes, covering a diverse range of activities, including sports, daily interactions, and object manipulations. Each class contains short video clips sourced from YouTube, ensuring diversity in background, lighting, and motion dynamics. We execute every experiment with a class-balanced evaluation dataset, which contains a total of 20,000 video samples.

Common Corruption Benchmark

In image classification tasks, the Common Corruptions Benchmark (CCB) [81] serves as an evaluation benchmark for assessing model robustness against real-world distortions. Previous research has documented the vulnerability of CNN-based models to these corruptions, with significant performance degradation observed under various types of perturbations [81].

We extend these CCB evaluations to transformer-based image models. While these studies revealed that transformer architectures also experience significant performance drops under natural corruptions (as shown in Table 17). Only certain models, including Swin and ViT, demonstrated slight resilience compared to CNN architectures. CCB corruptions significantly impact transformer-based image models, with performance degradations ranging from 10.6% to 72.2% across different architectures.

However, the impact of common corruption on video transformer models remains unexplored. This gap is particularly significant given the fundamental differences in how video models process information. Image models process frames independently, making

Table 17: Impact of Common Corruptions on Image Model Performance

Model Type	Model	F1 Score (Original)	F1 Score (CCB attacked)
CNN	ConvNext [121]	0.814	0.674
	ResNet [79]	0.748	0.539
Transformer	CVT [210]	0.764	0.358
	Swin [120]	0.834	0.648
	SegFormer [212]	0.651	0.181
	ViT [47]	0.743	0.664

them vulnerable to frame-level corruption. Video models leverage joint spatial and temporal dependencies. Transformer architectures employ attention mechanisms that aggregate information across time, potentially providing enhanced resilience against CCB-induced distortions



Figure 46: Examples of common corruptions applied to video frames. From left to right: (a) Original frame, (b) Gaussian Noise, (c) Shot Noise, (d) Motion Blur, and (e) Zoom Blur. The corruptions are applied consistently across all frames in each video sequence to simulate real-world distortions.

We test a subset of four representative corruptions from CCB [81], as shown in Fig. 46, selected based on their prevalence in real-world video scenarios. Gaussian Noise simulates sensor noise and environmental graininess that commonly occurs in video capture. Shot Noise models high-frequency disturbances typically encountered in low-light conditions or when using high-ISO settings. Motion Blur represents frame-wise blurring effects caused by camera movement or subject motion during recording. Finally, Zoom Blur simulates

out-of-focus effects that are common in video recording, particularly during dynamic camera operations. These corruptions are chosen to comprehensively test the model's resilience against different types of real-world visual distortions.

In our experimental setup, these corruptions are applied consistently across all frames in each video sequence, maintaining uniform perturbation levels throughout the temporal dimension. This approach allows us to evaluate both the immediate impact on spatial features and the potential compensatory effects of temporal attention mechanisms.

Table 18: Impact of Common Corruptions on TimeSformer Video Model Performance

Corruption Type	**Performance Metrics**			
	Accuracy	Precision	Recall	F1 Score
Clean (Baseline)	0.543	0.603	0.543	0.572
Gaussian Noise	0.485	0.573	0.485	0.525
Motion Blur	0.523	0.593	0.523	0.555
Shot Noise	0.533	0.595	0.533	0.562
Zoom Blur	0.456	0.539	0.456	0.494

The results in Table 18 demonstrate several key findings regarding the robustness of video transformers to common corruptions. While all corruption types lead to some performance degradation, the impact varies across different types of perturbations. Shot Noise and Motion Blur show relatively minor impacts on model performance, suggesting video transformers maintain robustness against these types of perturbations. Zoom Blur emerges as the most detrimental corruption, causing the largest decrease across all performance metrics.

However, despite causing some performance degradation, none of these corruptions

consistently mislead the model into incorrect predictions. The model still maintains reasonable classification accuracy even under these perturbations, indicating that common corruptions alone cannot serve as effective adversarial attacks. This suggests that corruption-based approaches are insufficient for compromising video transformer models, necessitating more sophisticated attack strategies that specifically target the model's spatiotemporal attention mechanisms. These results indicate that the temporal modeling capabilities of video transformers provide natural resilience against common corruptions, marking a significant advancement over image-based architectures. Our new spatiotemporal adversarial attack methods fill this gap.

Frame-wise and Sparse Transferable Attack

The recent works about the transferable attack on video models [198] are listed in Table 16 in the Section of Related Works. The recent work demonstrates the feasibility of cross-model adversarial attacks [206]. We evaluate these two approaches: frame-wise transfer and sparse transfer attacks.

In Frame-wise Transfer, we apply adversarial perturbations generated from an ImageNet-trained ResNet-101 model to each video frame sequentially. This approach serves as another baseline in our evaluation framework. First, it represents the most straightforward transferability test of image attacks to video domains. Second, it enables direct comparison with prior work on CNN-based video models, where this method has proven effective. Third, by testing this approach on transformer architectures, we can quantify how their unique attention mechanisms respond differently to perturbations designed for convolutional networks.

For Sparse Key-frame Transfer, we propose an optical flow-guided approach to identify and perturb keyframes efficiently. The process consists of three main stages. (1) `Optical Flow Estimation`. We compute dense optical flow between consecutive frames using

the Farneback algorithm in OpenCV. This provides a detailed motion field that captures the temporal dynamics of the video sequence. (2) `Motion-based Frame Selection`. Keyframes are identified based on significant changes in optical flow magnitude. Frames with larger motion variations are prioritized as they typically correspond to important action transitions or significant visual changes. (3) `Adaptive Perturbation`. FGSM perturbations are applied selectively to these keyframes.

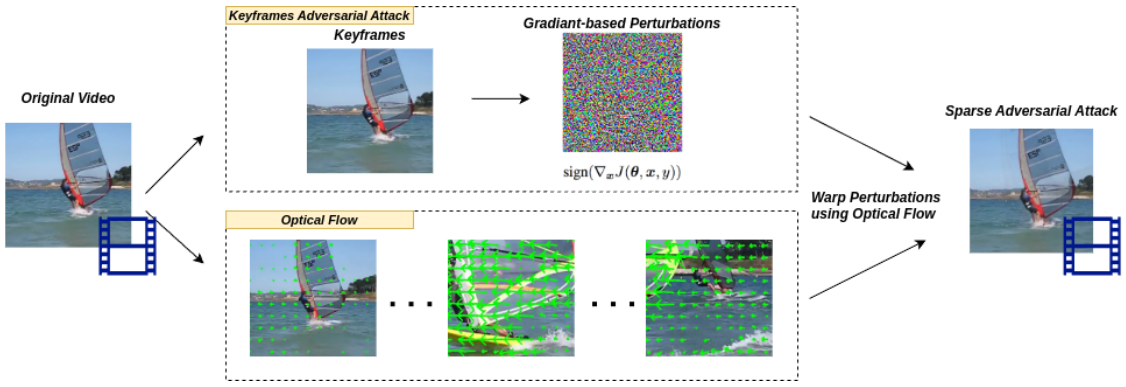


Figure 47: Flow chart of our sparse attack implementation. The attack pipeline consists of two parallel branches: (1) A gradient-based perturbation branch that applies FGSM to selected keyframes. (2) An optical flow analysis branch that identifies frames with significant motion. The outputs from both branches are combined to generate the final sparse adversarial attack that maintains temporal consistency while reducing computational overhead compared to frame-wise attacks.

As illustrated in Fig. 47, our sparse attack implementation follows a two-branch architecture. The first branch focuses on generating gradient-based perturbations using FGSM on selected keyframes, while the second branch performs optical flow analysis to identify frames containing significant motion information. This dual-branch approach ensures that our adversarial perturbations target the frames with significant motion information. The experimental results are presented in Table 19 for discussion.

Query-based V-BAD Attack

We evaluate V-BAD [91], a state-of-the-art black-box attack method for video models. As shown in Fig. 48, this method combines transferable perturbations from image models with partition-based gradient estimation, which reduces the required number of queries. V-BAD is a three-stage approach: generating tentative perturbations from pre-trained image models, applying partition-based rectification, and performing gradient approximation.

To evaluate V-BAD's effectiveness, we employ original R3D (3D ResNet) [78] as the estimator model. R3D extends the ResNet architecture by incorporating 3D convolutions that simultaneously operate across spatial and temporal dimensions. This architectural choice enables direct spatiotemporal feature learning from video data through an integrated framework of 3D convolutions, residual connections, and pooling operations. The model processes video inputs of size $(T \times 224 \times 224 \times 3)$, where T represents the temporal dimension, typically set to 16 frames. Our experimental evaluation uses the Kinetics-400 dataset. The perturbation bound ϵ is also set to 0.01.

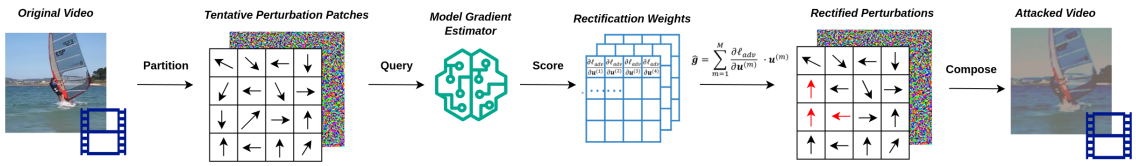


Figure 48: The V-BAD attack [91] pipeline demonstrates a systematic approach to compromising R3D models. The method employs partition-based gradient estimation, followed by refined perturbation application across video frames.

Our experimental results reveal V-BAD's effectiveness against R3D models. They demonstrate that simply adding temporal dimensions to convolutional architectures creates a well-designed adversarial attack. This observation directly validates our approach of developing attack methods that explicitly target integrated spatiotemporal features, rather than treating temporal modeling as a separate component. The persistent vulnerability of R3D models, despite their temporal processing capabilities, underscores the necessity of

our joint spatiotemporal attack framework that specifically addresses the unified nature of space-time feature learning in advanced video models.

Adversarial Attack Performance Discussion

In addition to standard classification metrics, including Precision, Recall, and F1 Score, we introduce two specialized metrics to provide an analysis of adversarial attack effectiveness.

The Attack Success Rate (ASR) quantifies the percentage of samples where the adversarial perturbation successfully causes misclassification:

$$\text{ASR} = \frac{|\{x_i : f(x_i) = y_i \wedge f(x_i^{adv}) \neq y_i\}|}{|\{x_i : f(x_i) = y_i\}|} \tag{27}$$

where $f(\cdot)$ represents the model prediction, x_i is the original sample, x_i^{adv} is the adversarial example, and y_i is the ground truth label.

We analyze the impact of adversarial attacks on model prediction confidence through confidence score analysis:

$$C(x) = \max(\text{softmax}(f(x))) \tag{28}$$

where $C(x)$ represents the maximum softmax probability (confidence score) for sample x. Lower confidence scores indicate greater uncertainty and potential vulnerability, while maintaining high confidence in incorrect predictions suggests the attack generates highly deceptive perturbations.

For all of the attack methods we introduced in the previous subsections, we summarize the results in Table 19. The TimeSformer model achieves 0.544 accuracy on clean data. Frame-wise attacks reduce accuracy to 0.473, demonstrating less effectiveness. Parse Frame attacks maintain higher precision (0.606) through selective frame targeting, achieving comparable accuracy (0.531) with reduced computational cost.

