UC Davis

UC Davis Electronic Theses and Dissertations

Title

Responsible and Efficient AI through Collaborative Intelligence

Permalink

https://escholarship.org/uc/item/8wt0m6j3

ISBN

9798189226939

Author

Xuan, Xiwei

Publication Date

2026-06-25

Peer reviewed|Thesis/dissertation

eScholarship.org Powered by the California Digital Library
University of California

Responsible and Efficient AI through Collaborative Intelligence

By

XIWEI XUAN

DISSERTATION

Submitted in partial satisfaction of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

Computer Science

in the

OFFICE OF GRADUATE STUDIES

of the

UNIVERSITY OF CALIFORNIA

DAVIS

Approved:

Kwan-Liu Ma, Chair

Jiawei Zhang

Liu Ren

Committee in Charge

2026

Copyright © 2026 by

XIWEI XUAN

All rights reserved.

CONTENTS

Abstract vi

Acknowledgments viii

1 Introduction **1**

1.1 Motivation 1

1.2 Background 4

1.2.1 Explainable Decision-making for Trustworthy AI 4

1.2.2 Validation and Enhancement of Models and Data 6

1.2.3 Multimodal AI Systems for Open-world Scaling 7

1.3 Dissertation Outline 10

2 Understanding Model Decisions through Visual Intelligence **12**

2.1 Introduction 13

2.2 Related Work 16

2.2.1 Visual Explanation Methods for Interpretable CNN 16

2.2.2 Visual Analytics for CNN Comparison 17

2.2.3 Causal Perspectives on CNN Explanations 18

2.3 Visual Analytics for Understanding and Comparing CNNs 19

2.3.1 Methodology 20

2.3.2 System Interface 26

2.3.3 Use Cases 29

2.4 Visual Interpretation of CNNs from a Causal Perspective 35

2.4.1 CNN Classification as a Causal Mechanism 36
2.4.2 Method 37
2.4.3 Experiments 40
2.5 Conclusion 48

3 Validating Model Behavior with Human-Aided Analytical Frameworks 49

3.1 Introduction 50
3.2 Related Work 53
3.2.1 Data Slice Finding 53
3.2.2 Visual Analytics for Model Validation 54
3.2.3 Spuriousness Mitigation 54
3.3 Discovering Model Issues through Visual Analytics 55
3.3.1 *AttributionScanner* 55
3.3.2 Case Studies 67
3.3.3 Evaluation 73
3.4 Mitigating Spuriousness with Minimal Annotations 77
3.4.1 Problem Formulation 77
3.4.2 Method 79
3.4.3 Attention Annotation and Expansion 80
3.4.4 Balanced Data Curation 81
3.4.5 Training 83
3.4.6 Experiments 84
3.4.7 Experimental Setup 84

3.4.8 Spuriousness Mitigation Performance (Q1) . . . 85
3.4.9 Attention Labeling Performance (Q2) . . . 88
3.4.10 Ablation Study (Q3) . . . 90
3.5 Conclusion . . . 91

4 Validating Multimodal Data Quality through Visual Analytics 92

4.1 Introduction . . . 93
4.2 Related Work . . . 95
4.2.1 Foundation Model-Involved Data Pipeline. . . . 95
4.2.2 Human-in-the-loop Data Labeling and Validation . . . 95
4.2.3 Open-vocabulary Image Segmentation . . . 96
4.3 Motivation and *VISTA* Framework . . . 97
4.3.1 OVS and Data Pipeline with FMs . . . 97
4.3.2 Challenges for Quality Enhancement . . . 98
4.3.3 Requirements for Quality Enhancement . . . 99
4.3.4 *VISTA* Framework . . . 100
4.4 Method . . . 100
4.4.1 Data Structuring . . . 101
4.4.2 Quality Quantification . . . 102
4.4.3 Data Issue Summarization . . . 103
4.4.4 Online Evaluation . . . 106
4.5 Visual Interface . . . 107
4.5.1 Interface Overview . . . 107

4.5.2 Analytical Workflow . . . 110
4.5.3 Data Issue Resolution . . . 112
4.6 Evaluation . . . 112
4.6.1 Use Case I: Comprehensive Scene Parsing . . . 113
4.6.2 Use Case II: Traffic Scene Understanding . . . 116
4.6.3 Expert Study . . . 118
4.7 Conclusion . . . 120

5 Scaling AI Systems in Open-World Settings via Collaborative Models 122

5.1 Introduction . . . 123
5.2 Related Work . . . 126
5.3 Our Approach . . . 128
5.3.1 Initial Pairing . . . 129
5.3.2 Data Enhancing . . . 129
5.3.3 Similarity-based Retrieval . . . 132
5.4 Experiments . . . 134
5.4.1 Experimental Setup . . . 134
5.4.2 Comparison to the State-of-the-arts . . . 135
5.4.3 Analysis, Ablation, and Discussion . . . 136
5.5 Conclusion . . . 141

6 Conclusion 143

ABSTRACT

Responsible and Efficient AI through Collaborative Intelligence

Over the past decade, artificial intelligence (AI) systems have grown rapidly in terms of scale and capability. While such developments enabled a wide range of opportunities and seemingly impressive performance across various tasks, the significant complexity of modern AI systems has made it substantially challenging to understand their decision processes, analyze their failure cases, adapt them to new domains, or validate the corresponding datasets.

To address the aforementioned challenges, this dissertation advances collaborative intelligence from humans, models, and data for understanding, validating, and scaling AI systems responsibly and efficiently. Instead of seeking ever-larger models or extensive supervision, the introduced approaches integrate intelligent visualization, human-aided analytical frameworks, multimodal representations, and model collaborations to enable more beneficial and practical AI solutions in real-world settings.

To facilitate the effective establishment of AI trustworthiness, this dissertation first investigates how visual intelligence supports the understanding of model decision-making. By developing visualization techniques that systematically expose decision patterns and learned behaviors, the introduced frameworks enable humans to reason about complex predictions beyond aggregate performance metrics, forming a foundation for trustworthy AI systems.

Upon understanding models or before model deployment, there are critical demands for AI system validation and enhancement. The dissertation introduces analytical frameworks for validating and improving models under limited supervision. Through visual analytics, active learning strategies, and human-aided analysis, these approaches enable efficient identification of spu-

rious correlations, failure modes, and reliability issues without requiring exhaustive annotation or heavy computation. The multiple frameworks introduced enable effective and scalable solutions for model assessment and targeted enhancement in resource-limited applications.

Following model validation, the dissertation addresses data quality challenges, another important perspective for responsible AI systems. Specifically, through multimodal learning, representation-level analysis, and visual analytics, the introduced data validation techniques uncover distributional biases, inconsistencies, and quality degradation at scale, effectively and efficiently improving multimodal data quality and supporting reliable learning through data-centric validation.

As foundation models have been enabling increasingly advanced capabilities, the dissertation explores efficient AI scaling in open-world environments. By developing agent collaboration mechanisms, these frameworks support adaptive reasoning, fine-grained perception, and other capability expansion in a cost-efficient manner, allowing AI systems to scale efficiently across tasks, modalities, and evolving environments.

Together, these contributions establish a cohesive framework for responsible and efficient AI scaling through collaborative intelligence. By integrating human insight, analytical reasoning, data-centric validation, and model collaboration, this dissertation advances AI systems to be more interpretable, reliable, and adaptable in real-world applications.

ACKNOWLEDGMENTS

I would like to begin by expressing my deepest gratitude to my advisor, Dr. Kwan-Liu Ma. Without his long-term guidance, encouragement, and support, I would not have been able to complete my Ph.D. journey. Through our many discussions, meetings, and collaborations, I have benefited tremendously from his knowledge and experience. As I prepare to step into academia as an incoming faculty member, I feel even more strongly how fortunate I have been to be his student.

I would also like to sincerely thank my qualifying exam and dissertation committee members: Dr. Ren Liu, Dr. Jiawei Zhang, Dr. Hamed Pirsiavash, and Dr. Hao-Chuan Wang. Their guidance and insightful feedback have greatly shaped my research and improved the quality of my work over the years. I truly appreciate the time and effort they have devoted to supporting me.

I am grateful to all my lab mates in the ViDi Lab for creating such a supportive and collaborative environment. The lab has felt like a close-knit community throughout my Ph.D., regardless of whether members have graduated or are still here. In particular, I would like to thank Oh-Hyun Kwon, who was my first research mentor in the lab and has provided invaluable and continuous guidance since the beginning. I also want to thank Takanori Fujiwara, Qi Wu, Yiran Li, Xiaoyu Zhang, and David Bauer for their support at different stages of my Ph.D. Although I cannot list everyone here, I am sincerely thankful to all Vidians for their friendship and support.

Finally, I would like to express my sincere gratitude to my family. A Ph.D. is a challenging journey, and I would not have done it without their constant support and understanding. I would especially like to remember my grandmother, Juxiang Zhao, who has always been my strongest supporter. She was the strongest and truly inspiring woman I have known: resilient, optimistic, and independent. Her spirit continues to guide and motivate me throughout my life.

Chapter 1

Introduction

1.1 Motivation

Artificial Intelligence (AI) and Machine Learning (ML) systems have become a driving force of the technological transformation [1, 2]. Revolutionary developments have been observed regarding deep neural networks [3, 4, 5], large-scale datasets [6, 7, 8], and foundation models [9, 10, 11]. They expand AI capabilities across a wide range of applications, such as autonomous driving [12], healthcare [13], scientific discovery [14], and advanced manufacturing [15].

However, fundamental challenges have also been exposed along with AI's rapid scaling, including but not limited to concerns about trust, accountability, reliability, and efficiency [16]. Firstly, regarding the increasing complexity of AI systems, it has been more difficult to interpret their decision-making processes [17, 18]. Although explainable AI (XAI) [19] techniques can, to some extent, help make sense of models' decisions, they are often static and instance-based, failing to deliver actionable and scalable insights into models' behavior patterns in general. Such explanations are insufficient for supporting the trustworthiness of AI in high-stakes domains [20].

Beyond interpretability challenges, there are also more demanding requirements for evaluating and improving models or datasets in a more comprehensive manner [21]. For models, conventional

aggregate metrics such as accuracy or F1 score often fail to reflect detailed model performance issues [22, 23, 24, 25, 26]. Consequently, significant model failures may remain hidden even when overall performance appears strong. A common example arises when models learn spurious correlations [27, 28, 23, 29, 30]. For example, for skin cancer classification using skin images, models may rely on artificial patches rather than the skin region to make a decision, leading to brittle predictions when such irrelevant conditions change [28]. In practice, such critical issues are often hidden by high overall performance measured by conventional metrics. Therefore, detecting such failures requires systematic validation mechanisms that can reveal patterns of incorrect behavior and identify their underlying causes.

From a data perspective, a significant and continuous explosion in data volume has been observed across domains. Although helpful for many data-driven AI systems when used carefully, large volumes of data from various sources and modalities also pose unprecedented challenges for data quality assurance [31], making it unclear whether harmful content, spurious correlations, or incorrect labels exist. However, it is a non-trivial task to validate and improve data quality in such extreme volume and complexity [32, 33]. Existing approaches often either sample a small subset for manual quality checks [10, 33] or rely on static metrics for data filtering [34, 35, 36, 32], both of which fail to deliver data of satisfactory quality. This calls for more systematic, comprehensive, and efficient solutions for identifying data quality issues and improving data quality at scale.