V-BAD significantly outperforms these previous approaches, reducing model precision to 0.326 and accuracy to 0.192. This effectiveness comes from its dynamic partition-based gradient estimation and transferable perturbations from pre-trained image models. The method's query efficiency validates the advantages of Natural Evolution Strategies over pixel-level gradient estimation.

The ASR metric provides direct quantification of adversarial effectiveness. As shown in Table 19, our joint spatiotemporal method achieves the highest ASR of 76.30%, representing an advancement over existing approaches. Frame-wise attacks achieve only a 20.57% success rate, demonstrating the limited effectiveness of independent frame perturbations against video transformers' temporal integration mechanisms. Besides, sparse attacks also show remarkably low effectiveness with merely 2.71% ASR, suggesting that selective keyframe perturbation fails to disrupt the model's spatiotemporal attention sufficiently. This finding also supports our research motivation, which we learn that isolated frame modifications have limited attack effectiveness [87]. The complex and robust video models require coordinated spatiotemporal perturbations rather than frame-based perturbations.

V-BAD, as recent SOTA study [91], demonstrates substantial effectiveness with 70.74% ASR, confirming its status as a strong baseline for video adversarial attacks. V-BAD incorporates temporal modeling through 3D convolutions that operate across both spatial and temporal dimensions simultaneously. The method's partition-based gradient estimation enables efficient exploitation of these 3D convolutional features. However, V-BAD is designed for CNN-based architectures rather than transformer architectures. Our method's 5.56 percentage point improvement (76.30% vs 70.74%) represents progress in exploiting video transformer vulnerabilities specifically.

Our method achieves both the lowest accuracy (0.139) and highest ASR (76.30%). With transformer-based video models increasingly adopted in safety-critical applications,

the demonstrated vulnerability to our spatiotemporal attacks underscores the urgent need for specialized defense mechanisms.

Table 19: Performance comparison of different adversarial attack methods

Method	Precision	Recall	F1-Score	Accuracy	ASR (%)
Original	0.610	0.544	0.575	0.544	-
Frame-wise	0.580	0.473	0.521	0.473	20.57
Sparse	0.606	0.531	0.566	0.531	2.71
V-BAD	0.326	0.192	0.242	0.192	70.74
Our Method	**0.235**	**0.139**	**0.175**	**0.139**	**76.30**

We provide the visualization of how different adversarial attacks affect model confidence through violin plot analysis. As shown in Fig. 49, each violin displays the probability density distribution of confidence scores, where the width at any confidence level indicates the relative frequency of samples achieving that confidence value.

Then, we discuss the results from each specific column. Clean samples exhibit a high-confidence distribution with a median of 0.732. The violin shape shows substantial density in the upper confidence range (0.7-1.0), indicating that the TimeSformer model typically maintains strong certainty in its correct predictions. This baseline establishes the model's natural confidence characteristics under normal operating conditions.

Frame-wise perturbations reduce the median confidence to 0.636, representing a moderate degradation in model certainty. The violin distribution becomes broader compared to the original, showing increased spread across confidence levels. However, the distribution maintains considerable density in higher confidence regions, reflecting the limited effectiveness of independent frame modifications against video transformers' temporal integration mechanisms.

Sparse attacks demonstrate minimal impact on confidence distribution, with the median remaining at 0.729, nearly identical to the original baseline. The violin shape closely mirrors the clean distribution, providing strong evidence that selective keyframe perturbation

fails to significantly compromise model confidence. This observation directly correlates with the extremely low ASR of 2.71%, confirming that isolated frame modifications are insufficient against transformer-based spatiotemporal processing.

V-BAD produces substantial confidence degradation, reducing the median to 0.394. The violin reveals a distinctive bimodal distribution pattern, with notable density concentrations in low-confidence regions. This shape indicates that V-BAD reduces the model's confidence in a significant portion of samples, while some predictions retain moderate confidence levels.

Our proposed method achieves the most confidence degradation, with the median dropping to 0.388. The violin shape demonstrates a predominantly concentrated distribution in the low-confidence region, with substantially reduced density in higher confidence ranges (at the top of the column, near 1.0). These results indicate consistent undermining of model confidence across diverse video samples, creating a more uniform distribution of low-confidence predictions compared to other methods.

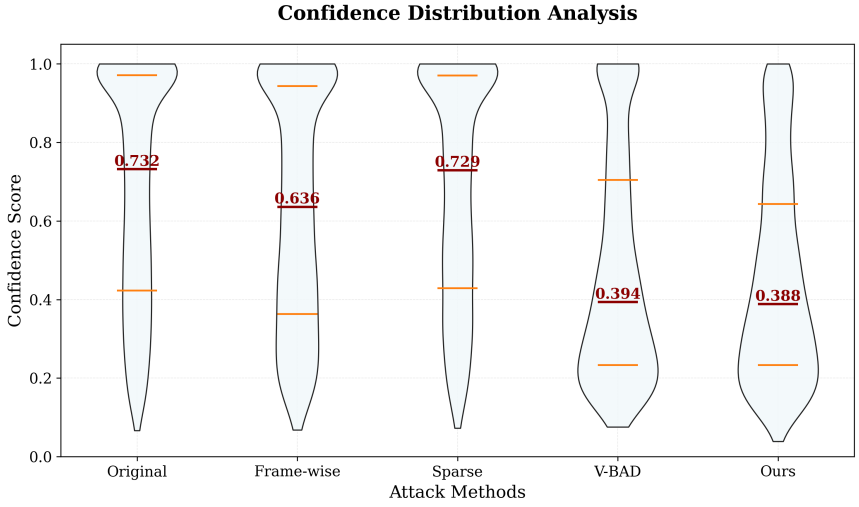


Figure 49: Confidence score distribution analysis using violin plots across attack methods.

In summary, the confidence distributions reveal nuanced differences in attack mechanisms. While both V-BAD and our method achieve similar low median confidence levels

(0.394 vs 0.388) and superior ASR performance, the distribution patterns reveal critical distinctions in their effectiveness.

The violin plot analysis demonstrates that confidence score distribution provides valuable complementary information to accuracy metrics. Methods that achieve lower median confidence scores generally correlate with higher attack success rates, validating the coherence of our multi-metric evaluation framework. However, the distribution shape and density patterns offer additional insights into attack consistency and mechanism effectiveness that median values alone cannot capture.

Critically, our method shows substantially reduced density in the high-confidence region (the top of each column), indicating that the model loses confidence across the majority of data samples. In contrast, other methods maintain relatively high density at the top, suggesting they affect only subsets of vulnerable samples. This confidence analysis, combined with our superior ASR (76.30%), demonstrates that coordinated spatiotemporal perturbations more effectively exploit transformer attention vulnerabilities than the selected approaches.

Frame-wise attacks show limited effectiveness with only 20.57% ASR, while sparse attacks achieve minimal impact with merely 2.71% ASR, both failing to disrupt transformer attention mechanisms effectively. The query-based V-BAD method demonstrates substantial effectiveness (70.74% ASR), establishing a strong baseline for video adversarial attacks. However, our proposed spatiotemporal approach significantly outperforms all existing methods, achieving the highest ASR of 76.30% and lowest residual accuracy of 0.139. This superior performance stems from our method's explicit targeting of transformer attention mechanisms across both spatial and temporal dimensions simultaneously, proving more effective than existing approaches designed for CNN-based video models.

5.2.5 Implications for Model Robustness Enhancement

This section explores the practical study of our findings and proposes concrete approaches for enhancing model security in real-world deployments. Based on our analysis of spatiotemporal attacks on video transformers, we investigate defense mechanisms that can mitigate the vulnerabilities.

Adversarial Training-Based Defense Setting Strategy

Our spatiotemporal attack method achieves the highest Attack Success Rate against video transformer models. We now investigate potential defense strategies to mitigate these vulnerabilities. We explore whether adversarial training using attack-generated examples can reduce the observed ASR and enhance model robustness. We implement a targeted defense strategy through adversarial training [67, 127], using adversarial data generated by each attack method to enhance model robustness during the training phase.

Our adversarial training procedure integrates original and adversarial examples through a controlled training regimen. To ensure fair comparison across different attack methods, we maintain training configurations for all experiments. Each defense model undergoes training for exactly ten epochs, with convergence typically observed between the third and fifth epochs. We implement early stopping based on adversarial validation accuracy stabilization. The training dataset construction follows a balanced approach, maintaining a 50:50 ratio between clean and adversarial examples in each batch. For each attack method evaluated (Spatiotemporal, V-BAD, Frame-wise, and Sparse), we generate adversarial examples using a consistent perturbation bound of $\epsilon = 0.01$. The complete training set comprises 10,000 video samples from Kinetics-400, with 5,000 clean and 5,000 adversarial examples balanced across all 400 action categories.

The optimization process employs the AdamW optimizer [125]. Each training batch is

equally divided between clean and adversarial samples. Mixed precision training accelerates computation without affecting model accuracy. We incorporate TRADES regularization [224]. The TRADES objective balances natural accuracy with adversarial robustness by minimizing the divergence between predictions on clean and perturbed samples.

Defense Effectiveness Analysis

The evaluation encompasses our proposed spatiotemporal attack, V-BAD [91], frame-wise attacks [206], and sparse [204] attacks. The evaluation is conducted on a validation set of 3,000 video samples from the Kinetics-400 dataset, maintaining class balance across all 400 action categories.

Table 20 presents the quantitative results of adversarial training defense across multiple attack methodologies. Our spatiotemporal attack experiences the largest defense improvement, with ASR decreasing from 76.30% to 33.10%, resulting in a 43.20 percentage point defense gain. This demonstrates that adversarial training can effectively mitigate the specific vulnerability patterns exploited by the training attack method. The effectiveness is expected, given that the adversarial training dataset consists of examples generated using our attack methodology.

Table 20: Adversarial Training Configuration and Defense Effectiveness Against Various Attack Methods

Attack Method	**Adv. Training Config.**		**Attack Success Rate (ASR)**		
	Trained Type	**Clean:Adv**	**Original**	**Defended**	**Δ ASR**
Spatiotemporal (Ours)	Spatiotemporal	50:50	76.30%	33.10%	+43.20%
V-BAD	V-BAD	50:50	70.74%	34.17%	+36.57%
Frame-wise	Frame-wise	50:50	20.57%	15.62%	+4.95%
Sparse	Sparse	50:50	2.71%	4.51%	-1.80%

Note: All models trained on Kinetics-400. **Clean:Adv** = proportion of clean vs adversarial samples. **Trained Type:** Each model is adversarially trained using its corresponding attack method ($\epsilon = 0.01$). **Δ ASR** = Original ASR - Defended ASR (positive values indicate successful defense).

V-BAD attacks test also show substantial defense improvement when trained with V-BAD-generated adversarial examples, with ASR decreasing from 70.74% to 34.17%, achieving a defense gain of 36.57 percentage points. However, the defense effectiveness varies significantly for simpler attack methods. Frame-wise attacks show modest improvement, with ASR decreasing from 20.57% to 15.62%, resulting in a defense gain of 4.95 percentage points. Considering the original ASR is not substantial, the defense gain is reasonably limited. Sparse attacks exhibit an anomalous result, with ASR increasing from 2.71% to 4.51%. The original ASR is low. Then, the adversarial training on adversarial data creates vulnerabilities in the model.