As model size continues to grow, we've reached a tipping point where efficiency in data, annotation, and computation is a core requirement for real-world deployment [37, 38]. Adapting models to new tasks or domains often requires additional data collection [39, 40], extensive manual annotation [41, 42], or costly retraining of large-scale models [43, 44, 45]. While such approaches may be manageable in controlled research settings, they often fall apart in the field, where com-

putational resources, response latency, and operational costs are tightly constrained. Whether it's an autonomous vehicle navigating a street [12], a medical system assisting in a surgery [46], or a space probe operating in deep space [47], there simply isn't room for bulky, slow, or resource-heavy AI. To be more practical, AI systems must operate under limited compute budgets while still meeting strict real-time requirements.

The aforementioned challenges and research gaps motivate the need for new approaches to efficiently improve the interpretability, reliability, and scalability of AI systems. Rather than following the common trend of scaling up model size or increasing supervision, this dissertation explores collaborative intelligence, where the strengths of data, human insight, and machine intelligence are integrated to improve AI systems [26, 48, 49, 28, 50, 31]. By leveraging human reasoning for interpretation, structured analytical frameworks for validation, and collaborative mechanisms among AI components, it becomes possible to enable responsible and efficient AI in real-world settings.

Problem Statement. Three fundamental AI challenges remain insufficiently addressed:

• *Limited understanding of model behavior.* The decision processes of complex models remain difficult to interpret, making it challenging to reason about why models succeed or fail.

• *Insufficient validation of models and data.* Existing evaluation practices often fail to reveal systematic model failures, dataset biases, or spurious correlations that affect reliability.

• *Inefficient mechanisms for extending AI capabilities.* Extending models to new tasks or modalities often requires costly data collection, extensive annotation, and substantial computation.

Addressing these challenges requires approaches that integrate interpretability, systematic validation, and scalable improvement strategies within a unified framework.

Research Questions. Motivated by the challenges above, this dissertation investigates the follow-

ing research questions:

- How can humans effectively understand the decision-making processes of complex AI models beyond single-instance explanations and aggregate performance metrics?
- How can large-scale model failures and reliability issues be systematically detected and analyzed without extensive supervision?
- How can validation be performed for large-scale multimodal data, particularly synthetic data generated by foundation models?
- How can AI systems be efficiently extended to new tasks and environments without relying solely on costly retraining or large-scale supervision?

This dissertation addresses these questions by developing methods that integrate intelligent visualizations, human-aided analytical frameworks, multimodal data analysis, efficient learning strategies, and collaborative models.

1.2 Background

This section situates our work within the broader context of explainable AI (XAI), model and data validation, and multimodal systems.

1.2.1 Explainable Decision-making for Trustworthy AI

The push for trustworthy AI has centered on making the "black box" of model reasoning more transparent [19, 51]. Visualization has been a cornerstone of this effort, as it translates high-dimensional computational processes into intuitive visual maps [52]. Early techniques like Layer-wise Relevance Propagation (LRP) [53] and Class Activation Maps (CAM) [54] laid the groundwork by using heatmaps to pinpoint which input regions drive a prediction. This lineage evolved into gradient-based methods like Grad-CAM and its successors [55, 56, 57], which offer more

flexible visual explanations across diverse neural architectures.

Building upon these early techniques, a diverse family of interpretation methods has been developed to provide insight into how machine learning systems make predictions [58, 59, 60, 54, 55, 56]. One group of methods aims to visualize the features extracted by a model through backward-pass analysis, including Vanilla Backpropagation [61], Guided Backpropagation [60], and Deconvolution [62]. These methods can produce fine-grained visualizations, but they often fail to reveal information that is directly related to model decisions. Feature visualization approaches [58, 59] instead attempt to visualize decision-relevant features by maximizing the activations of specific neurons or layers. In addition, saliency-based techniques highlight spatial regions that are most influential for a given prediction, including perturbation-based methods [63, 64] and CAM-based approaches [55, 56, 65, 66, 67]. Together, these approaches provide useful tools for understanding deep models and examining how they associate input patterns with predicted outcomes.

Beyond instance-level explanations, visualization and visual analytics techniques can further support in-depth understanding of AI systems by integrating multiple visual components and interactive analysis tools [68, 69, 70, 71, 72, 73, 74]. Such approaches enable analysts to explore model behaviors across datasets, compare prediction patterns, and identify systematic errors. In practical applications, interactive exploration plays a crucial role in fostering human understanding and generating valuable insights [75, 73, 76, 77]. For instance, by interactively manipulating input features or visualizing intermediate representations, users can observe how changes in the input influence model predictions and better understand the decision boundaries learned by the model [78]. These capabilities allow practitioners to reason about model behavior more effectively and identify potential weaknesses in the learned representations. Together, they illustrate how visualization can bridge the gap between complex AI models and human reasoning.

While significant progress has been made, most existing explanation methods remain limited in their ability to provide actionable insights into model behavior [26, 79]. For example, instance-level explanations may reveal which features influence a particular prediction, but they do not necessarily explain broader patterns of model behavior across datasets or tasks. Consequently, there remains a need for methods that support deeper reasoning about model decision-making and enable humans to understand how models generally behave across different contexts.

1.2.2 Validation and Enhancement of Models and Data

Understanding model decisions alone is not sufficient to ensure the reliability of AI systems [80, 16]. As machine learning models are increasingly deployed in real-world environments, systematic validation becomes essential for detecting hidden failure modes, biases, and inconsistencies in both models and the data on which they rely [79, 26, 81]. Identifying such issues is critical for improving model robustness, transparency, and accountability in practical applications.

One important line of research focuses on uncovering systematic model failures through subgroup analysis or *data slice discovery*. Machine learning models often exhibit heterogeneous performance across different subsets of data, even when their overall accuracy appears high [22, 23, 24, 25]. A data slice refers to a subgroup of instances that share coherent characteristics and exhibit similar model behavior [22, 23, 24, 25, 82, 79]. By identifying slices with unusually poor performance or incorrect behavioral patterns, practitioners can diagnose hidden failure modes that would otherwise remain obscured when examining aggregate metrics. However, many existing slice discovery methods rely on extensive metadata annotations [83, 84, 85] or metadata generated by auxiliary models such as vision-language systems [86]. Obtaining such metadata can be costly, while automatically generated descriptions may be unreliable in domain-specific scenarios [31].

In parallel, XAI techniques have also been used to support model validation by revealing the

evidence that models rely on when making predictions. Attribution-based explanations [63, 54, 55, 64] allow practitioners to inspect which input regions influence model outputs. When combined with human analysis, these explanations can help identify spurious correlations [27], dataset biases, or mislabeled samples that degrade model reliability. However, most existing explanation techniques operate at the instance level, producing explanations for individual predictions without providing scalable mechanisms for systematically validating model behavior across large datasets.

The challenge becomes even more pronounced as modern AI systems increasingly rely on large-scale and multimodal datasets. Data issues such as distribution shifts, inconsistent annotations, and hidden biases can significantly affect model performance. These challenges have motivated the emerging paradigm of data-centric AI [87], which emphasizes improving dataset quality rather than focusing solely on model architectures. Nevertheless, many existing approaches still rely on static evaluation metrics [34, 35, 36, 32] or manual inspection of downsampled datasets [10, 33], which limits their effectiveness when applied to large-scale, heterogeneous data.

Consequently, efficiently validating and improving the quality of both models and datasets has become a critical requirement for the responsible deployment of modern AI systems. Addressing this challenge requires systematic validation frameworks that integrate model explanations, data analysis, and human insight to support scalable diagnosis and targeted improvement of AI systems.

1.2.3 Multimodal AI Systems for Open-world Scaling

Complex real-world environments demand AI systems to operate beyond the assumptions of closed-world learning [88, 89]. Such scenarios often involve heterogeneous data modalities, evolving tasks, and dynamic environments that cannot be fully captured by fixed training datasets or predefined label spaces. These challenges require AI systems that can flexibly integrate information, adapt to new concepts, and extend their capabilities without extensive retraining.

Multimodal learning has emerged as a promising direction for addressing this challenge by enabling models to jointly reason over multiple modalities such as images, language, audio, and sensor data [90, 91, 92]. By learning shared representations across modalities, multimodal models can capture richer semantic relationships and support a wider range of perception and reasoning tasks. Recent advances in large-scale vision-language models (VLMs) [35] and other foundation models [93, 94] have further accelerated this progress by providing general-purpose representations that can be transferred across tasks with minimal supervision.

However, simply scaling individual foundation models does not fully address the open-world challenges [95]. As models become increasingly complex, interactions among multiple models, data sources, and reasoning processes must be carefully coordinated. Therefore, recent research has begun exploring collaborative AI systems [96], where multiple models interact to combine complementary strengths and perform complex tasks that would be difficult for a single model.

One important and representative example of open-world scaling arises in visual perception. Open-vocabulary semantic segmentation (OVS) aims to segment images according to arbitrary textual concepts without requiring a fixed predefined label set. As such, it reflects a broader goal of modern AI systems: extending existing models to recognize more fine-grained concepts in flexible and open-ended environments. Many existing approaches address OVS by introducing additional densely annotated segment-text datasets and further training or fine-tuning large vision-language models [97, 98, 99, 100, 101]. While effective in terms of performance, these solutions incur substantial annotation and computational costs. This dependence on extra supervision is particularly noteworthy given that foundation models such as CLIP have already been trained on hundreds of millions of image-text pairs [35]. The need for further large-scale annotation and optimization suggests that simply continuing to scale training data and model size may not be the

most efficient path for extending model capabilities.

Existing models often already possess complementary strengths that remain underutilized. For example, vision-language foundation models [35] provide broad semantic alignment across images and text, while vision models such as DINOv2 [102] offer strong fine-grained visual perception capabilities. Similarly, multimodal large language models [103, 104, 105] can support richer image understanding and more detailed semantic reasoning. These observations suggest that the challenge is not merely the absence of capability, but the lack of principled mechanisms for coordinating and leveraging the capabilities that different models already possess.

This gap motivates the study of collaborative multimodal AI systems for efficient open-world scaling [106]. Rather than relying on manual annotation or task-specific fine-tuning, the systems seek to unlock new capabilities efficiently through coordinated interactions among models, data representations, and external knowledge sources. Developing such mechanisms is essential for enabling AI systems to scale more effectively in open-world environments while maintaining practical efficiency in data usage and computation.

Taken together, these research directions address complementary challenges in building responsible and scalable AI systems. Explainability methods aim to improve human understanding of model decision-making, validation techniques focus on detecting and mitigating systematic failures in models and data, and multimodal AI systems explore how AI capabilities can scale to open-world environments with evolving tasks and heterogeneous information sources.

This dissertation connects these research directions through the concept of collaborative intelligence. Rather than addressing interpretability, validation, and capability expansion as independent problems, the work presented here investigates how intelligent visualizations, human-aided analytical frameworks, multimodal data analysis, efficient learning strategies, and collaborative AI

systems can jointly improve the reliability, scalability, and adaptability of modern AI systems.

1.3 Dissertation Outline

This dissertation investigates how collaborative intelligence can support the understanding, validation, and scaling of AI systems. The remainder of this dissertation is organized as follows.

Chapter 2 investigates how visual intelligence can support the understanding of model decision-making. It introduces visualization and explanation techniques that expose patterns in model predictions and learned representations, enabling humans to analyze complex model behaviors beyond aggregate performance metrics. Through interactive visual analytics and causally grounded explanations, this chapter provides tools for interpreting model decisions and identifying potential reliability issues in deep neural networks.

Chapter 3 focuses on validating model behavior through human-aided analytical frameworks. It presents visual analytics techniques for systematically discovering failure modes such as spurious correlations and mislabeled data, as well as data-efficient strategies that mitigate these issues with minimal human supervision. These approaches enable scalable diagnosis and improvement of model reliability in real-world deployment scenarios.

Chapter 4 extends validation from models to data by addressing the quality of large-scale multimodal datasets. This chapter introduces visual analytics and representation-level analysis techniques for identifying distributional biases, inconsistencies, and quality degradation in multimodal data, enabling more reliable model training through data-centric validation.

Finally, Chapter 5 explores how AI systems can scale to open-world environments through collaborative models. It investigates mechanisms that allow multiple models and data representations to interact and complement one another, enabling AI systems to extend their capabilities to new

concepts and tasks without relying solely on costly retraining or extensive manual annotation.

Together, the research works presented in these chapters establish a unified framework for improving the interpretability, reliability, and scalability of modern AI systems. By integrating human insight, visual analytics, data-centric validation, and collaborative model interactions, this dissertation advances collaborative intelligence as a practical paradigm for building more trustworthy and adaptable AI systems.

Chapter 2

Understanding Model Decisions through Visual Intelligence

As deep neural networks (DNNs) have become central to modern artificial intelligence systems, understanding how and why models make specific decisions is essential for trustworthy AI. In this chapter, convolutional neural networks (CNNs) are used as a representative class of DNNs to study model decision-making in the context of computer vision. Existing practices often rely on aggregate performance metrics or isolated visual explanations, which are insufficient for capturing complex decision behaviors or for systematically comparing multiple models. This chapter advances visual intelligence as a foundation for understanding model decisions by integrating interactive visual analytics with principled interpretation from a causal perspective. It first presents a visual analytics approach that enables in-depth inspection and comparison of CNN models at scale, supporting flexible, task-driven analysis of both quantitative characteristics and qualitative behaviors beyond pairwise comparisons. Building on this system-level understanding, the chapter then investigates visual interpretation of model decisions through complementary notions of necessity and sufficiency, providing more informative and convincing explanations across architectures and datasets. Together, these contributions establish a cohesive framework for understanding model

decisions through visual intelligence, laying the groundwork for subsequent chapters on efficient model and data validation.

2.1 Introduction

In recent years, deep neural networks (DNNs) have driven unprecedented progress across a wide range of application domains, particularly in computer vision, where convolutional neural networks (CNNs) have become a dominant and representative model class [107, 108, 109, 5, 110, 111, 112, 113, 114]. Along with these advances, ML practitioners with diverse technical backgrounds increasingly face the practical need to understand and compare multiple CNN models when selecting, deploying, or refining models for real-world tasks. Such comparison tasks are particularly challenging for novices with foundational but incomplete ML knowledge, especially when the number of candidate models is large and their architectural designs and behavioral characteristics vary substantially. For example, a graduate student in a medical domain may wish to adopt a CNN for disease detection; yet filtering tens of available architectures and identifying models with desirable behavioral properties remains nontrivial.

Conventional approaches for comparing CNNs often focus on architecture-level analysis [115, 116, 117] or static quantitative metrics such as accuracy and computational efficiency [118, 115, 119]. While informative, such comparisons provide limited insight into why different models behave differently on the same data. This limitation motivates approaches that improve transparency and support model selection through deeper, behavior-oriented understanding.

An interpretable CNN comparative study can be broadly divided into two complementary phases: model interpretation and model comparison. For model interpretation, theXAI community has proposed a variety of post-hoc visual explanation techniques that highlight input regions

relevant to a model's decision [19, 54, 55, 120, 56, 57, 66, 67]. A prominent family of techniques is Class Activation Map (CAM)-based explanations [54], which produce saliency maps by associating spatial regions of an input image with a predicted class. CAM's architectural constraints motivated gradient-weighted variants [55, 56, 57], but gradient saturation and vanishing can yield noisy explanations [121]. To bypass gradient limitations, contribution-based methods such as Score-CAM [66] and Group-CAM [67] estimate feature importance by measuring changes in model outputs under feature masking. Although these techniques can improve decision interpretability for a single model, they are typically used in isolation and are rarely integrated into workflows for systematically comparing many models.

For model comparison, a number of visual analytics systems enable interactive comparison of CNNs from multiple perspectives, such as feature activations or parameter distributions [71, 72, 73, 74, 122, 123]. Some tools support multi-model comparison, but they often lack integrated interpretability [71, 72, 73, 74] or restrict comparison to only two models at a time [72, 122, 123]. As the number of available CNN architectures continues to grow, there is a pressing need to consolidate advances from both interpretation and comparison into a unified framework that supports flexible, scalable, and interpretable comparative studies.

To address these challenges, this chapter advances visual intelligence as a foundation for understanding CNN decision-making. The first part of the chapter presents a visual analytics approach for understanding and comparing CNNs at scale, building upon the VAC-CNN system. It enables in-depth inspection of individual models as well as comparative studies across tens of models, integrating quantitative characteristics with qualitative evidence in a highly interactive workflow. In particular, VAC-CNN coordinates multiple class-discriminative visual explanation methods, including Grad-CAM [55], BBMP [120], Grad-CAM++ [56], Smooth Grad-CAM++ [57], and

Score-CAM [66], and presents analysis results progressively to support task-driven reasoning beyond static metrics or pairwise comparison.

Beyond system-level comparative understanding, trustworthy interpretation of individual decisions also depends on whether explanations align with how humans naturally reason about "why" questions. From a causal perspective, explanations should account for complementary notions of necessity and sufficiency [124, 125, 126, 127]. Many CAM-style explanations can be viewed as emphasizing sufficiency (stability of an outcome given preserved features), while the necessity perspective (how outcomes change under counterfactual feature changes) is often missing. Causal explanation formalisms based on causal effect [128] and responsibility [129] provide relevant foundations, yet they pose challenges for producing precise attributions and frequently overlook sufficiency. Prior efforts to incorporate causality into CNN explanations, such as CexCNN [130] and related approaches [131], highlight the promise of causal grounding but also reveal open challenges in interpretability, precision, and completeness. Motivated by these gaps, the second part of this chapter introduces *SUNY* (**SU**fficiency and **N**ecessit**Y** causal explanation), a framework that explicitly leverages necessity and sufficiency to better rationalize CNN model decisions.

Together, VAC-CNN and *SUNY* establish complementary perspectives on visual intelligence for understanding model decisions—VAC-CNN at the system and comparative level, and *SUNY* at the causal and decision level. By combining interactive visual analytics for comparative reasoning with causally grounded interpretation for decision-level insight, this chapter lays the methodological groundwork for subsequent chapters on model and data validation, where understanding model behavior becomes a prerequisite for efficient and responsible AI validation at scale.

2.2 Related Work

2.2.1 Visual Explanation Methods for Interpretable CNN

Visual explanation methods play an essential role in improving the transparency of deep CNN models. According to the visualization purpose, the existing visual explanation methods can be grouped into three kinds. The first group of methods mainly focus on visualizing the activations of neurons and layers inside a specific model, such as Feature Visualization[58] and Deep Dream[59]. These methods focus on exploring a single model's internal operation mechanism, which is not scalable for comparing multiple models. The second group of methods represents the view of an entire model, which visualizes all extracted features of a model without highlighting decision-related information, such as Vanilla Backpropogation[61], Guided Backpropogation[60], and Deconvolution[62]. This group of explanations' primary processing method is the backward pass, which is time-efficient and can produce fine-grained results. However, this group of methods fails to explain models' decision-making convincingly because they indistinctively represent all the extracted information.

The third group of methods is the class-discriminative visual explanation [54, 55, 120, 56, 57, 66], which can explain the model decision by localizing the regions essential for model predictions and is sensitive to different classes. Zhou et al. [54] introduce CAM (Class Activation Map), which is an initial approach of localizing a specific image region for a given image class. However, researchers have to re-train the entire model to get the results of CAM. As an approach to address the shortcomings of CAM, Grad-CAM [55] is proposed as a more efficient approach, which can explain the predictions of CNN models without re-training or changing their structure. In 2017, a perturbation-based method called BBMP [120] was introduced, which highlights the ROI of input images with the help of perturbations on input images. Since BBMP requires additional

pre-processing and multiple iterations, it was time-consuming and challenging to be implemented in real-time applications. Recently, plenty of Grad-CAM-inspired methods have been proposed including Grad-CAM++[56], Smooth Grad-CAM++[57], and Score-CAM[66]. Consistent with Grad-CAM, these methods are applicable to a wide variety of CNN models.

Aiming to provide interpretable CNN model comparison, we include multiple class-discriminative visual explanation methods to support the understanding of models' decisions.

2.2.2 Visual Analytics for CNN Comparison

A variety of visual analytics tools are aiming at supporting interpretable comparisons of CNN. Some focus on visualizing and interpreting the internal working mechanism of a single CNN model [68, 69, 70, 132, 133, 75, 76], combining various visualization techniques such as dimension reduction for understanding networks' hidden activities [68], a directed acyclic graph to disclose multiple neurons' facets and interactions [69], hierarchy analysis of similar classes [132], or feature visualizations and interactions [75]. However, such in-depth inspection helps develop interpretation but is insufficient for scenarios where model comparison and selection are needed.

Researchers have developed some visual analytics frameworks for comparing CNN models [71, 72, 73, 74, 122, 123, 134, 77]. Prospector [71] leverages partial dependence plots to visualize different performances of multiple models on one input sample. To assist model training, CNN Comparator [72] compares models from different training stages in the aspect of model structures, parameter distributions, etc. Utilizing predictions of the labels, Manifold [73] allows users to compare multiple models at the feature level using scatter plots. BEAMES [74] is a multimodel steering system providing multi-dimensional inspection to help domain experts in model selection. However, these methods lack interpretability because they mainly use numerical features of CNNs. To assist interpretable comparison, researchers apply techniques such as linking model structures

instances for comparing two binary classifiers [122], visualizing qualitative differences in how models interpret input data [123], etc. These techniques can assist better model interpretation, but only support the comparison among a small number of models.

In conclusion, most of the existing visual analytics methods for interpretable CNN comparison are either based on handcrafted quantitative parameters or only supporting comparison between two models. Only few of them allow CNN interpretation and multi-model comparison at the same time. With comparing and interpreting different CNN models becoming a growing demand, there is a need for comparative studies that support a larger number of models' comparisons and present quantitative and qualitative information at the same time for more thorough evaluations.