These results demonstrate that adversarial training can partially defend against the attacks. While each method shows improvement against its own attack type (except for the anomalous sparse attack case), the variation in defense gains highlights the challenge of developing universal robustness. This pattern aligns with established findings regarding the robustness-accuracy trade-off in adversarial machine learning [156, 224].

5.2.6 Summary

This study advances our understanding of adversarial vulnerabilities in transformer-based video models through evaluation. Our investigation reveals that while video transformers exhibit some inherent resilience against conventional attack approaches due to their temporal modeling capabilities, they remain susceptible to our adversarial attack method that specifically targets their spatiotemporal attention mechanisms.

We make three primary contributions to the field. First, we develop a novel joint spatiotemporal attack framework specifically designed for video transformer architectures. This approach synchronizes perturbation generation across precise multi-frame temporal windows that align with the transformer's native processing mechanisms. By directly manipulating the model's integrated spatiotemporal features through gradient computations,

our method achieves substantial performance degradation while maintaining computational efficiency through linear scaling.

Second, we establish XAI analysis for understanding adversarial impacts on video transformers. Our approach combines spatial analysis using structural similarity metrics with temporal correlation analysis to provide a complete picture of how adversarial perturbations affect model behavior. The introduction of a novel 3D visualization methodology unifies these analyses, offering an interpretable representation of how attacks simultaneously compromise both spatial focus within frames and temporal consistency across sequences.

Third, our empirical study provides valuable insights into the comparative effectiveness of different attack methodologies against video transformer models. The findings demonstrate that frame-wise and common corruption approaches achieve limited success against these architectures, while methods targeting the integrated spatiotemporal dependencies prove significantly more effective. This underscores the fundamental difference between CNN-based and transformer-based video models in terms of their vulnerability profiles.

These methodological advances collectively establish a foundation for developing more robust video understanding systems. Our research demonstrates that effective security measures for video transformers must address both spatial and temporal dimensions simultaneously, as the integration of these features within transformer architectures creates unique vulnerability patterns that differ significantly from those observed in conventional models. The analytical tools and attack methodologies introduced in this work provide essential resources for future research on adversarial robustness in complex spatiotemporal models.

Chapter 6

Conclusion

Motivated by the increasing demand for trustworthy AI in real-world deployments, this research has addressed critical gaps in the current XAI landscape, focusing on enhancing interpretability, robustness, and reproducibility across both cloud-based services and open-source machine learning models.

6.1 Summary of Contributions

This thesis delivers four interconnected contributions.

1. **Contribution 1: XAI Service Framework for Cloud and Open-Source Models:** We developed XAIport, a microservice architecture that standardizes explainable AI deployment across diverse computing environments. The framework provides three key capabilities. First, it enables integration with commercial cloud AI services (Azure, Google Cloud, AWS) through model-agnostic explanation techniques, allowing organizations to audit proprietary black-box models without requiring internal access. Second, it supports deployment of open-source models from repositories through standardized REST APIs. Third, it incorporates comprehensive provenance tracking and reproducibility mechanisms, ensuring that every explanation can be

traced, validated, and recreated. Empirical validation demonstrates that this framework reduces XAI integration overhead compared to traditional ad-hoc implementations.

2. **Contribution 2: Automated Evaluation Pipeline for XAI Consistency and Quality Assessment** We established the pipeline for evaluating explanation reliability across different XAI methods and model architectures. The methodology introduces consistency metrics that measure explanation stability under controlled perturbations and agreement across different attribution methods. Our evaluation pipeline tests XAI approaches across multiple dimensions. The framework incorporates automated quality assessment protocols that evaluate explanation faithfulness, monotonicity, and computational efficiency. Through testing across cloud AI services and open-source models, we revealed quality differences between popular XAI methods. The pipeline also demonstrated that data augmentation techniques improve explanation consistency, providing actionable guidance for improving XAI deployment quality.

3. **Contribution 3: Spatio-Temporal Attention Attribution (STAA) for Video Transformer Explanation** We introduced STAA, a XAI method specifically designed for video transformer models that simultaneously captures spatial and temporal feature importance. This addresses the research gap in current XAI capabilities, as existing methods fail to explain spatiotemporal dependencies. STAA operates by directly extracting and aggregating attention weights from transformer architectures. The method provides unified explanations that reveal both spatial importance within individual frames and temporal relevance across video sequences. Performance evaluation on Kinetics-400 and Something-Something V2 datasets demonstrates STAA's superiority across multiple metrics. The method achieves faithfulness scores of 0.844

and monotonicity scores of 0.850, significantly outperforming model-agnostic approaches. STAA reduces explanation generation time by over 97%, enabling real-time video explanation with sub-150ms latency. This computational efficiency makes STAA practical for edge computing applications and real-time video analysis systems that traditional XAI methods cannot reach.

4. **Contribution 4: XAI-Guided Adversarial Attack Framework for Video Transformer Security Assessment** We developed an adversarial attack methodology that simultaneously targets both predictive performance and explanation stability in video transformer models. Our attack framework employs a joint spatiotemporal perturbation strategy that exploits the integrated attention mechanisms of video transformers. The experiments incorporate XAI analysis tools to quantify attack effectiveness through both prediction accuracy degradation and explanation stability metrics. Experimental results on video transformer architectures demonstrate the framework's effectiveness with attack success rates of 76.30% on Kinetics-400, outperforming existing approaches. Spatial analysis reveals systematic disruption of attention patterns with median SSIM scores of 0.353. The framework also includes 3D visualization tools that provide an understanding of how attacks compromise both spatial focus and temporal consistency.

6.2 Limitations and Considerations

While this thesis presents significant advances in explainable AI frameworks, several limitations warrant acknowledgment and provide context for future research directions.

6.2.1 Computational and Scalability Constraints

The computational overhead of comprehensive XAI analysis remains substantial despite our optimization efforts. Although we achieved significant latency reductions for specific methods, the broader framework still imposes non-trivial computational costs when deployed at scale. Production environments processing millions of predictions daily may find the cumulative overhead challenging, particularly when multiple explanation methods are employed simultaneously for cross-validation. This limitation becomes more pronounced for resource-constrained edge devices or real-time applications where explanation generation must compete with prediction tasks for computational resources.

Our experimental validation, while comprehensive within its scope, was necessarily limited to specific datasets and model architectures. The framework was primarily tested on established benchmarks including ImageNet for vision tasks, Kinetics-400 for video understanding, and selected tabular datasets. While these represent diverse application domains, the generalizability to specialized domains such as medical imaging, scientific computing, or industrial IoT applications requires further validation. The computational requirements may scale differently with domain-specific data characteristics, potentially requiring architectural adjustments for optimal performance.

6.2.2 Explanation Quality and Human Factors

Our evaluation of explanation quality relies on proxy metrics that may not fully capture human interpretability needs. While metrics such as faithfulness, monotonicity, and consistency provide quantitative assessment capabilities, they do not guarantee that generated explanations will be meaningful or actionable for end-users. The gap between mathematical rigor and practical utility remains a fundamental challenge that our framework partially addresses through multiple explanation methods, but does not completely resolve. User studies across diverse populations and application domains would be necessary to fully

validate the practical effectiveness of our approach.

Furthermore, the framework currently lacks sophisticated mechanisms for adapting explanations to different user expertise levels. While we provide multiple explanation methods, the selection and presentation of appropriate explanations for specific user groups—from ML experts to domain specialists to end-users—remains largely manual. This limitation may hinder adoption in scenarios requiring automated explanation generation for diverse stakeholder groups with varying technical backgrounds and information needs.

6.2.3 Black-box Model Limitations

The framework's effectiveness for proprietary cloud AI services is inherently limited by the black-box nature of these systems. Without access to model internals, our explanations must rely on input-output behavior analysis, which may miss important architectural biases or training data influences. This limitation is particularly relevant for large language models and other complex architectures where post-hoc explanations may provide incomplete or potentially misleading insights about actual model reasoning processes.

Our approach to explaining cloud AI services through API interactions, while practical and widely applicable, cannot capture certain critical aspects of model behavior. For instance, we cannot directly analyze the impact of specific training data samples, identify potential backdoors embedded during training, or fully understand the effect of proprietary optimization techniques employed by cloud providers. These limitations become more pronounced as cloud AI services increasingly employ ensemble methods or hybrid architectures that combine multiple models, making the attribution of decisions to specific components impossible without internal access.

6.2.4 Adversarial Robustness Scope

Our adversarial robustness analysis focused primarily on video transformer models, and the generalizability of these findings to other architectures and modalities requires further investigation. The spatiotemporal attack methods we developed exploit specific architectural properties that may not translate directly to other model types, potentially limiting the broader applicability of our security insights. While we demonstrated the effectiveness of our joint spatiotemporal attack method against transformer-based video models, the robustness characteristics of emerging architectures such as neural radiance fields, diffusion models, or multimodal transformers remain unexplored within our framework.

Additionally, our adversarial evaluation primarily considered untargeted attacks aimed at degrading model performance. Targeted attacks, which seek to induce specific misclassifications, may reveal different vulnerability patterns that our current analysis does not capture. The framework's ability to detect and explain sophisticated adversarial manipulations, particularly those designed to evade explanation-based detection methods, requires further development and validation.

These limitations, while significant, do not diminish the contributions of this research but rather highlight the complexity of achieving truly explainable AI systems. They provide important context for interpreting our results and identify concrete directions for future research. Addressing these limitations will require continued collaboration between researchers, practitioners, and policymakers to develop more sophisticated frameworks that balance technical rigor, practical utility, and societal needs.

6.3 Future Work

The framework and methodologies presented in this thesis establish a foundation for trustworthy AI systems, yet several important research directions remain to be explored. Rather

than pursuing incremental improvements across numerous dimensions, future work should focus on three interconnected research trajectories that address fundamental challenges in explainable AI deployment.

6.3.1 Advancing Toward Causal and Multimodal Explanations

Current XAI methods, including those integrated into our framework, primarily focus on correlational relationships between features and predictions. The natural evolution of this work involves transitioning from correlation-based explanations to causal inference mechanisms that can identify true causal relationships in model decision-making. This advancement requires developing new theoretical foundations that combine causal discovery algorithms with deep learning architectures, potentially leveraging recent advances in structural causal models and interventional analysis. Such capabilities would enable practitioners to distinguish between spurious correlations and genuine causal factors, significantly enhancing the reliability of AI-driven decisions in critical domains such as healthcare diagnostics and financial risk assessment.

Simultaneously, the increasing prevalence of multimodal AI systems demands explanation methods that can coherently interpret cross-modal interactions. Future research should develop unified explanation frameworks capable of analyzing how models integrate information across text, vision, audio, and structured data modalities. This involves creating novel attention mechanisms that preserve interpretability while maintaining computational efficiency, as well as developing visualization techniques that can effectively communicate complex multimodal relationships to diverse stakeholders. The challenge extends beyond technical implementation to include human factors research on how users comprehend and act upon multimodal explanations.

6.3.2 Extension to Video Language Models

The convergence of video understanding and natural language processing represents a transformative paradigm shift in artificial intelligence [223]. Video-language models enable sophisticated tasks including video captioning, visual question answering, and video-grounded dialogue, while video-capable Large Language Models (LLMs) push the boundaries of multimodal understanding [223]. This subsection discuss how STAA can be extended to explain these emerging models in the future.

Vision-Language Pre-trained Models

Vision-language pre-trained models employ separate encoders for video and text modalities, connected through cross-modal attention layers that learn to align visual and textual features [223]. The training process typically relies on contrastive learning objectives that encourage the model to associate matching video-text pairs while distinguishing non-matching pairs. This architectural design supports applications including video retrieval, video-text matching, and coarse-grained video captioning, where the primary task involves measuring similarity or correspondence between visual and linguistic content rather than generating novel text.

Current STAA methodology explains visual attention patterns within the video encoder, generating spatio-temporal attribution maps that reveal which spatial regions and temporal segments influence the model's processing of video content. However, extending this capability to video-language models requires three additional analytical dimensions. First, the method must continue to extract and aggregate attention weights from the video encoder to maintain the existing capability of generating spatio-temporal attribution maps. Second, the extension must capture attention weights from cross-modal layers to visualize how specific video regions correspond to particular text tokens, thereby quantifying the strength of video-text alignment at a granular level. Third, the method must track how

text queries influence visual attention patterns, identifying which spatial-temporal regions become salient in response to specific words or phrases and generating text-conditioned visual explanations that reveal query-dependent processing.

Technical Challenges and Solution Strategies Extending STAA to video-language models presents several interrelated technical challenges that must be addressed through coordinated solution strategies. The challenge of multi-modal attention fusion arises because current STAA focuses exclusively on visual attention within video encoders, whereas video-language models distribute information processing across separate visual and textual attention streams as well as cross-modal layers that integrate information between modalities. The proposed solution involves extending the attention extraction mechanism to capture cross-modal attention matrices and implementing separate aggregation pathways for visual attention and cross-modal attention, allowing the method to track information flow both within and between modalities.

6.3.3 Scalable Deployment and Real-time Adaptation

The transition from research prototypes to production-ready XAI systems presents significant engineering and algorithmic challenges. Future work must address the computational bottlenecks that currently limit XAI deployment at scale, particularly for real-time applications. This involves developing approximate explanation methods that provide guaranteed accuracy bounds while operating within strict latency constraints. Research should explore hardware-accelerated XAI implementations, potentially leveraging specialized neural processing units designed for explanation generation alongside prediction tasks.

Moreover, production environments demand XAI systems that can adapt to distribution shifts and evolving model architectures without manual reconfiguration. Future research

should investigate meta-learning approaches that enable explanation methods to automatically adjust their parameters based on observed model behavior and data characteristics. This includes developing online learning algorithms for explanation generation that can incrementally update their understanding of model behavior as new data becomes available, while maintaining consistency with previously generated explanations. The integration of such adaptive mechanisms with existing MLOps infrastructure requires careful consideration of versioning, monitoring, and rollback strategies specific to explanation pipelines.

6.3.4 Regulatory Alignment and Standardization

As AI regulation evolves globally, future work must ensure that XAI frameworks align with emerging legal requirements while maintaining technical rigor. This involves developing formal verification methods that can certify explanation properties required by regulations such as the EU AI Act, including fairness, consistency, and comprehensiveness. Research should establish standardized metrics and benchmarks for evaluating explanation quality across different regulatory jurisdictions, enabling organizations to demonstrate compliance through quantifiable measures rather than subjective assessments.

The development of industry-standard XAI protocols requires collaboration between technical researchers, legal experts, and domain practitioners. Future work should focus on creating domain-specific explanation templates that satisfy both regulatory requirements and practical utility for end-users. This includes investigating how explanations can be automatically tailored to different stakeholder groups while maintaining semantic consistency, and developing audit trails that document the entire explanation generation process for regulatory review. The challenge extends to creating explanation methods that remain robust under adversarial scrutiny, as regulatory compliance may involve defending explanations against deliberate attempts to find inconsistencies or biases.

6.4 Broader Impact and Concluding Remarks

The practical implications of this research extend beyond technical contributions to influence how organizations develop, deploy, and govern AI systems. By providing a unified framework that bridges cloud services and open-source models, this work enables smaller organizations and research groups to implement enterprise-grade XAI capabilities without substantial infrastructure investments. The microservice architecture facilitates gradual adoption, allowing teams to incrementally integrate explainability features rather than requiring wholesale system redesigns.

From an industry perspective, the framework's cloud-agnostic design and standardized APIs reduce vendor lock-in concerns while enabling comparative analysis across competing AI services. This capability is particularly valuable for organizations evaluating multiple AI providers or transitioning between platforms. The reproducibility features embedded in our pipeline architecture also address growing demands for AI audit trails in regulated industries, providing documented evidence of model behavior and explanation consistency over time.

The societal impact of accessible, robust XAI extends to enhancing public trust in AI systems. By making explanation capabilities more readily deployable, this research contributes to democratizing AI transparency, enabling broader scrutiny of automated decision-making systems that increasingly influence critical aspects of daily life. The adversarial robustness components of our work particularly address concerns about AI system manipulation, providing tools to identify and mitigate potential security vulnerabilities before they can be exploited maliciously.

In conclusion, this thesis has presented a comprehensive framework that advances the state of practice in explainable AI through systematic integration of interpretation, robustness analysis, and reproducible deployment mechanisms. While significant challenges remain in achieving truly interpretable and trustworthy AI, the contributions presented here

provide essential building blocks for continued progress. The convergence of technical capability, regulatory pressure, and societal demand for AI transparency suggests that explainable AI will transition from a research specialty to a fundamental requirement for AI system deployment. Through continued development of frameworks like ours, the AI community can work toward a future where powerful AI capabilities coexist with human understanding, accountability, and control.

Appendix A

Additional XAI Case Study for Tabular and NLP Model

A.1 Case Study for Tablular model: Academic Paper Ranker

The first case study explains the academic papers ranking model, a black-box model with tabular inputs. We select four XAI methods: ALE, Shapley Value, Anchor, and SHAP to derive the overall feature importance order. We evaluate those methods by explanation stability and explanation consistency.

Model and Data

The target model is from Semantic Scholar [60], a scientific literature search engine that uses the open-source machine learning ranking model s2search [58]. *s2search* model ranks papers according to the collected user behaviour data, such as search logs and user clicks. In this work, our objective is to determine the order of the importance of the six features : *title*, *abstract*, *venue*, *authors*, *year*, *n_citations*.

For the data set, we choose the arXiv data set [37] meta topics as classification labels

under the field of *Computer Science*. We use the subset with secondary categories of *Computer Science* provided by arXiv. Each paper is categorized with one or more meta topics. Finally, we have thirty-eight data sets categorized by arXiv with 542,877 papers' metadata for the case study experiment.

Explanation Consistency Analysis

According to the taxonomy in Figure 1, we select ALE, Shapley Value, Anchor, and SHAP to derive feature importance order for the black-box model. We develop the software solution for the four XAI methods. The source code is available from this GitHub repository[1].

Summary from Feature Importance Order For all methods that we selected, we define the feature contribution value for feature j over data set X as:

$$\phi_j(\delta_j^X) = \frac{1}{n}\sum_{i=1}^{n} |\delta_j^{x^{[i]}}|$$

where the $\delta_j^{x^{[i]}}$ is the contribution value for feature j on data sample $x^{[i]}$. For the ALE explanation, we set each local effect from it for feature j as the feature contribution value $\delta_j^{x^{[i]}}$. For the Shapley Value or SHAP explanation on data sample $x^{[i]}$, we set each contribution value from it for feature j as $\delta_j^{x^{[i]}}$. For the Anchor explanation on each data sample $x^{[i]}$, we set the partial precision value for feature j as the feature contribution value $\delta_j^{x^{[i]}}$. Repeating this process to every feature and get feature contribution values and further derive the feature importance order over the data set X. The results are listed in Table 21.

Table 21: Summary of feature importance order across XAI methods (feature name with importance score)

Method	**Feature Importance Order**					
ALE	abstract (5.4923)	title (2.6864)	venue (2.5473)	year (0.506)	n_citations (0.266)	authors (0.2504)
Shapley Value	abstract (5.4701)	title (1.6359)	year (1.0298)	venue (0.8989)	n_citations (0.171)	authors (0.0987)
Anchor	abstract (0.4659)	year (0.1724)	title (0.1592)	venue (0.0853)	n_citations (0.0461)	authors (0.0074)
SHAP	abstract (4.5293)	title (1.2729)	year (0.8781)	venue (0.7461)	n_citations (0.1473)	authors (0.0899)

[1] `https://github.com/youyinnn/s2search`

Stability Observation

To observe the explanation stability, we run each XAI method on thirty-eight data sets. This experiment produces thirty-eight feature importance orders as the explanation summaries. We apply *Kendall tau Distance* algorithm as the distance function to compute $f_d^{[k]}(\xi^i, \xi^j), (i \neq j, i \leq 38, j \leq 38)$. We compute 703 distances for each XAI method and produce the violin plot with mean value bar. The result indicates that the ALE method has the lowest mean value on the stability metric.

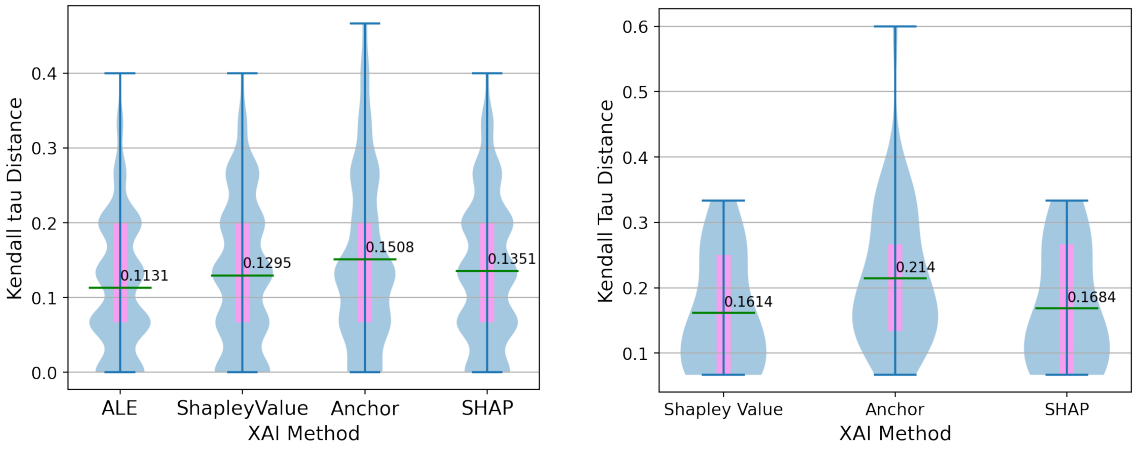


Figure 50: Explanation stability and consistency. Left: Stability of each method; higher mean values indicate better robustness across runs. Right: Consistency of explanations compared to ALE. Green lines denote mean values across samples.