2.2.3 Causal Perspectives on CNN Explanations

Based on the HP causality [135, 136] introduced by Halpern and Pearl, recent research for causal importance quantification involves two main genres, *causal effect* and *responsibility*. The average *causal effect* [128] measures the expected outcome difference when the values of selected causes are changed, which regards the set of causes as a whole. An individual cause can thus affect the outcome very differently, i.e., receive completely different effect quantification, when being analyzed with different sets of interacting causes [137]. The second genre, the degree of *responsibility* [129], quantifies the minimal set of causes whose removal leads to a satisfactory outcome change, which overlooks the actual value of the outcome [138, 139]. Besides, current *causal effect* and *responsibility* both focus on the *N* perspective, i.e., the change of causes, and the other *S* perspective needs to be fulfilled for a thorough explanation [124, 125, 126, 127]. More recent works such as [127] and [140] address this deficiency by measuring the causal importance from both *N* and *S*. However, their proposed quantification focuses on the probability of the outcome change when causes are changed (*N*) or unchanged (*S*), which also misses the actual outcome value.

Given recent works on CNN causal explanations, two main groups exist according to the type of hypothesized causes, the model's input features, or inner filters. For the first group, [141] and [131] estimate the average *causal effect* of input features. The Shapley value-inspired methods [142, 143, 144, 145, 146] also implicitly analyze the causality of input features from the *S* perspective. However, these methods are too computationally intensive or only provide a hazy outline of salient regions. Two CAM-based methods, Score-CAM [66] and Group-CAM [67], as discussed before, can be regarded as partially causal-grounded by analyzing the *sufficiency* of input features. On the other side, the perturbation-based methods [62, 63, 64] analyze input features' *necessity*.

The second group interprets CNN through model filter intervention, such as [147], which explores CNN's inner workings by ranking filters according to their counterfactual (CF) influence. However, without any visualizations, this causal explanation is less intuitive than saliency maps. CexCNN [130] provides visualizations with the help of CAMs and uses *responsibility* to quantify the CNN's filter importance, which means only the *N* aspect has been analyzed.

2.3 Visual Analytics for Understanding and Comparing CNNs

To operationalize visual intelligence for understanding model decisions, this section presents a visual analytics approach for systematic inspection and comparison of convolutional neural networks. While individual explanation methods can reveal localized decision evidence for a single model, practical model selection and deployment often require reasoning across multiple architectures, performance characteristics, and behavioral patterns. Therefore, understanding CNN behavior in realistic settings involves not only interpreting isolated predictions but also comparing models at different levels of abstraction.

Building upon the VAC-CNN system, this section introduces an interactive visual analytics

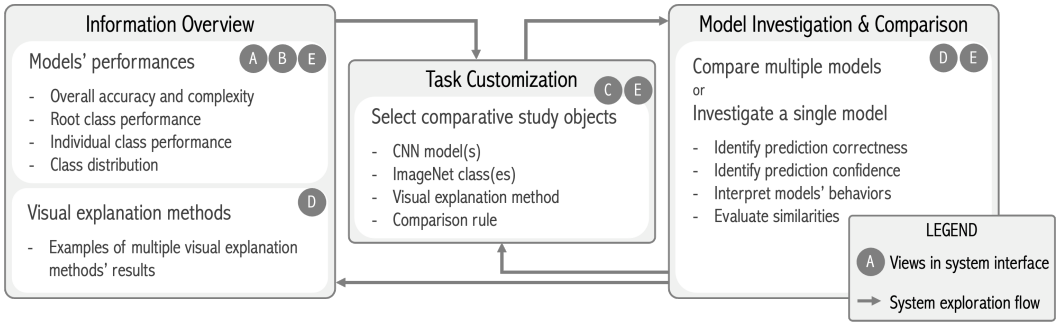


Figure 2.1: The comparative analysis workflow of VAC-CNN consists of three phases: information overview, task customization, and model investigation & comparison. Each phase involves user interaction with multiple views of our system.

framework that integrates quantitative metrics, qualitative explanations, and comparative reasoning into a unified workflow. The framework supports flexible analysis scenarios, ranging from high-level screening of many candidate models to detailed inspection of a single model's behavior. By coordinating multiple visual explanation techniques and presenting results progressively, the system enables users to investigate decision patterns, identify behavioral consistencies and discrepancies, and reason about model differences in a task-driven manner.

2.3.1 Methodology

VAC-CNN is built upon thirteen widely-used models, to cover various state-of-the-art architectures such as AlexNet[109], ResNet [5], SqueezeNet [110], DenseNet [111], MobileNet [112], and ShuffleNet [113]. The models are pre-trained on the ImageNet dataset[148] for the image classification task and we develop our system on the ImageNet (ILSVRC2012) validation set with $1,000$ image classes and $50,000$ images. In this section, we introduce the analysis workflow and the integrated methodologies of our system.

Workflow. We model the comparative analysis procedure with VAC-CNN as a three-phase workflow (see Fig.2.1). The workflow starts from **Phase 1**, which provides an information overview to help ML beginners get a general understanding of model performances and visual explanations.

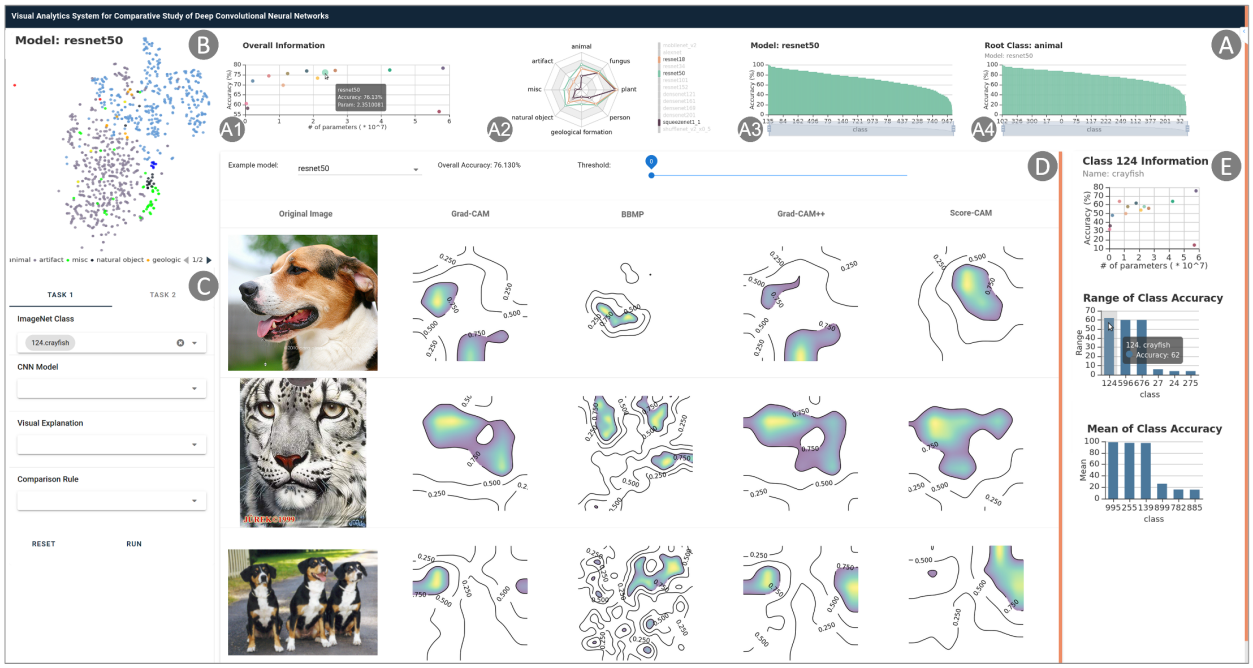


Figure 2.2: The interface of our visual analytics system. **(A) Overall Information View** visualizes multiple quantitative matrices of each CNN model. **(B) Distribution Graph View** shows the aggregating activations of 1000 ImageNet classes generated from the distance matrix of each model's predictions. **(C) Task Selection Sidebar** facilitates users to customize multiple selections to interactively compare the models' behaviors. **(D) Visual Explanation View** provides two types of information - before running any task, it presents the visual explanation examples; when users are running a task, it shows the explanation results. **(E) Supplemental View** works as a supplementary information board.

Phase 2 provides task customization to support flexible study options, ImageNet classes, visual explanation method, and comparison rule. Based on the customized comparison requirements, **Phase 3** presents coordinated visualizations and qualitative information for multi-model comparison or single-model investigation, respectively. We will connect our discussion about the methodology in this section and the interface design in the following section with these phases.

Distribution Graph Generation. We provide a comprehensive and novice-friendly information overview for our users to understand the model's high-level performances. One way to achieve this is to investigate the class distribution which is generated from the model's prediction and reflects how the model interprets the data.

Algorithm 1 Constructing Distance Matrix of N ImageNet Classes

Input:

The image class list of all images in the dataset, *imgClasses*;
The confidence matrix of the model, $confMat$;

Output:

The distance matrix of N image classes, $distMat$;
$distMat \leftarrow zeros((N,N))$
$distMatCount \leftarrow zeros((N,N))$
// iterate through all images
for $imgIdx$ **in** $range(M)$ **do**
    $curClass \leftarrow imgClasses[imgIdx]$ // get ground-truth class
    $P \leftarrow confMat[imgIdx, :]$ // get confidence score vector
    // iterate through all classes
    for $compClass$ **in** $range(N)$ **do**
        $confScore \leftarrow P[compClass]$
        $distMatCount[curClass, compClass] \mathrel{+}= 1$
        // calculate distance
        if $compClass \neq curClass$ **then**
            $distMat[curClass, compClass] \mathrel{+}= (1 - confScore)$
$distMat \leftarrow divide(distMat, distMatCount)$ // get the average
$distMat \leftarrow [distMat + transpose(distMat)]/2$

To visualize the distribution of image classes in respect to a specific model, we create a distribution graph (Fig. 2.2 (B) based on each model's predictions. In Algorithm 1, we use a confidence matrix as the baseline to generate this graph, since it reflects how a model understands the input data. For a given model, every input image is classified according to a confidence vector of size $(1,N)$, where N is the number of image classes. The confidence vector is generated from the Softmax function and presents the model's prediction of the input. By concatenating the confidence vectors of all input images, we get the model's confidence matrix of size (M,N), which includes the model's predictions of the entire input dataset of size M.

Based on the confidence matrix, the distance matrix $distMat$ of the N image classes is generated as described in Algorithm 1. Firstly, the distance matrix $distMat$ of the N image classes is initialized as a zero matrix of size (N,N). Then we assign the class of each input image to *curClass*

and each image's confidence matrix of size $(1,N)$ to P (line 4-5). After that, we iterate over the *distMat* and update the value through the iteration among P (line 8-14). Then, we calculate the *distMat* using the iteration results and the iteration counts *distMatCount* (line 16-17). Finally, we apply dimensionality reduction to the resulting matrix *distMat* using t-SNE [149] to generate the 2D projection matrix for the distribution graph.

The distribution graph presents the distribution of the N classes with respect to the predictions of each model. Since the ImageNet structure is based on the WordNet hierarchy, there are eight root classes representing how human beings classify the $N = 1000$ image classes in the ImageNet dataset. In the distribution graph of our system, each root class is represented by a specific color, allowing users to compare the model's classification and human's classification easily.