Consistency Observation

We select ALE as the baseline more for inter-XAI method consistency observation since ALE performs better in term of method stability across multiple data sets on the same prediction model. To calculate the consistency between the ALE method and other methods, we use the Equation 6 with the *Kendall tau Distance* algorithm as the distance function $f_d^{[k]}(\xi^{ALE}, \xi^j), (I \neq j, j \leq 4, k = 1, 2, 3)$. Figure 50 reads as the explanation difference between Shapley Value and ALE, Anchor and ALE, and SHAP and ALE in term of distances measured on thirty-eight explanation summaries. Shapley Value and SHAP has

approximate mean distance values compared to ALE. The plot indicates Shapley Value and SHAP produce more consistent explanation with ALE than Anchor does.

A.2 Case Study for NLP Model: Code Vulnerability Classifier

Popular code vulnerability data sets such as the Open Web Application Security Project (OWASP) [152] Benchmark and Juliet test suite [23] provide a reliable training corpus. These data sets contain mined method-level software code files with the Common Weakness Enumeration (CWE) labels. The code files usually contain the code comments, code body, and import packages. A class of technique is that machine learning models treat the code file as the text content and classify it into different CWE labels. It is a natural language processing classification problem. In this case study, we observe whether machine learning models capture semantics from code bodies, code comments or import packages and make the classification decision.

Model and Data

The data sets are from The Open Web Application Security Project (OWASP) Benchmark [152] and Juliet test suite [23] for Java. OWASP Benchmark contains 2,740 test cases with 52% files of vulnerable code and 48% non-vulnerable files. The 52% vulnerable files contain 11 CWE (Common Weakness Enumeration) labels. The Juliet test suite contains 217 vulnerable and 297 non-vulnerable files with 112 different CWE labels. The CWE labels are the ground truth for multiclass classification.

We practice this case study with the state-of-the-art Natural Language Processing (NLP) model XLNet [216]. It is a model that could capture bi-directional text information and outperform other state-of-the-art NLP models. Each text content in a code file containing

a method with a CWE label is considered the data instance and the label pair. To observe which part of these contexts highly impacts the machine learning classifier, we identify three features in the data: comments, code body and import packages. Comments contain the file description and the code comments. Further, the description comment of the Juliet test case includes the CWE types directly. We remove this content to avoid data leakage during the model's training. Data leakage is defined as unintentionally leaking the signal. In our case, the CWE type content to the model potentially increases the accuracy [154]. We directly remove one of the three features and remain the other two features to get the prediction from the model as masked the feature.

The label distributions in both data sets are also unbalanced, and the non-vulnerable code accounts for over 50%. In practice, we combine labels whose numbers count less than ten percent of the total size and generate four labels for both data sets.

The Juliet test case for Java and the OWASP Benchmark test suite are treated as two data sets to perform the experiments. We shuffle each data set into a training set and testing set by the percentage of eighty and twenty. The training set is used for fine-tuning the XLNet model. The testing set is used for the masking feature and collects the prediction for XAI processing. The features are removed before input to XLNet model [216] to output the log-odd probability of the ground truth CWE label.

Explanation Consistency Analysis

Under the objective of the case study, observing which feature among the three features affects the most model deciding, we select methods that could reflect the feature influence. Meanwhile, features in text content can be evaluated by masking. We pick up Shapley Value [179], SHAP [126], Preddiff [231] and Mean-Centroid Preddiff [115] to perform the feature masking-based explanation.

Stability Observation

In this case study, we have two data sets level feature importance order in each XAI method for OWASP Benchmark and Juliet, respectively. These eight results are directly demonstrated in Table 22. The run-time for each XAI method also is compared in this subsection. Shapley Value and SHAP derive feature contribution value for each instance first and then are aggregated into data set level feature contribution value $\phi_j(\delta_j(X))$. The feature importance order are sorted by $order = sort(abs(\phi_j(\delta_j(X))))$. The Preddiff and Mean-Centroid Preddiff could directly aggregate the data set level feature importance order by $order = sort(abs(\phi_j(\delta_j(X))))$, where δ is the prediction difference in paper [115].

Consistency Observation

We summarize the data set level feature importance order in Table 22. The feature importance order results of four XAI methods are consistent for Juliet test cases. *comment* has the most importance than *code* and *import*. Each method has a zero *Kendall tau distance* with other methods.

For the OWASP Benchmark data set, Preddiff and Mean-Centriod Preddiff have different insights on features. Measuring by *Kendall tau distance*, we calculate the average feature importance order distance for Preddiff and Mean-Centroid Preddiff are 0.33 from the other two methods. We observe Shapley Value and SHAP achieve a higher consistency.

Table 22: Feature Importance Order Summary of Code Vulnerability Detection Case Study

XAI Methods	**Juliet test cases for Java**	**OWASP Benchmark**
Preddiff	`comment > code > import`	`code > import > comment`
Mean-Centroid Preddiff	`comment > code > import`	`code > import > comment`
Shapley value	`comment > code > import`	`comment > code > import`
KernelSHAP	`comment > code > import`	`comment > code > import`

Bibliography

[1] K. Abhishek and D. Kamath. Attribution-based xai methods in computer vision: A review. *arXiv preprint arXiv:2211.14736*, 2022.

[2] A. Adadi and M. Berrada. Peeking inside the black-box: a survey on explainable artificial intelligence (xai). *IEEE access*, 6:52138–52160, 2018.

[3] N. Akhtar and A. Mian. Threat of adversarial attacks on deep learning in computer vision: A survey. *IEEE Access*, 6:14410–14430, 2018.

[4] L. Akritidis. Product Classification and Clustering. UCI Machine Learning Repository, 2023. DOI: https://doi.org/10.24432/C5M91Z.

[5] Alibaba. Alibaba cloud, 2023. Accessed: 2023.

[6] J. Amann, A. Blasimme, E. Vayena, D. Frey, V. I. Madai, and P. Consortium. Explainability for artificial intelligence in healthcare: a multidisciplinary perspective. *BMC medical informatics and decision making*, 20:1–9, 2020.

[7] Amazon. Amazon web services, 2023. Accessed: 2023.

[8] S. Amershi, A. Begel, C. Bird, R. DeLine, H. Gall, E. Kamar, N. Nagappan, B. Nushi, and T. Zimmermann. Software engineering for machine learning: A case study. In *2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)*, pages 291–300, 2019.

[9] D. W. Apley and J. Zhu. Visualizing the effects of predictor variables in black box supervised learning models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 82(4):1059–1086, 2020.

[10] S. Ö. Arik and T. Pfister. Tabnet: Attentive interpretable tabular learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 6679–6687, 2021.

[11] A. Arnab, M. Dehghani, G. Heigold, C. Sun, M. Lučić, and C. Schmid. Vivit: A video vision transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6836–6846, 2021.

[12] L. Arras, G. Montavon, K.-R. Müller, and W. Samek. Explaining recurrent neural network predictions in sentiment analysis. 2017.

[13] A. B. Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. García, S. Gil-López, D. Molina, R. Benjamins, et al. Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information fusion*, 58:82–115, 2020.

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

[15] H. Baniecki, W. Kretowicz, P. Piatyszek, J. Wisniewski, and P. Biecek. dalex: Responsible machine learning with interactive explainability and fairness in python. *The Journal of Machine Learning Research*, 22(1):9759–9765, 2021.

[16] D. Baylor, E. Breck, H.-T. Cheng, N. Fiedel, C. Y. Foo, Z. Haque, S. Haykal, M. Ispir, V. Jain, L. Koc, et al. Tfx: A tensorflow-based production-scale machine learning platform. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1387–1395, 2017.

[17] K. Belhajjame, R. B'Far, J. Cheney, S. Coppens, S. Cresswell, Y. Gil, P. Groth, G. Klyne, T. Lebo, J. McCusker, et al. Prov-dm: The prov data model. *W3C Recommendation*, 14:15–16, 2013.

[18] Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828, 2013.

[19] G. Bertasius, H. Wang, and L. Torresani. Is space-time attention all you need for video understanding? In *Proceedings of the International Conference on Machine Learning*, pages 813–824, 2021.

[20] U. Bhatt, A. Xiang, S. Sharma, A. Weller, A. Taly, Y. Jia, J. Ghosh, R. Puri, J. M. Moura, and P. Eckersley. Explainable machine learning in deployment. *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pages 648–657, 2020.

[21] L. Biewald. Experiment tracking with weights and biases, 2020. Software available from wandb.com.

[22] A. Binder, S. Bach, G. Montavon, K.-R. Müller, and W. Samek. Layer-wise relevance propagation for deep neural network architectures. In *Information science and applications (ICISA) 2016*, pages 913–922. Springer, 2016.

[23] P. E. Black and P. E. Black. *Juliet 1.3 test suite: Changes from 1.2*. US Department of Commerce, National Institute of Standards and Technology, 2018.

[24] M. Branco and L. Moreau. Enabling provenance on large scale e-science applications. In *Provenance and Annotation of Data: International Provenance and Annotation Workshop, IPAW 2006, Chicago, IL, USA, May 3-5, 2006, Revised Selected Papers*, pages 55–63. Springer, 2006.

[25] A. Brinckman, K. Chard, N. Gaffney, M. Hategan, M. B. Jones, K. Kowalik, S. Kulasekaran, B. Ludäscher, B. D. Mecum, J. Nabrzyski, et al. Computing environments for reproducibility: Capturing the "whole tale". *Future Generation Computer Systems*, 94:854–867, 2019.

[26] L. Budach, M. Feuerpfeil, N. Ihde, A. Nathansen, N. Noack, H. Patzlaff, F. Naumann, and H. Harmouch. The effects of data quality on machine learning performance. *arXiv preprint arXiv:2207.14529*, 2022.

[27] A. Bussone, S. Stumpf, and D. O'Sullivan. The role of explanations on trust and reliance in clinical decision support systems. In *2015 international conference on healthcare informatics*, pages 160–169. IEEE, 2015.

[28] N. Carlini and D. Wagner. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pages 39–57. Ieee, 2017.

[29] J. Carreira and A. Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017.

[30] A. Chattopadhay, A. Sarkar, P. Howlader, and V. N. Balasubramanian. Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In *2018 IEEE winter conference on applications of computer vision (WACV)*, pages 839–847. IEEE, 2018.

[31] H. Chefer, S. Gur, and L. Wolf. Transformer interpretability beyond attention visualization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 782–791, 2021.

[32] K. Chen, Z. Wei, J. Chen, Z. Wu, and Y.-G. Jiang. Attacking video recognition models with bullet-screen comments. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 312–320, 2022.

[33] P.-Y. Chen, H. Zhang, Y. Sharma, J. Yi, and C.-J. Hsieh. Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *Proceedings of the 10th ACM workshop on artificial intelligence and security*, pages 15–26, 2017.

[34] Z. Chen, L. Xie, S. Pang, Y. He, and Q. Tian. Appending adversarial frames for universal video attack. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 3199–3208, 2021.

[35] Chinu and U. Bansal. Explainable ai: To reveal the logic of black-box models. *New Generation Computing*, pages 1–35, 2023.

[36] M. Cisse, Y. Adi, N. Neverova, and J. Keshet. Houdini: Fooling deep structured prediction models. *arXiv preprint arXiv:1707.05373*, 2017.

[37] C. B. Clement, M. Bierbaum, K. P. O'Keeffe, and A. A. Alemi. On the use of arxiv as a dataset, 2019.

[38] P. Cremonesi, Y. Koren, and R. Turrin. Performance of recommender algorithms on top-n recommendation tasks. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 39–46, 2010.

[39] A. Cummaudo, S. Barnett, R. Vasa, and J. Grundy. Threshy: Supporting safe usage of intelligent web services. In *Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, pages 1645–1649, 2020.

[40] A. Cummaudo, R. Vasa, S. Barnett, J. Grundy, and M. Abdelrazek. Interpreting cloud computer vision pain-points: a mining study of stack overflow. In *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering*, page 1584–1596, 2020.

[41] A. Cummaudo, R. Vasa, J. Grundy, M. Abdelrazek, and A. Cain. Losing confidence in quality: Unspoken evolution of computer vision services. In *2019 IEEE International Conference on Software Maintenance and Evolution (ICSME)*, pages 333–342. IEEE, 2019.

[42] J. R. Dabrowski and E. V. Munson. Is 100 milliseconds too fast? In *CHI '01 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '01, page 317–318, New York, NY, USA, 2001. Association for Computing Machinery.

[43] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[44] B. J. Dietvorst, J. P. Simmons, and C. Massey. Algorithm aversion: People erroneously avoid algorithms after seeing them err. *Journal of Experimental Psychology: General*, 144(1):114, 2015.

[45] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2625–2634, 2015.

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

[47] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[48] R. L. Draelos and L. Carin. Hirescam: Faithful location representation in visual attention for explainable 3d medical image classification. *arXiv preprint arXiv:2011.08891*, 2020.

[49] J. H. Drew, A. G. Glen, and L. M. Leemis. Computing the cumulative distribution function of the kolmogorov–smirnov statistic. *Computational statistics & data analysis*, 34(1):1–15, 2000.

[50] C. Ebert, G. Gallardo, J. Hernantes, and N. Serrano. Devops. *IEEE Software*, 33(3):94–100, 2016.

[51] U. Ehsan, Q. V. Liao, M. Muller, M. O. Riedl, and J. D. Weisz. Expanding explainability: Towards social transparency in ai systems. *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–19, 2021.

[52] C. Etmann, S. Lunz, P. Maass, and C.-B. Schönlieb. On the connection between adversarial robustness and saliency map interpretability. *arXiv preprint arXiv:1905.04172*, 2019.

[53] European Commission. Proposal for a regulation laying down harmonised rules on artificial intelligence, 2021.

[54] S. Fabbrizzi, S. Papadopoulos, E. Ntoutsi, and I. Kompatsiaris. A survey on bias in visual datasets. *Computer Vision and Image Understanding*, 223:103552, 2022.

[55] M. Felderer and R. Ramler. Quality assurance for ai-based systems: overview and challenges. *arXiv preprint arXiv:2102.05351*, 2021.

[56] S. A. Fezza, Y. Bakhti, W. Hamidouche, and O. Déforges. Perceptual evaluation of adversarial attacks for cnn-based image classification. In *2019 eleventh international conference on quality of multimedia experience (QoMEX)*, pages 1–6. IEEE, 2019.

[57] A. Fisher, C. Rudin, and F. Dominici. All models are wrong, but many are useful: Learning a variable's importance by studying an entire class of prediction models simultaneously. 2018.

[58] A. I. for AI. Github - allenai/s2search: The semantic scholar search reranker., 2020.

[59] A. Fox, R. Griffith, A. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, et al. Above the clouds: A berkeley view of cloud computing. *Dept. Electrical Eng. and Comput. Sciences, University of California, Berkeley, Rep. UCB/EECS*, 28(13):2009, 2009.

[60] S. Fricke. Semantic scholar. *Journal of the Medical Library Association: JMLA*, 106(1):145, 2018.

[61] J. H. Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.

[62] R. Fu, Q. Hu, X. Dong, Y. Guo, Y. Gao, and B. Li. Axiom-based grad-cam: Towards accurate visualization and explanation of cnns. *arXiv preprint arXiv:2008.02312*, 2020.

[63] S. Garg, K. Kaur, G. S. Aujla, G. Kaddoum, P. Garigipati, and M. Guizani. Trusted explainable ai for 6g-enabled edge cloud ecosystem. *IEEE Wireless Communications*, 30(3):163–170, 2023.

[64] A. Ghorbani, A. Abid, and J. Zou. Interpretation of neural networks is fragile. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3681–3688, 2019.

[65] P. Gohel, P. Singh, and M. Mohanty. Explainable ai: current status and future directions. *ArXiv*, abs/2107.07045, 2021.

[66] A. Goldstein, A. Kapelner, J. Bleich, and E. Pitkin. Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation. *journal of Computational and Graphical Statistics*, 24(1):44–65, 2015.

[67] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[68] B. Goodman and S. Flaxman. European union regulations on algorithmic decision-making and a "right to explanation". *AI Magazine*, 38(3):50–57, 2017.

[69] Google. Google cloud platform, 2023. Accessed: 2023.

[70] Y. Gorishniy, I. Rubachev, V. Khrulkov, and A. Babenko. Revisiting deep learning models for tabular data. *Advances in Neural Information Processing Systems*, 34:18932–18943, 2021.

[71] Government of Canada. Directive on automated decision-making, 2021.

[72] R. Goyal, S. Ebrahimi Kahou, V. Michalski, J. Materzynska, S. Westphal, H. Kim, V. Haenel, I. Fruend, P. Yianilos, M. Mueller-Freitag, et al. The" something something" video database for learning and evaluating visual common sense. In *Proceedings of the IEEE international conference on computer vision*, pages 5842–5850, 2017.

[73] Y. Goyal, A. Feder, U. Shalit, and B. Kim. Explaining classifiers with causal concept effect (cace). *arXiv preprint arXiv:1907.07165*, 2019.

[74] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi. A survey of methods for explaining black box models. 51(5), 2018.

[75] D. Gunning and D. Aha. Darpa's explainable artificial intelligence (xai) program. *AI magazine*, 40(2):44–58, 2019.

[76] T. Guo, T. Lin, and N. Antulov-Fantulin. Exploring interpretable lstm neural networks over multi-variable data. In *International conference on machine learning*, pages 2494–2504. PMLR, 2019.

[77] K. M. Habibullah, G. Gay, and J. Horkoff. Non-functional requirements for machine learning: An exploration of system scope and interest. In *Proceedings of the 1st Workshop on Software Engineering for Responsible AI*, pages 29–36, 2022.

[78] K. Hara, H. Kataoka, and Y. Satoh. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 6546–6555, 2018.

[79] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[80] D. Hendrycks and T. Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*, 2019.

[81] D. Hendrycks and T. Dietterich. Benchmarking neural network robustness to common corruptions and perturbations, 2019.

[82] L. Hiley, A. Preece, Y. Hicks, D. Marshall, and H. Taylor. Discriminating spatial and temporal relevance in deep taylor decompositions for explainable activity recognition. *arXiv preprint arXiv:1908.01536*, 2019.

[83] A. Hopkins and S. Booth. Machine learning practices outside big tech: How resource constraints challenge responsible development. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*, pages 134–145, 2021.

[84] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[85] J. Huang, Z. Wang, D. Li, and Y. Liu. The analysis and development of an xai process on feature contribution explanation. In *2022 IEEE International Conference on Big Data (Big Data)*, pages 5039–5048, 2022.

[86] X. Huang, A. Khetan, M. Cvitkovic, and Z. Karnin. Tabtransformer: Tabular data modeling using contextual embeddings. *arXiv preprint arXiv:2012.06678*, 2020.

[87] J. Hwang, J.-H. Kim, J.-H. Choi, and J.-S. Lee. Just one moment: Structural vulnerability of deep action recognition against one frame attack. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 7668–7676, 2021.

[88] A. Ilyas, S. Santurkar, D. Tsipras, L. Engstrom, B. Tran, and A. Madry. Adversarial examples are not bugs, they are features. *Advances in neural information processing systems*, 32, 2019.

[89] K. A. S. Immink and J. H. Weber. Minimum pearson distance detection for multilevel channels with gain and/or offset mismatch. *IEEE Transactions on Information Theory*, 60(10):5966–5974, 2014.

[90] X. Jia, X. Wei, and X. Cao. Identifying and resisting adversarial videos using temporal consistency. *arXiv preprint arXiv:1909.04837*, 2019.

[91] L. Jiang, X. Ma, S. Chen, J. Bailey, and Y.-G. Jiang. Black-box adversarial attacks on video recognition models, 2019.

[92] P.-T. Jiang, C.-B. Zhang, Q. Hou, M.-M. Cheng, and Y. Wei. Layercam: Exploring hierarchical class activation maps for localization. *IEEE Transactions on Image Processing*, 30:5875–5888, 2021.

[93] E. Jonas, J. Schleier-Smith, V. Sreekanti, C.-C. Tsai, A. Khandelwal, Q. Pu, V. Shankar, J. Carreira, K. Krauth, N. Yadwadkar, et al. Cloud programming simplified: A berkeley view on serverless computing. *arXiv preprint arXiv:1902.03383*, 2019.

[94] D. Justus, J. Brennan, S. Bonner, and A. S. McGough. Predicting the computational cost of deep learning models. In *2018 IEEE international conference on big data (Big Data)*, pages 3873–3882. IEEE, 2018.

[95] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

[96] M. G. Kendall. A new measure of rank correlation. *Biometrika*, 30(1/2):81–93, 1938.

[97] J.-H. Kim, W. Choo, and H. O. Song. Puzzle mix: Exploiting saliency and local statistics for optimal mixup. In *International Conference on Machine Learning*, pages 5275–5285. PMLR, 2020.

[98] D. N. Kluttz, N. Kohli, and D. K. Mulligan. Shaping our tools: Contestability as a means to promote responsible algorithmic decision making in the professions. *Ethics, Law and Governance of AI*, 2020.

[99] N. Kokhlikyan, V. Miglani, M. Martin, E. Wang, B. Alsallakh, J. Reynolds, A. Melnikov, N. Kliushkina, C. Araya, S. Yan, and O. Reblitz-Richardson. Captum: A unified and generic model interpretability library for pytorch, 2020.

[100] D. Kreuzberger, N. Kühl, and S. Hirschl. Machine learning operations (mlops): Overview, definition, and architecture. *IEEE Access*, 11:31866–31879, 2023.

[101] L. Krieger, R. Nijzink, G. Thakur, C. Ramakrishnan, R. Roskar, and S. Schymanski. Repeatable and reproducible workflows using the renku open science platform. In *EGU General Assembly Conference Abstracts*, pages EGU21–7655, 2021.

[102] S. Krishna, L. Han, A. Tenenbaum, T. L. Wolberg, E. Horel, A. Galstyan, G. Ver Steeg, and A. Kobren. Disagreement in explanation-based adversarial transferability. In *International Conference on Learning Representations*, 2022.

[103] T. Kulesza, S. Stumpf, M. Burnett, S. Yang, I. Kwan, and W.-K. Wong. Too much, too little, or just right? ways explanations impact end users' mental models. In *2013 IEEE Symposium on visual languages and human centric computing*, pages 3–10. IEEE, 2013.