Visual Explanation Methods. Visual explanation methods, especially the class-discriminative ones, can help novice ML practitioners to understand CNN model behaviors. Because they can highlight specific regions on the input image that is inferred to contribute the most to the model's decision-making. However, most of the existing visual analytics tools for CNN comparison do not

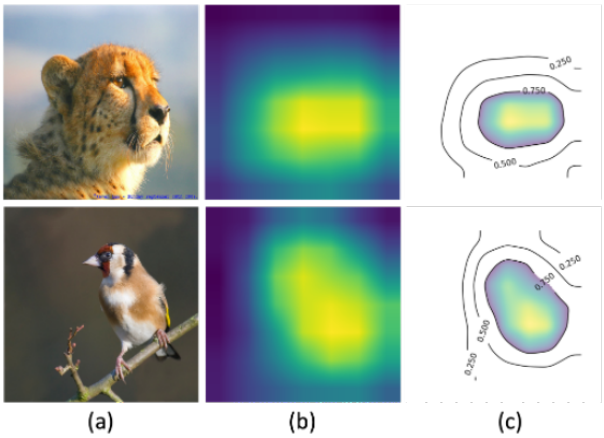


Figure 2.3: Examples of how we improve the presentation of the visual explanation method's results in VAC-CNN. Images at the same row are associate with the same original image. From left to right: **(a)** The original images; **(b)** The conventional visualizations of a visual explanation method; **(c)** The improved visualizations in VAC-CNN.

include any visual explanation methods. To fill this gap, we include five class-discriminative visual explanation methods in VAC-CNN, including Grad-CAM [55], BBMP [120], Grad-CAM++ [56], Smooth Grad-CAM++ [57], and Score-CAM [66]. Examples of these five methods are shown in Fig. 2.2 (D). The reason why these five methods are included is to cover multiple kinds of methods such as gradient-based explanations (Grad-CAM, Grad-CAM++, Smooth Grad-CAM++), perturbation-based explanation (BBMP), and score-based explanation (Score-CAM). Our analytics system is designed to be extensible, so other visual explanation methods can be easily added.

We consolidate the presentation of the visual explanation method's result. As shown in Fig. 2.3 (b), a conventional approach is showing the heatmap, which doesn't provide any direct quantitative information. Thus, the subtle difference among multiple heatmaps can be hard to identify, making it not informative enough for the model comparison task. In VAC-CNN, we add the quantitative information about the visual explanation method's result by overlaying multiple contour lines over the heatmaps [150], which are associated with the attention matrix generated by the visual explanation method (with attention scores of $[0, 1]$, 0 for "no attention"). To support the highlighting of ROI, we also add a customizable threshold for users to remove regions of little attention accordingly. For example, a threshold of 0.5 means the region with attention scores lower than 0.5 will not be highlighted. As shown in Fig. 2.3(c), our improved visualizations of the explanation results incorporate qualitative information and quantitative measures of the attention level, which can support users in model comparison tasks more effectively.

Similarity Matrix Generation. When comparing multiple models based on a single image, users can benefit from a similarity matrix that intuitively shows the correlation of visual explanation methods' results for the CNN models. We demonstrate the method in Algorithm 2.

In this algorithm, the generated saliency map from visual explanation methods are stored as

Algorithm 2 Constructing Similarity Matrix of Selected Models

Input:

The list of visual explanation results of models, $expResults$;
The function of computing similarity scores, $simFunc$;

Output:

The similarity matrix of selected models, $simMatrix$;
$L \leftarrow len(expResults)$
$simMatrix \leftarrow zeros((L, L))$
for $idx1$ **in** $range(len(expResults))$ **do**
    for $idx2$ **in** $range(len(expResults))$ **do**
        // iterate through all visual explanation results
        $expRes1 \leftarrow expResults[idx1]$
        $expRes2 \leftarrow expResults[idx2]$
        $simMatrix[idx1, idx2] \leftarrow simFunc(expRes1, expRes2)$

matrices in a list *expResults*. We provide multiple widely-used image similarity measurements, including the structural similarity index (SSIM), the mean-square error (MSE), the $L1$ measure, and the hash function. The default similarity measurement is set to $L1$ because of its wide acceptance, and other options are provided for users to select a different rule as needed. Based on the user-specified similarity comparison rules, we use the corresponding function *simFunc* to calculate the similarity score of two visual explanation results. After iterating over every element of *expResults*, we can get the similarity matrix *simMatrix* quantifying the similarity of each pair of the visual explanation results. To represent the value intuitively, we use *seaborn* [151] to generate the resulting matrix's heat map. Then, users can interactively compare the behaviors of the selected CNN models through our designed interface.

Image Statistical Analysis. In some circumstances, the conventional visual explanation method may fail to provide enough information to explain the model's prediction. For example, when the prediction result is wrong but the localization is correct, visual explanation methods don't explain why the model made a wrong decision. To solve this problem, we go one step further by analyzing the information generated from the image region that the model cites as essential.

CNN classifiers pre-trained on ImageNet have been proven to rely on texture information rather than the global object shape [152, 153]. However, current algorithms using image texture are often deep-learning-based [154, 155], which can severely interfere our system's response speed. In VAC-CNN, we apply color intensity histograms (CIHs) to measure image information, which are commonly used to analyze the image content and evaluate the image similarity [156, 157, 158]. In this way, the analysis results can be generated in real-time. Our process of image statistical analysis is shown in Fig. 2.4. Based on the model's explanation, which highlights a specific image region essential for the model to make predictions, we can filter the original image by removing the "inessential" part. Then we visualize color intensity information (Fig. 2.4 (C)) to depict the statistical details of the image region that the model cites as essential in making predictions.

As supplementary information of the visual explanation result, the color intensity histogram can help users further analyze what information the model extracts from the input. Through comparing the visual explanation results and the color intensity histograms, users can gain more insights into the underlying behaviors of the deep CNN model.

2.3.2 System Interface

To achieve our design goals described in Sec. 3, we integrate the techniques introduced in Sec. 4 into a web-based visual analytics system, VAC-CNN, for the comparative studies of deep CNN

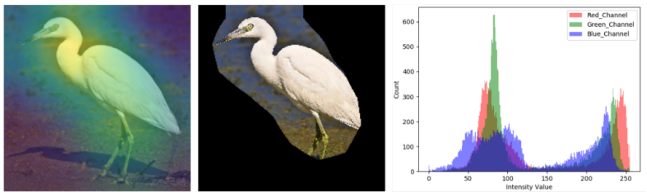


Figure 2.4: Example of the image statistical analysis process. From left to right: **(a)** A visual explanation method highlights ROI on the sample image; **(b)** The filtered image by only keeping the highlighted region of (a); **(c)** The color intensity histogram of the filtered image in (b).

models. As shown in Fig. 2.2, the system interface includes five primary views: "Overall Information View" (A), "Distribution Graph View" (B), "Task Selection Sidebar" (C), "Visual Explanation View" (D), and "Supplemental View" (E). In this section, we illustrate how these views coordinate to facilitate the three phases of comparison workflow described in Sec. 4.1.

2.3.2.1 Information Overview

In order to assist non-experts ML practitioners, VAC-CNN provides an information overview for users to explore high-level CNN model performances and the general behaviors of multiple visual explanation methods. The analysis in this phase requires information from View (A), (B), (D), and (E) in our visual interface.

The Overall Information View (Fig. 2.2 (A)) illustrates the overall and detailed quantitative information of the included CNN models with multiple visualizations. The scatterplot labeled as (A1) indicates each model's complexity and overall accuracy on the entire ImageNet validation set, where each point represents a CNN model. The radar chart labeled as (A2) reveals the accuracy performance of the selected models on the eight root classes. Each line of the radar chart represents one model's performance, and the selectable legend located at the right of the chart enables users to remove uninterested models and only compare selected ones. Additionally, our interactive design allows users to change the pinned model or the pinned root class by a simple click, which can update the two zoomable bar charts shown at (A3) and (A4) of Fig. 2.2, representing leaf classes' accuracy information of the model and root class, respectively, where the leaf classes are ranked in descending order of their accuracies. Thus, each part of the Overall Information View can work synergistically to illustrate each CNN model's quantitative information from multiple aspects, helping users perceive models' performances and achieve efficient high-level multi-model screening accordingly.

The Distribution Graph View (Fig. 2.2 (B)) reveals the distribution of the 1000 ImageNet classes. Each point represents a single image class, and the colors correspond to eight root classes. Generated from each model's confidence score matrix, this visualization presents the model's class-level behavior, enabling users to discover the model's coherent or inconsistent behaviors across clusters of image classes. Besides, by looking at the clusters, users can also discover typical image class groups for further investigation in the following phases, which means this view also serves as a class recommendation. Similarly, smooth user interactions, including hovering over, clicking, zooming, etc., are supported as well.

As discussed in Sec. 4.3, the Visual Explanation View (Fig. 2.2 (D)) presents the example results of multiple visual explanation methods, informing non-expert users how each of the visual explanation method's result looks like.

Finally, the Supplemental View (Fig. 2.2 (E)) provides users with supplemental information. At the information overview phase, two bar charts are presented at this view before users make any ImageNet class selections at the Task Selection Sidebar (Fig. 2.2 (C)). The first bar chart, "Range of Class Accuracy", visualizes the range of the thirteen models' accuracies on six image classes, including image classes on which the models have either diverging or parallel performances. And the second bar chart, "Average of Class Accuracy", includes information related to six image classes, on which the models have coherent good performances or bad performances. These two bar charts illustrate image classes with abnormal statistical characteristics, suggesting interesting images for users to explore in more detail.

2.3.2.2 Task Customization

VAC-CNN also supports users to customize the comparative study with the Task Selection Sidebar (Fig. 2.2 (C)). From this view, users can select multiple CNN model(s), ImageNet class(es), visual

explanation method(s), etc. Based on different selections, multiple subtasks can be performed in the following phase, including comparing multiple models over a particular image class, investigating a single model's behaviors on multiple image classes, and explaining a single model's behavior on images within a particular class. For the multi-model comparison task, VAC-CNN supports the users to select up to 13 models for comparison.

2.3.2.3 Model Investigation & Comparison

In the model investigation & comparison phase, the views (D) and (E) will be updated to present information based on the result from the user-specified comparison task.

In the Visual Explanation View, various information is presented through a table format representation. With multiple rows, this table presents comparison results of up to 13 models selected by users through task customization (as described in Sec. 5.2). Besides, the interaction features allow users to sort on multiple quantitative columns and search specific information to filter the results and get a deeper understanding.

In the Supplemental View, users can find various supplementary information to support model comparison and investigation according to different analysis needs. When users compare multiple models, this view includes information such as the original image selected by users, the similarity matrix of the models' visual explanation results, and the scatterplots presenting the models' accuracies on each selected image class (Fig. 2.5 (2-E)). When users investigate a single model, this view only shows the accuracy scatterplots of each selected image class since most of the essential information is already available in the Visual Explanation View.

2.3.3 Use Cases

In this section, we demonstrate how VAC-CNN can help novice ML practitioners conduct comparative studies with two use cases: (1) comparing the behavior of multiple models on the same

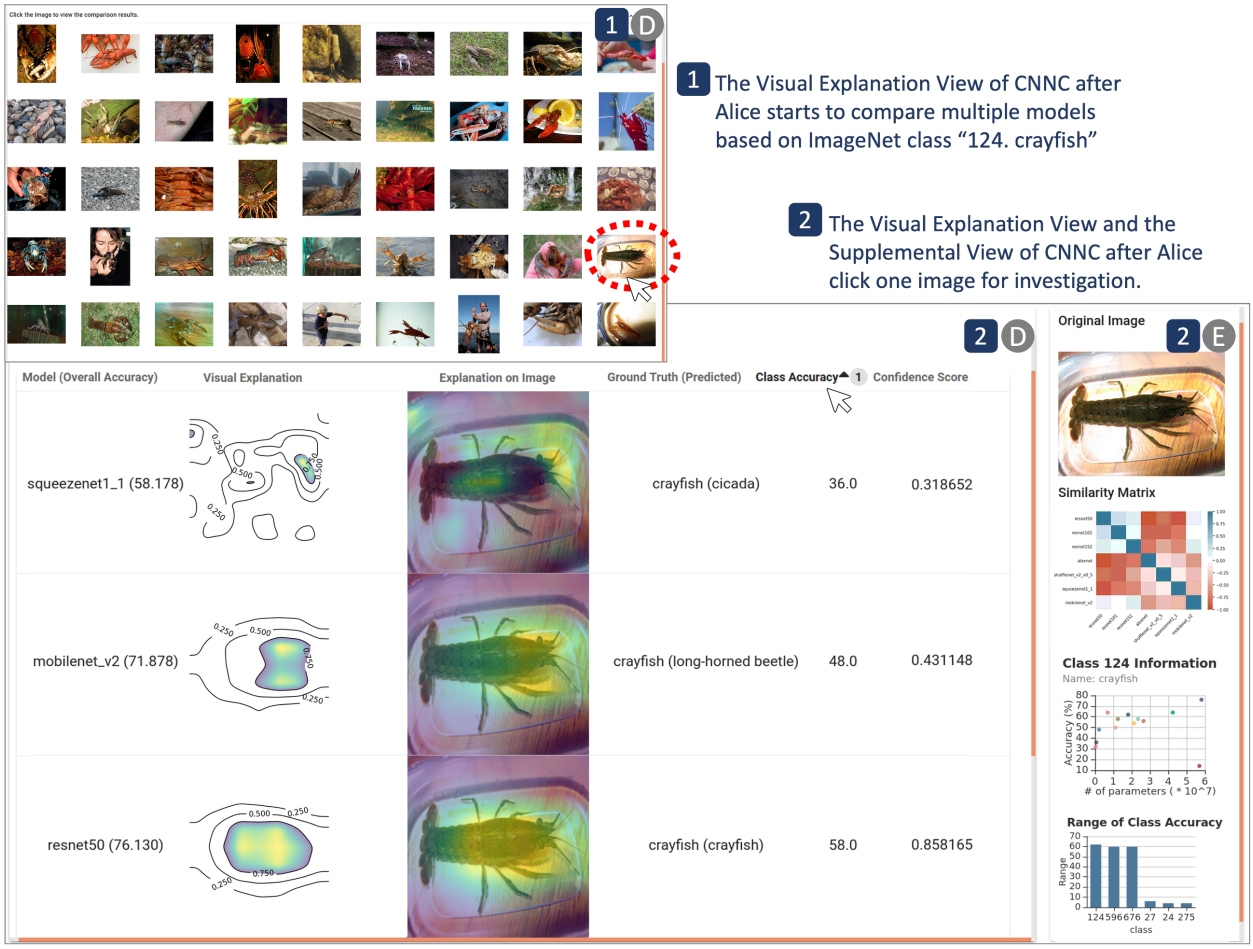


Figure 2.5: Comparing the performance of 7 models on a selected image from class “124 *crayfish*” (Sec. 6.1). In (1-D), we can find a common characteristic of images in this class is their complicated background. After selecting one image of a shrimp, we can find more information with the updated views (2-D) and (2-E) (The views are marked as (D) and (E) to keep consistent with Fig. 2.2). By sorting the result in the table of (2-D) with different metrics, we can check the potential relationship between the model’s prediction performances and the ROI size.

image, (2) investigating a single model’s behavior on different images. The first use case demonstrates how VAC-CNN supports multi-step model comparison, from high-level screening over 13 models to the in-depth interpretable comparison of 7 models. The second use case is about single-model inspection, showing how our provided informative visual explanation assists users.

2.3.3.1 Comparing Multiple Models on Same Image

Alice is a Master’s student majoring in animal behavior study. She gained some basic knowledge about CNN and deep learning from a public course provided by the Computer Science Department

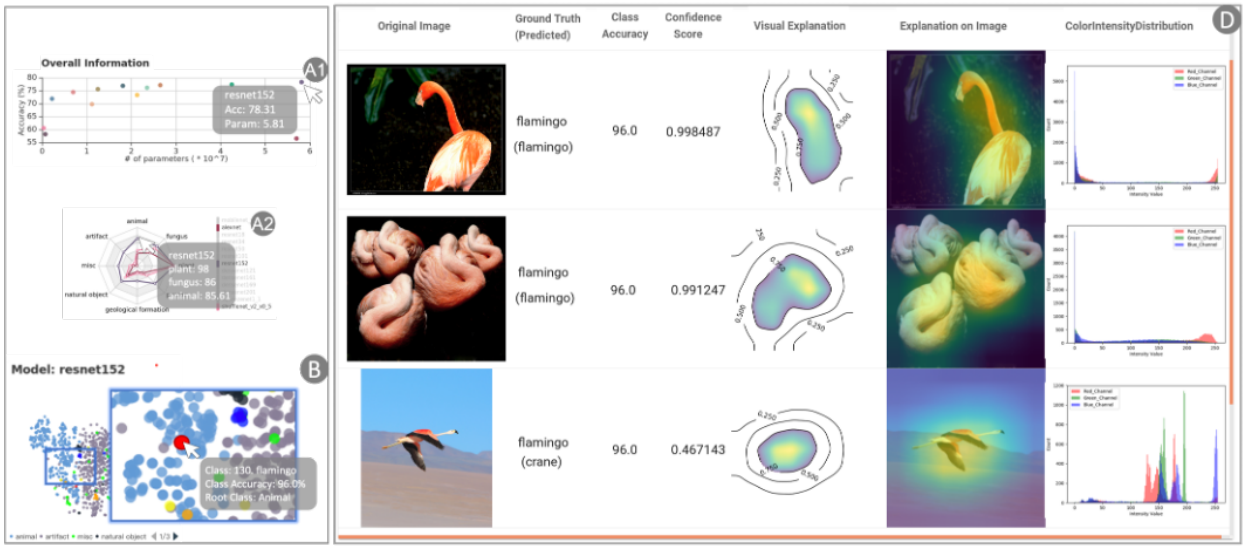


Figure 2.6: Investigating a single model *resnet152* on different images of class "flamingo" (Sec. 6.2). The multiple visualizations in (A1), (A2), and (B) can help users notice the high-level behavior pattern of *resnet152* from different perspectives. And the coordinated view (D) shows the CIH results to help users discover the common patterns among models' failure cases on the "flamingo" class. (The views are marked as (A1), (A2), (B), (D) to keep consistent with Fig. 2.2.)

and wants to apply it to her own major. Therefore, she uses VAC-CNN to explore the performances and behaviors of multiple CNN models on a group of images about animals.

After opening up the system, Alice starts by deciding the models, ImageNet classes, and visual explanation methods for her comparison task. She looks over the different plots in the Overall Information View (Fig. 2.2 (A)) to inspect the performances of the 13 CNNs and becomes interested in ResNet models when she notices the performance boost from *resnet18* to *resnet152*. She also notices that ResNet architectures often have good performance on the "animal" group from the radar chart in this view. Moreover, Alice finds "animal" forms better cluster for *resnet50* from the Distribution Graph View (Fig. 2.2 (B)), so she adds *resnet50*, *resnet101*, *resnet152* into the list of models. Then she looks at the first bar chart in the Supplemental View (Fig. 2.2 (E)) and finds that the models' accuracies vary significantly on class "124 *crayfish*", which belongs to the "animal" group, so she decides to choose this class for model comparison. Finally, Alice explores the Visual

Explanation View (Fig. 2.2 (D)) and notices the ROI provided by "Grad-CAM" is very clear in general, so she decides to use "Grad-CAM" as the visual explanation method.

After having the models, ImageNet classes, and visual explanation methods she wants to select in mind, Alice moves on to the Task Selection Sidebar to customize her comparison task (Fig. 2.2 (C)). When restricting the ImageNet class selection to "124 *crayfish*", Alice notices that a scatter plot in the Supplemental View is updated, as shown in Fig. 2.2 (E). And one model with remarkably bad performance, *alexnet* (14%), stands out. Besides, there are 3 other models whose accuracy is lower than 50%, *shufflenet_v2_x0_5* (32%), *squeezenet1_1* (36%), and *mobilenet_v2* (48%). Curious about the reasons behind those models' failures, Alices also decides to add them to the model list for comparison. In this way, Alice has finalized the objectives of the model comparison task with 7 models:

- Models: *resnet50*, *resnet101*, *resnet152*, *alexnet*, *shufflenet_v2_x0_5*, *squeezenet1_1*, *mobilenet_v2*;
- ImageNet Class: 124 *crayfish*;
- Visual Explanation Method: Grad-CAM.

After finishing all of the customizations, Alice starts the comparison with the Visual Explanation View (Fig. 2.2 (D)) and the Supplemental View (Fig. 2.2 (E)). She first looks over the original images within the selected class (Fig. 2.5 (1-D)) from the Visual Explanation View. She finds that the image background of the main object—"crayfish"—is very complicated for almost every image in this class, which can be a possible cause of the varied model performances. With this hunch, Alice clicks on one image and begins to compare the models' behaviors with the updated Visual Explanation View and Supplemental View (Fig. 2.5 (2)). As shown in Fig. 2.5 (2-D), by sorting the table according to the class accuracy, Alice inspects the visual explanation method's results and the associated numerical information of the 7 models. She notices that, for the 3 models

shown in Fig. 2.5 (2-D), *resnet50* is the only model that correctly classifies the input, while both *squeezenet1_1* and *mobilenet_v2* make incorrect predictions. By inspecting the visual explanation methods' results, Alice realizes that the size of each model's ROI has a positive relationship to the model's prediction correctness: *alexnet* (lowest class accuracy) only highlights a very small region while the ROI of *resnet152* (highest class accuracy) is among the largest ones. After checking more images, Alice confirms the consistency of this observation. Given most of the images in this class have complicated backgrounds, Alice concludes that models with smaller views (i.e. smaller ROI) can't perform very well in this object classification task. From this comparative study, Alice learns that when the images she is dealing with have complicated backgrounds, she will consider selecting CNNs with broader views (e.g., *resnet50*) over others.

2.3.3.2 Investigating a Single Model on Different Images

This use case involves Bob, a first-year Ph.D. student majoring in Computer Science. He is developing a bird recognition App for a course project. And he wants to find the best model for the bird image classification function in his App.