[104] A. Kurakin, I. J. Goodfellow, and S. Bengio. Adversarial examples in the physical world. In *Artificial intelligence safety and security*, pages 99–112. Chapman and Hall/CRC, 2018.

[105] Z. Kurtanović and W. Maalej. Automatically classifying functional and non-functional requirements using supervised machine learning. In *2017 IEEE 25th International Requirements Engineering Conference (RE)*, pages 490–495. Ieee, 2017.

[106] A. Lacoste, A. Luccioni, V. Schmidt, and T. Dandres. Quantifying the carbon emissions of machine learning. *arXiv preprint arXiv:1910.09700*, 2019.

[107] H. Lakkaraju, S. H. Bach, and J. Leskovec. Interpretable decision sets: A joint framework for description and prediction. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1675–1684, 2016.

[108] H. Lakkaraju and O. Bastani. How do i fool you? manipulating user trust via misleading black box explanations. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, pages 79–85, 2020.

[109] P. Lambin, R. T. H. Leijenaar, T. M. Deist, J. Peerlings, E. E. de Jong, J. E. van Timmeren, S. Sanduleanu, R. T. H. M. Larue, A. J. G. Even, A. Jochems, Y. van Wijk, H. Woodruff, J. van Soest, T. Lustberg, E. Roelofs, W. van Elmpt, A. Dekker, F. M. Mottaghy, J. E. Wildberger, and S. Walsh. Radiomics: the bridge between medical imaging and personalized medicine. *Nature Reviews Clinical Oncology*, 14:749–762, 2017.

[110] N. Larsen, J. Stallrich, S. Sengupta, A. Deng, R. Kohavi, and N. T. Stevens. Statistical challenges in online controlled experiments: A review of a/b testing methodology. *The American Statistician*, (just-accepted):1–32, 2023.

[111] J. Larson, S. Mattu, L. Kirchner, and J. Angwin. How we analyzed the compas recidivism algorithm. *ProPublica (5 2016)*, 9(1):3–3, 2016.

[112] G. Lee and S. Lee. Toward reliable software analytics: Systematic integration of explanations from different model-agnostic techniques. *IEEE Software*, 40(03):34–42, may 2023.

[113] K. Lee and N. Ha. Ai platform to accelerate api economy and ecosystem. In *2018 International Conference on Information Networking (ICOIN)*, pages 848–852, 2018.

[114] B. Letham, C. Rudin, T. H. McCormick, and D. Madigan. Interpretable classifiers using rules and bayesian analysis: Building a better stroke prediction model. *The Annals of Applied Statistics*, 9(3):1350–1371, 2015.

[115] D. Li, Y. Liu, J. Huang, and Z. Wang. A trustworthy view on explainable artificial intelligence method evaluation. *Computer*, 56(4):50–60, 2023.

[116] J. Li, K. Gao, Y. Bai, J. Zhang, S.-t. Xia, and Y. Wang. Fmm-attack: A flow-based multi-modal adversarial attack on video-based llms. *arXiv preprint arXiv:2403.13507*, 2024.

[117] X. Li, M. Du, J. Chen, Y. Chai, H. Lakkaraju, and H. Xiong. A unified xai benchmark for faithfulness evaluation of feature attribution methods across metrics, modalities and models. *Advances in Neural Information Processing Systems*, 36:1630–1643, 2023.

[118] X.-H. Li, Y. Shi, H. Li, W. Bai, Y. Song, C. C. Cao, and L. Chen. Quantitative evaluations on saliency methods: An experimental study. *arXiv preprint arXiv:2012.15616*, 2020.

[119] Y. Liu, M. Boels, L. C. García-Peraza-Herrera, T. K. M. Vercauteren, P. Dasgupta, A. Granados, and S. Ourselin. Lovit: Long video transformer for surgical phase recognition. *ArXiv*, abs/2305.08989, 2023.

[120] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021.

[121] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11976–11986, 2022.

[122] Z. Liu, J. Ning, Y. Cao, Y. Wei, Z. Zhang, S. Lin, and H. Hu. Video swin transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3202–3211, 2022.

[123] L. Longo, M. Brcic, F. Cabitza, J. Choi, R. Confalonieri, J. Del Ser, R. Guidotti, Y. Hayashi, F. Herrera, A. Holzinger, et al. Explainable artificial intelligence (xai) 2.0: A manifesto of open challenges and interdisciplinary research directions. *Information Fusion*, page 102301, 2024.

[124] P. Lopes, E. Silva, C. Braga, T. Oliveira, and L. Rosado. Xai systems evaluation: A review of human and computer-centred methods. *Applied Sciences*, 12(19):9423, 2022.

[125] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2017.

[126] S. M. Lundberg and S.-I. Lee. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017.

[127] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

[128] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards deep learning models resistant to adversarial attacks, 2019.

[129] J. N. Mandrekar. Receiver operating characteristic curve in diagnostic test assessment. *Journal of Thoracic Oncology*, 5(9):1315–1316, 2010.

[130] C. D. Manning. *An introduction to information retrieval*. Cambridge university press, 2009.

[131] C. Meske and E. Bunde. Transparency and trust in human-ai-interaction: The role of model-agnostic explanations in computer vision-based decision support. In *International Conference on Human-Computer Interaction*, pages 54–69. Springer, 2020.

[132] Microsoft. Azure cloud services, 2023. Accessed: 2023.

[133] T. Miller. Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence*, 267:1–38, 2019.

[134] T. Miller. Explanation in artificial intelligence: Insights from the social sciences, 2019.

[135] S. Mohseni, N. Zarei, and E. D. Ragan. A multidisciplinary survey and framework for design and evaluation of explainable ai systems, 2021.

[136] C. Molnar. *Interpretable machine learning*. Lulu. com, 2020.

[137] S.-M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard. Universal adversarial perturbations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1765–1773, 2017.

[138] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard. Deepfool: A simple and accurate method to fool deep neural networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2574–2582, 2016.

[139] P. Moro S., Rita and C. P. Bank Marketing. UCI Machine Learning Repository, 2014. DOI: https://doi.org/10.24432/C5K306.

[140] M. B. Muhammad and M. Yeasin. Eigen-cam: Class activation map using principal components. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7. IEEE, 2020.

[141] K. Naik and P. Tripathy. *Software testing and quality assurance: theory and practice*. John Wiley & Sons, 2011.

[142] National Institute of Standards and Technology. Artificial intelligence risk management framework (ai rmf 1.0). NIST Special Publication 1270, U.S. Department of Commerce, 2023.

[143] E. Neghawi, Z. Wang, J. Huang, and Y. Liu. Linking team-level and organization-level governance in machine learning operations through explainable ai and responsible ai connector. In *2023 IEEE 47th Annual Computers, Software, and Applications Conference (COMPSAC)*, pages 1223–1230. IEEE, 2023.

[144] H. Nori, S. Jenkins, P. Koch, and R. Caruana. Interpretml: A unified framework for machine learning interpretability. *arXiv preprint arXiv:1909.09223*, 2019.

[145] M. Nourani, C. Roy, T. Rahman, E. D. Ragan, N. Ruozzi, and V. Gogate. Don't explain without verifying veracity: An evaluation of explainable ai with video activity recognition. *ArXiv*, abs/2005.02335, 2020.

[146] J. J. Ohana, S. Ohana, E. Benhamou, D. Saltiel, and B. Guez. Explainable ai (xai) models applied to the multi-agent environment of financial markets. In *Explainable and Transparent AI and Multi-Agent Systems: Third International Workshop, EXTRAAMAS 2021, Virtual Event, May 3–7, 2021, Revised Selected Papers 3*, pages 189–207. Springer, 2021.

[147] S. Palacio, A. Lucieri, M. Munir, S. Ahmed, J. Hees, and A. Dengel. Xai handbook: towards a unified framework for explainable ai. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3766–3775, 2021.

[148] Y. Pan, J.-J. Huang, Z. Chen, W. Zhao, and Z. Wang. SVASTIN: Sparse Video Adversarial Attack via Spatio-Temporal Invertible Neural Networks . In *2024 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6, 2024.

[149] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami. The limitations of deep learning in adversarial settings. In *2016 IEEE European symposium on security and privacy (EuroS&P)*, pages 372–387. IEEE, 2016.

[150] U. Pawar, D. O'Shea, S. Rea, and R. O'Reilly. Incorporating explainable artificial intelligence (xai) to aid the understanding of machine learning in the healthcare domain. In *AICS*, pages 169–180, 2020.

[151] V. Petsiuk, A. Das, and K. Saenko. Rise: Randomized input sampling for explanation of black-box models. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2018.

[152] P. R. Phil. Owasp top 10: The top 10 most critical web application security threats enhanced with text analytics and content by pagekicker robot phil 73. 2014.

[153] S. Polley, R. R. Koparde, A. B. Gowri, M. Perera, and A. Nuernberger. Towards trustworthiness in the context of explainable search. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2580–2584, 2021.

[154] M. T. Ribeiro, S. Singh, and C. Guestrin. Why should i trust you? explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016.

[155] M. T. Ribeiro, S. Singh, and C. Guestrin. Anchors: High-precision model-agnostic explanations. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[156] L. Rice, E. Wong, and J. Z. Kolter. Overfitting in adversarially robust deep learning. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org, 2020.

[157] D. Rolnick, P. L. Donti, L. H. Kaack, K. Kochanski, A. Lacoste, K. Sankaran, A. S. Ross, N. Milojevic-Dupont, N. Jaques, A. Waldman-Brown, et al. Tackling climate change with machine learning. *ACM Computing Surveys (CSUR)*, 55(2):1–96, 2022.

[158] A. Ross and F. Doshi-Velez. Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

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

[160] B. S. and R. Nagapadma. RT-IoT2022 . UCI Machine Learning Repository, 2024. DOI: https://doi.org/10.24432/C5P338.

[161] V. P. Saikrishnan and D. M. Karthikeyan. Automated object detection and classification using metaheuristics with deep learning on surveillance videos. *2023 International Conference on Sustainable Computing and Data Communication Systems (ICSCDS)*, pages 29–34, 2023.

[162] W. Samek, G. Montavon, A. Vedaldi, L. K. Hansen, and K.-R. Müller. *Explainable AI: interpreting, explaining and visualizing deep learning*, volume 11700. Springer Nature, 2019.

[163] S. Samuel, F. Löffler, and B. König-Ries. Machine learning pipelines: provenance, reproducibility and fair data principles. In *Provenance and Annotation of Data and Processes*, pages 226–230. Springer, 2020.

[164] S. Schelter, J.-H. Boese, J. Kirschnick, T. Klein, and S. Seufert. Automatically tracking metadata and provenance of machine learning experiments. 2017.

[165] A. D. Selbst. Disparate impact in big data policing. *Georgia Law Review*, 52:109, 2017.

[166] A. D. Selbst, D. Boyd, S. A. Friedler, S. Venkatasubramanian, and J. Vertesi. Fairness and abstraction in sociotechnical systems. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, FAT* '19, page 59–68, New York, NY, USA, 2019. Association for Computing Machinery.

[167] J. Selva, A. S. Johansen, S. Escalera, K. Nasrollahi, T. B. Moeslund, and A. Clapés. Video transformers: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(11):12922–12943, 2023.