Like Alice, Bob starts by deciding on the model, ImageNet classes, and visual explanation methods for his task. He first checks the models' differences in complexity and overall accuracy using the scatter plot in the Overall Information View (Fig. 2.6 (A1)). He finds that *resnet152* achieves the best performance among the CNN models. And such an advantage is particularly prominent with the root classes "animal" and "fungus" according to the radar chart in the Overall Information View (Fig. 2.6 (A2)). Therefore, Bob chooses *resnet152* to dig deeper into. As shown in Fig. 2.6 (B), then he zooms into the distribution graph of *resnet152* to check the cluster of "bird" species and decides to select class "130 *Flamingo*" to explore model behaviors on it. Finally, after reviewing examples of multiple visual explanation methods in the Visual Explanation View

(Fig. 2.2 (D)), he chooses "Smooth Grad-CAM++" as the visual explanation method for model interpretation. In summary, the objectives of Bob's model comparison task are:

- Model: *resnet152*;
- ImageNet Class: 130 *Flamingo*;
- Visual Explanation Method: Smooth Grad-CAM++.

After customizing his comparison task with the Task Selection Sidebar, Bob starts the model investigation from the updated view (Fig. 2.6 (D)). He first notices that Smooth Grad-CAM++ indicates correct localization of the main object in every image in the class "130 *Flamingo*", even for those incorrectly predicted ones. He feels excited about this discovery and continues to look for the cause of those incorrect predictions made by *resnet152*. He finds that *resnet152* correctly classifies the first two images with high confidence scores but misclassifies the third as "Crane" in Fig. 2.6 (D). In contrast, Bobs thinks the second image is more challenging to recognize than the third one in his eyes. He tries to explain this phenomenon from the color intensity histograms (CIH) provided by VAC-CNN. By comparing the CIHs of the three images, he realizes that the second image's CIH is highly similar to the first one, while the third image's CIH looks much more different from the other two (Fig. 2.6 (D)). After checking the conditions with other image classes of bird species, Bob finds such observation still holds for most failure cases. He shares this interesting discovery with his course instructor. The instructor suggests he construct a small subgroup of the bird classes that most confuses *resnet152*, apply data augmentation specifically, and use it to fine-tune the model. Bob optimizes his model following this idea, and makes his bird recognition App more potent in the classification task.

2.4 Visual Interpretation of CNNs from a Causal Perspective

While the previous section focuses on system-level understanding and comparison of CNNs through interactive visual analytics, understanding individual model decisions requires a more principled interpretation framework. In particular, visual explanations should not only highlight regions correlated with a prediction, but also provide insight into why a model arrives at a specific outcome. From a causal reasoning perspective, this requires reasoning about how hypothesized causes—such as input regions or internal model components—affect prediction outcomes under intervention.

Existing saliency-based explanation methods, especially those built upon Class Activation Maps (CAM) [54] and their gradient-based variants [55, 56, 57], primarily reveal correlations between spatial features and class predictions. Although contribution-based approaches such as Score-CAM [66] and Group-CAM [67] mitigate gradient-related issues, most of these techniques are not explicitly grounded in causal semantics. Explanations that emphasize feature preservation implicitly reflect a notion of *sufficiency*, while often neglecting the complementary notion of *necessity*, which evaluates how changes to hypothesized causes alter outcomes. Causal explanation formalisms based on causal effect [128] and *responsibility* [129] provide theoretical foundations, yet face challenges in generating precise, saliency-style attributions and in simultaneously accounting for both necessity and sufficiency. Prior efforts incorporating causality into CNN explanations, such as CexCNN [130] and related approaches [131], highlight the promise of causal grounding but leave open issues regarding interpretability, precision, and completeness.

Motivated by these limitations, this section presents *SUNY*(**SU**fficiency and **N**ecessit**Y** causal explanation), a framework that models CNN classification as a causal mechanism and quantifies the importance of individual causes from both necessity and sufficiency perspectives. By explic-

itly integrating these complementary views within a unified formulation, SUNY aims to produce explanations that are more faithful to model behavior and more aligned with human reasoning.

2.4.1 CNN Classification as a Causal Mechanism

With a CNN natural image classifier M, an input image I, and a target class c, our goal is to provide saliency maps to highlight which image portions lead to the model's prediction of c from a causal perspective. First, we recall the definition of *causal mechanism* as follows.

Causal mechanism [159]. A portable concept explaining how and why the hypothesized causes, in a given context u, contribute to a particular outcome.

Next, we define the outcome and causes in our problem, then demonstrate how we establish the causal mechanism and formulate our problem based on it.

Outcome: As a class-discriminative explanation, we directly measure the model's prediction probability w.r.t. the target class and set it as the outcome.

Causes: Given that existing CNN causal explanations regard input features or model filters as causes, we allow the flexibility of setting either of them as causes. Specifically, a single cause is (1) a region of the image identified by a feature map; or (2) a model filter in a convolutional layer.

When considering input features as causes, the context u is an unknown process determining the combination of image features. Regarding model filters as causes, the context u is the image region on which we apply the model filters. After establishing the causal mechanism, we can quantify the effect of causes by counterfactually intervening on the hypothesized causes and measuring the outcome, whose results can be used as weights for feature maps' linear combination. Finally, we can produce saliency maps as a causal explanation for the CNN classifier.

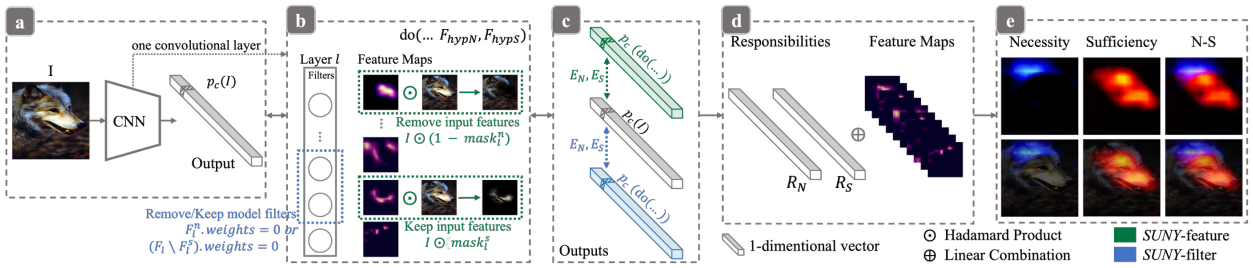


Figure 2.7: Overview of *SUNY* framework. Phase(a) is a forward pass of input image I through a CNN model, where the prediction probability of the target class is $p_c(F)$. Phase(b)-(e) present the generation of *SUNY*-feature (green) and *SUNY*-filter (blue), respectively, referring to different types of hypothesized causes. Note that they are not simultaneous processes. In Phase(b), we obtain filters and feature maps of a specified layer, and intervene on model filters or the corresponding input features. We get new prediction probabilities after the intervention and calculate **N-S Effect**, E_N, E_S in Phase(c), which are fed back to Phase(b) to construct hypothesized cause sets F_{hypN} and F_{hypS}. Through intervening on F_{hypN} and F_{hypS} (Phase(b)), we can obtain E_N, E_S (Phase(c)) and **N-S Responsibilities** R_N and R_S (Phase(d)), which are weights for the linear combination of feature maps. The saliency maps are generated in Phase(e), where we show *SUNY*-feature results as an example.

2.4.2 Method

2.4.2.1 Causal Importance Evaluation

To address the deficiencies of current causal importance evaluations, our evaluation should follow the design requirements below:

R1 Measure the importance of each individual cause in a group of coordinating causes

R2 Quantify the actual effect towards the outcome and pay attention to both N and S.

The Shapley value's framework for assessing marginal contributions across various coalitions lays the groundwork for achieving **R1**. We define the necessity and sufficiency functions to satisfy **R2**. However, previous SHAP [144] image analyses segment input images into equally sized patches, thereby limiting finer distinctions. Our model-integrated method regards a single feature map (or a set of feature maps) as a cause f_i (or a set of causes F_*), thereby providing more granular explanations. We utilize a general formulation of Shapley values in the following definition.

For a set of causes (i.e., a set of feature maps) to be analyzed as F_*, we define $\boldsymbol{N}$ value function

to measure the degree of the outcome change when removing F_*:

$$E_N(F_*) = \frac{p_c(I) - p_c(\text{do}\ (F \setminus F_*))}{p_c(I)}, \tag{2.1}$$

where $\text{do}(F \setminus F_*)$ represents the CF intervention of removing F_*. And $p_c(\cdot)$ refers to the model's prediction probability w.r.t. a target class c, where $p_c(I)$ is its original value without any intervention; $p_c(\text{do}(F \setminus F_*))$ represents the value after the removal intervention. A higher $E_N(F_*)$ indicates a more *necessary* F_*. Similarly, $\boldsymbol{S}$ value function is defined as:

$$E_S(F_*) = \frac{p_c(\text{do}\ (F_*))}{p_c(I)}, \tag{2.2}$$

where $\text{do}(F_*)$ represents the CF intervention of only keeping F_*, i.e., removing $\{F \setminus F_*\}$. A higher $E_S(F_*)$ indicates a more *sufficient* F_*.

To fulfill **R1**, different from covering all elements, we tend to focus on more necessary (sufficient) ones. $\forall f_i \in F$, where f_i is a single cause, we set $F_* = \{f_i\}$ to calculate $E_N(f_i)$ ($E_S(f_i)$) and construct a set $F_N \subseteq F$ ($F_S \subseteq F$) by combining the relatively more necessary (*sufficiency*) f_i. Then to analyze a single cause $f_n \in F_N$, we calculate the N Shapley value as:

$$\begin{aligned} R_N(f_n) = \sum_{F' \subseteq \{F_N \setminus f_n\}} & \frac{|F'|!(|F_N| - |F'| - 1)!}{|F_N|!} \\ & \times [E_N(F' \cup f_n) - E_N(F')]. \end{aligned} \tag{2.3}$$

Similarly, we can calculate S Shapley value for $f_s \in F_S$ as:

$$\begin{aligned} R_S(f_s) = \sum_{F' \subseteq \{F_S \setminus f_s\}} & \frac{|F'|!(|F_S| - |F'| - 1)!}{|F_S|!} \\ & \times [E_S(F' \cup f_s) - E_S(F')]. \end{aligned} \tag{2.4}$$

In the implementation, we reduce the amount of computation by estimating Eqns.(2.3), (2.4) using Shapley sampling values method [142]. Additionally, for $f'_n \in \{F \setminus F_N\}$ and $f'_s \in \{F \setminus F_S\}$, we set $R_N(f'_n) = 0$ and $R_S(f'_s) = 0$.

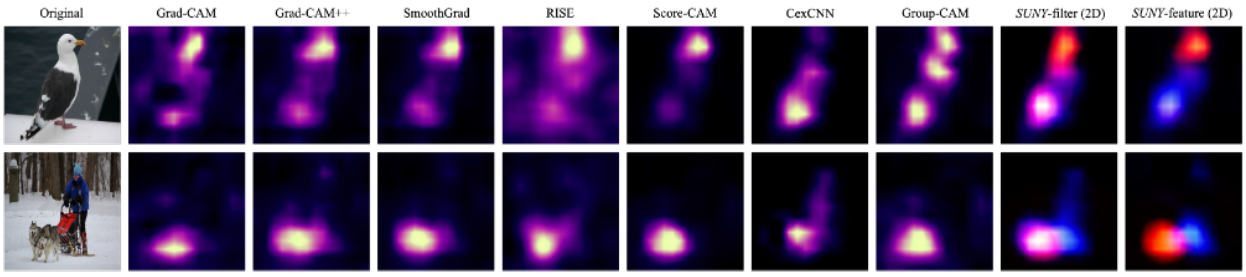


Figure 2.8: Visual comparison of saliency maps from different methods. The first row: a VGG16 trained on CUB-200-2011, and the image is correctly predicted as `Gull`. The second row: a VGG16 trained on ILSVRC2012, and the image is correctly predicted as `Dog-sled`.