[168] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

[169] L. S. Shapley. A value for n-person games, contributions to the theory of games, 2, 307–317, 1953.

[170] R.-K. Sheu and M. S. Pardeshi. A survey on medical explainable ai (xai): Recent progress, explainability approach, human interaction and scoring system. *Sensors*, 22(20):8068, 2022.

[171] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-c. Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. *Advances in neural information processing systems*, 28, 2015.

[172] A. Shrikumar, P. Greenside, A. Shcherbina, and A. Kundaje. Not just a black box: Learning important features through propagating activation differences. *ArXiv*, abs/1605.01713, 2016.

[173] D. Slack, S. Hilgard, E. Jia, S. Singh, and H. Lakkaraju. Fooling lime and shap: Adversarial attacks on post hoc explanation methods. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, pages 180–186, 2020.

[174] N. Sobahi, O. Atila, E. Deniz, A. Sengur, and U. R. Acharya. Explainable covid-19 detection using fractal dimension and vision transformer with grad-cam on cough sounds. *Biocybernetics and Biomedical Engineering*, 42(3):1066–1080, 2022.

[175] W. Song, C. Cong, H. Zhong, and J. Xue. Correction-based defense against adversarial video attacks via discretization-enhanced video compressive sensing. In *33rd USENIX Security Symposium*, pages 3603–3620, 2024.

[176] S. Srinivas and F. Fleuret. Full-gradient representation for neural network visualization. *Advances in neural information processing systems*, 32, 2019.

[177] A. Stergiou. Analyzing deep learning architectures for video action recognition. *arXiv preprint arXiv:2107.11681*, 2021.

[178] A. Stergiou, G. Kapidis, G. Kalliatakis, C. Chrysoulas, R. Poppe, and R. Veltkamp. Class feature pyramids for video explanation. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 4255–4264, 2019.

[179] E. Štrumbelj and I. Kononenko. Explaining prediction models and individual predictions with feature contributions. *Knowledge and information systems*, 41(3):647–665, 2014.

[180] S. Studer, T. B. Bui, C. Drescher, A. Hanuschkin, L. Winkler, S. Peters, and K.-R. Müller. Towards crisp-ml (q): a machine learning process model with quality assurance methodology. *Machine learning and knowledge extraction*, 3(2):392–413, 2021.

[181] J. Su, D. V. Vargas, and K. Sakurai. One pixel attack for fooling deep neural networks. *IEEE Transactions on Evolutionary Computation*, 23(5):828–841, 2019.

[182] M. Sundararajan, A. Taly, and Q. Yan. Axiomatic attribution for deep networks. In *International conference on machine learning*, pages 3319–3328. PMLR, 2017.

[183] Swagger. Openapi specification, 2023. Accessed: 2023.

[184] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[185] G. Tao, S. Ma, Y. Liu, and X. Zhang. Attacks meet interpretability: Attribute-steered detection of adversarial samples. *Advances in neural information processing systems*, 31, 2018.

[186] M. Testi, M. Ballabio, E. Frontoni, G. Iannello, S. Moccia, P. Soda, and G. Vessio. Mlops: A taxonomy and a methodology. *IEEE Access*, 10:63606–63618, 2022.

[187] A. J. Thirunavukarasu, D. S. J. Ting, K. Elangovan, L. Gutierrez, T. F. Tan, and D. S. W. Ting. Large language models in medicine. *Nature Medicine*, 29(8):1930–1940, Aug 2023.

[188] Z. Tong, Y. Song, J. Wang, and L. Wang. Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. *Advances in neural information processing systems*, 35:10078–10093, 2022.

[189] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 4489–4497, 2015.

[190] B. H. Van der Velden, H. J. Kuijf, K. G. Gilhuijs, and M. A. Viergever. Explainable artificial intelligence (xai) in deep learning-based medical image analysis. *Medical Image Analysis*, 79:102470, 2022.

[191] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[192] G. Vilone and L. Longo. Notions of explainability and evaluation approaches for explainable artificial intelligence. *Information Fusion*, 76:89–106, 2021.

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

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

[195] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. V. Gool. Temporal segment networks: Towards good practices for deep action recognition. In *European Conference on Computer Vision*, 2016.

[196] Y. Wang, M. Long, J. Wang, Z. Gao, and P. S. Yu. Predrnn: Recurrent neural networks for predictive learning using spatiotemporal lstms. *Advances in neural information processing systems*, 30, 2017.

[197] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

[198] Z. Wang and Y. Liu. Cloud-Based XAI Services for Assessing Open Repository Models Under Adversarial Attacks . In *2024 IEEE International Conference on Software Services Engineering (SSE)*, pages 141–152, Los Alamitos, CA, USA, July 2024. IEEE Computer Society.

[199] Z. Wang and Y. Liu. Staa: Spatio-temporal attention attribution for real-time interpreting transformer-based ai video models. *IEEE Access*, pages 1–1, 2025.

[200] Z. Wang, Y. Liu, A. Arumugam Thiruselvi, and A. Hamou-Lhadj. Xaiport: A service framework for the early adoption of xai in ai model development. In *Proceedings of the 46th International Conference on Software Engineering (ICSE 2024), New Ideas and Emerging Results*, Lisbon, Portugal, April 2024. ACM.

[201] Z. Wang, Y. Liu, and J. Huang. An open api architecture to discover the trustworthy explanation of cloud ai services. *IEEE Transactions on Cloud Computing*, 2024.

[202] Z. Wang, Y. Liu, A. A. Thiruselvi, and A. Hamou-Lhadj. Xaiport: A service framework for the early adoption of xai in ai model development. *arXiv preprint arXiv:2403.16858*, 2024.

[203] Z. Wang, C. Sha, and S. Yang. Reinforcement learning based sparse black-box adversarial attack on video recognition models. *arXiv preprint arXiv:2108.13872*, 2021.

[204] X. Wei, J. Zhu, S. Yuan, and H. Su. Sparse adversarial perturbations for videos. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8973–8980, 2019.

[205] Z. Wei, J. Chen, X. Wei, L. Jiang, T.-S. Chua, F. Zhou, and Y.-G. Jiang. Heuristic black-box adversarial attacks on video recognition models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 12338–12345, 2020.

[206] Z. Wei, J. Chen, Z. Wu, and Y.-G. Jiang. Cross-modal transferable adversarial attacks from images to videos. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15064–15073, 2022.

[207] K. Werder, B. Ramesh, and R. Zhang. Establishing data provenance for responsible artificial intelligence systems. *ACM Transactions on Management Information Systems (TMIS)*, 13(2):1–23, 2022.

[208] S. E. Whang and J.-G. Lee. Data collection and quality challenges for deep learning. *Proceedings of the VLDB Endowment*, 13(12):3429–3432, 2020.

[209] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest, and A. M. Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, Oct. 2020. Association for Computational Linguistics.

[210] H. Wu, B. Xiao, N. Codella, M. Liu, X. Dai, L. Yuan, and L. Zhang. Cvt: Introducing convolutions to vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 22–31, 2021.

[211] C. Xiao, B. Li, J.-Y. Zhu, W. He, M. Liu, and D. Song. Generating adversarial examples with adversarial networks. *arXiv preprint arXiv:1801.02610*, 2018.

[212] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *Advances in Neural Information Processing Systems*, 34:12077–12090, 2021.

[213] S. Xie, C. Sun, J. Huang, Z. Tu, and K. Murphy. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *Proceedings of the European conference on computer vision (ECCV)*, pages 305–321, 2018.

[214] H. Yang, C. Rudin, and M. Seltzer. Scalable bayesian rule lists. In *International conference on machine learning*, pages 3921–3930. PMLR, 2017.

[215] W. Yang, H. Le, S. Savarese, and S. C. Hoi. Omnixai: A library for explainable ai. *arXiv preprint arXiv:2206.01612*, 2022.

[216] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32, 2019.

[217] G. X. Yu, Y. Gao, P. Golikov, and G. Pekhimenko. Habitat: A runtime-based computational performance predictor for deep neural network training. In *USENIX Annual Technical Conference*, 2021.

[218] Z. Yuan, X. Song, L. Bai, W. gang Zhou, Z. Wang, and W. Ouyang. Temporal-channel transformer for 3d lidar-based video object detection for autonomous driving. *IEEE Transactions on Circuits and Systems for Video Technology*, 32:2068–2078, 2020.

[219] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, and Y. Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6023–6032, 2019.

[220] J. Zacharias, M. von Zahn, J. Chen, and O. Hinz. Designing a feature selection method based on explainable artificial intelligence. *Electronic Markets*, 32(4):2159–2184, 2022.

[221] F. Zafar, A. Khan, S. Suhail, I. Ahmed, K. Hameed, H. M. Khan, F. Jabeen, and A. Anjum. Trustworthy data: A survey, taxonomy and future trends of secure provenance schemes. *Journal of network and computer applications*, 94:50–68, 2017.

[222] M. Zaharia, A. Chen, A. Davidson, A. Ghodsi, S. A. Hong, A. Konwinski, S. Murching, T. Nykodym, P. Ogilvie, M. Parkhe, et al. Accelerating the machine learning lifecycle with mlflow. *IEEE Data Eng. Bull.*, 41(4):39–45, 2018.

[223] H. Zhang, X. Li, and L. Bing. Video-llama: An instruction-tuned audio-visual language model for video understanding. *arXiv preprint arXiv:2306.02858*, 2023.

[224] H. Zhang, Y. Yu, J. Jiao, E. P. Xing, L. E. Ghaoui, and M. I. Jordan. Theoretically principled trade-off between robustness and accuracy. *ArXiv*, abs/1901.08573, 2019.

[225] J. Zhang and C. Li. Adversarial examples: Opportunities and challenges. *IEEE transactions on neural networks and learning systems*, 31(7):2578–2593, 2019.

[226] Y. Zhang, F. Xu, J. Zou, O. L. Petrosian, and K. V. Krinkin. Xai evaluation: Evaluating black-box model explanations for prediction. In *2021 II International Conference on Neural Networks and Neurotechnologies (NeuroNT)*, pages 13–16, 2021.

[227] Z. Zhang, H. Al Hamadi, E. Damiani, C. Y. Yeun, and F. Taher. Explainable artificial intelligence applications in cyber security: State-of-the-art in research. *IEEe Access*, 10:93104–93139, 2022.

[228] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929, 2016.

[229] M. Zhou, J. Wu, Y. Liu, S. Liu, and C. Zhu. Dast: Data-free substitute training for adversarial attacks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 234–243, 2020.

[230] Y. Zhu, X. Li, C. Liu, M. Zolfaghari, Y. Xiong, C. Wu, Z. Zhang, J. Tighe, R. Manmatha, and M. Li. A comprehensive study of deep video action recognition. *arXiv preprint arXiv:2012.06567*, 2020.

[231] L. M. Zintgraf, T. S. Cohen, T. Adel, and M. Welling. Visualizing deep neural network decisions: Prediction difference analysis. *arXiv preprint arXiv:1702.04595*, 2017.

[232] M. Zuhair, P. Bhattacharya, V. K. Prasad, M. Barot, and M. Modi. Analysis of boosting mechanisms in cloud-based intrusion detection systems. In *2022 5th International Conference on Contemporary Computing and Informatics (IC3I)*, pages 961–966, 2022.