2.4.2.2 *SUNY* Implementation

We present the implementation of *SUNY* in Alg. 3 and a visual overview in Fig. 2.7. Following the definitions in Sec. 2.4.2.1, *SUNY* provides causality-driven CNN visual explanations regarding input features or model filters as hypothesized causes, respectively, represented by the cause type E in Alg. 3 with values *"feature"* or *"filter"*. The corresponding explanations are *SUNY*-feature and *SUNY*-filter. We first define $p_c(\cdot)$ as a function to calculate the model's prediction probability w.r.t. a class c for the input denoted by $\cdot$, as shown in line 1 of Alg. 3. Next, in line 3, we construct F_N and F_S, where F_N is the set of hypothetical single causes f_n with relatively higher $E_N(f_n)$. Similarly, F_S contains all hypothetical single causes f_s with higher $E_S(f_s)$. We then calculate **N-S Shapley Values** for every single cause in F_N and F_S following lines 5 - 20. Specifically, for *SUNY*-feature (lines 6 - 11), we upsample and normalize the feature maps, A_l^n and A_l^s, and use the generated *mask* as a feature extractor to intervene on input features from the image I. The intervention $\text{do}(F \setminus F_*)$ (removing F_*) is realized by the Hadamard product, $(I \bigodot (1 - mask))$. Similarly, $\text{do}(F_*)$ (keeping F_*) is implemented as $(I \bigodot mask)$. *SUNY*-filter (lines 13 - 20) removes and keeps the hypothesized causes by filter pruning, which means setting the corresponding filters' weights to zero. Line 14 and line 18 correspond to $\text{do}(F \setminus F_l^n)$ and $\text{do}(F_l^s)$, respectively. After calculating **N-S Shapley Values** for all single causes, in lines 21 and 22, we obtain small saliency

Algorithm 3 *SUNY*: Causal Explanation of CNN

Require: Image I, model M, layer l, class c, cause type E
Ensure: N-S saliency maps: N_{map}, S_{map}

$p_c(\bullet) = \text{Softmax}(M(\bullet))[c]$ ▷ prediction probability on c
$A_l \leftarrow M_l(I)$ ▷ feature maps of the layer l
$F_N, F_S \leftarrow \text{getHypCauses}(I, M, l, E)$
$R_N, R_S \leftarrow zeros(A_l.shape[0])$ ▷ initialize responsibilities
if E is "$feature$" **then**
    for A_l^n **in** F_N **do**
        $mask \leftarrow \text{norm}(\text{upsample}(A_l^n))$
        Compute $R_N(A_l^n)$ based on Eqn.(2.3)
    for A_l^s **in** F_S **do**
        $mask \leftarrow \text{norm}(\text{upsample}(A_l^s))$
        Compute $R_S(A_l^s)$ based on Eqn.(2.4)
else if E is "$filter$" **then**
    for F_l^n **in** F_N **do**
        $M^n \leftarrow \text{pruneFilters}(M, F_l^n)$
        $p_c^n(\bullet) = \text{Softmax}(M^n(\bullet))[c]$
        Compute $R_N(F_l^n)$ based on Eqn.(2.3)
    for F_l^s **in** F_S **do**
        $M^s \leftarrow \text{pruneFilters}(M, (F_l \setminus F_l^s))$
        $p_c^s(\bullet) = \text{Softmax}(M^s(\bullet))[c]$
        Compute $R_S(F_l^s)$ based on Eqn.(2.4)
$N_{map} = \text{norm}(\text{upsample}(\text{Relu}(\sum {R_N}^i A_l^i)))$
$S_{map} = \text{norm}(\text{upsample}(\text{Relu}(\sum {R_S}^i A_l^i)))$
return N_{map}, S_{map}

maps by $Relu(\sum {R_N}^i A_l^i)$, $Relu(\sum {R_S}^i A_l^i)$ and then upsample and normalize them to get the final saliency maps. The operation *norm* represents the min-max normalization w.r.t. each single map, $norm(X) = \frac{X - min(X)}{max(X) - min(X)}$, and $upsample$ represents the bilinear interpolation.

2.4.3 Experiments

2.4.3.1 Experimental Setup

Baseline Methods. The baseline methods we select to compare should be localization approaches. Similar to *SUNY*, they follow two requirements: (1) class-discriminative (different explanations for different specified classes); (2) presenting heatmaps highlighting input regions. Specifically,

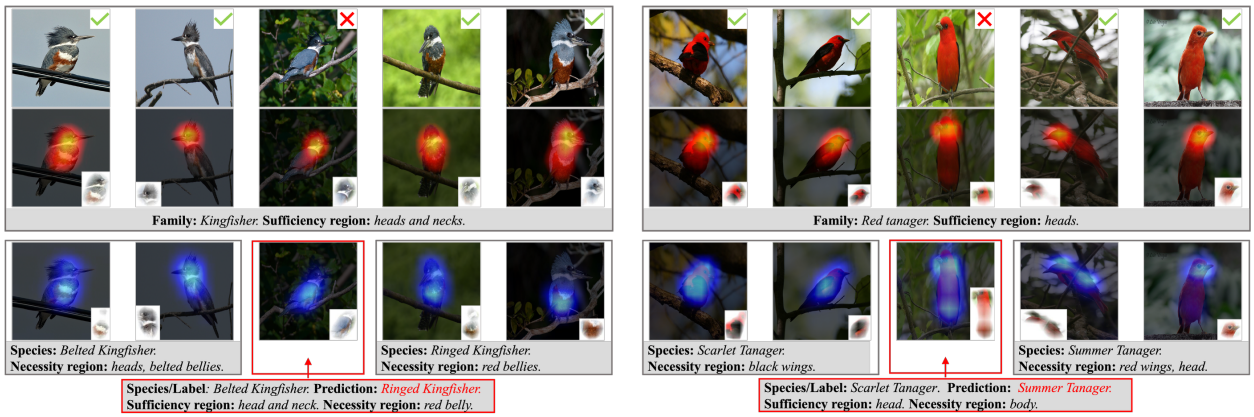


Figure 2.9: Semantic evaluation of *SUNY* explanations for a VGG16 trained on CUB for bird species classification. The bird images in the first row are from four bird species belonging to two families and the correct/incorrect predictions are marked by ✔ and ✖, respectively. For the two images marked by ✖, the model mistakes the actual species with the other species under the same family. Each column corresponds to one image; the second and third rows: *sufficiency* and *necessity* heatmaps. The small image in the bottom corner of each heatmap presents the highlighted image portion.

we compare *SUNY* with seven popular methods proposed between 2017 and 2021, covering approaches that require white-box models or black-box models: Grad-CAM [55], Grad-CAM++ [56], SmoothGrad [65], RISE [64], Score-CAM [66], CexCNN [130], and Group-CAM [67].

Datasets and CNN Models. The experiments involve the validation sets of two commonly-used image datasets, ILSVRC2012 (ILSVRC) [160] with 1000 classes and 50k images and CUB-200-2011 (CUB) [161] with 200 classes and 5794 images. We use all methods to explain three CNNs with different architectures, including VGG16 [162], Inception-v3 [163], and ResNet50 [5].

Implementation Details. We implement *SUNY* and the seven aforementioned visual explanation methods in Python using PyTorch framework [164]. Specifically, we run the official or publicly available code of other methods as the results at the same data scale are unavailable. The platform is equipped with two NVIDIA RTX 3090 GPUs. Unless explicitly stated, the comparisons discussed in this section are conducted following the same settings, including:

(1) Images are converted to the RGB format, resized to 224×224 (VGG16) or 299×299 (Inception-

v3), transformed to tensors, and normalized to the range of $[0, 1]$.

(2) All visual explanation methods' results are generated for both models in the validation sets of both datasets, where we use all images predicted correctly by the models.

(3) Explanations are applied to the last convolutional layer for explaining the predicted class. We bilinearly interpolate the results to the required size of each experiment.

2.4.3.2 Semantic Evaluation

In Fig. 2.8, we visually compare *SUNY* with other saliency map explanations and observe two advantages: (1) Saliency maps provided by *SUNY* contain fewer noises. (2) *SUNY* uniquely provides both *necessary* and *sufficient* information to support interpretation. For example, for the first image, *SUNY* tells that the bottom wing is *necessary* and the head is *sufficient* for `Gull` prediction. For the second image, *SUNY* shows that the sled is *necessary* and the dog is *sufficient* for the prediction of `Dog-sled`. However, other explanations overlook one side of *necessity* or *sufficiency*, and none provide such distinction. More examples can be found in the Supplementary Materials.

Causal explanations with *SUNY*. As shown in Fig. 2.9, we provide images from CUB corresponding to four bird species in two families[1]. By inspecting the *sufficiency* heatmaps in the second row, we can observe that the highlighted regions are similar for those belonging to the same bird family. – All `kingfisher` images share similar heads and belted necks; all `red tanager` images share similar heads. This explains why the model correctly identifies the bird family for every image. From the third row, we find the *necessity* heatmaps provide further explanations through the following observations. For the `kingfisher` family, the belted bellies are highlighted in images predicted as `belted kingfisher`, while the red bellies are highlighted in the images predicted as `ringed kingfisher`. This explains why the third image is mistaken – the red belly

[1]Family is a higher-level taxonomic category than species [165]

is easily observable through this view and is identical to a `ringed kingfisher`. For the `red tanager` family, the black wings are highlighted in images predicted as `Scarlet tanager`, while the red wings or heads are highlighted in the images predicted as `summer tanager`. For the `Scarlet tanager` image being misclassified, the black wing feature is not observable from this view, and the head and red body are captured by the model, which are characteristics belonging to the other species. Therefore, Fig. 2.9 shows that *sufficiency* and *necessity* provides semantically-complementary explanations to better support model behavior interpretation. More examples are provided in the Supplementary Materials.

Table 2.1: Comparative evaluation w.r.t. the *deletion*, *insertion*, and *overall* AUC on ILSVRC and CUB with VGG16, Inception_v3, and ResNet50, where lower *deletion*, higher *insertion*, and higher *overall* indicate a better explanation. The first and second best performances are marked in green and blue, respectively.

Dataset	Methods	VGG 16			Inception_v3			ResNet50		
		Deletion ↓	Insertion ↑	Overall ↑	Deletion ↓	Insertion ↑	Overall ↑	Deletion ↓	Insertion ↑	Overall ↑
ILSVRC	Grad-CAM[55]	0.1098	0.6112	0.5015	0.1276	0.6567	0.5291	0.1796	0.6889	0.5093
	Grad-CAM++[56]	0.1155	0.6033	0.4878	0.1309	0.6476	0.5167	0.1847	0.6799	0.4952
	SmoothGrad [65]	0.1136	0.6023	0.4887	0.1317	0.6465	0.5148	0.1849	0.6800	0.4951
	RISE [64]	0.1185	0.6188	0.5003	0.1404	0.6444	0.5040	0.1303	0.6932	0.5629
	Score-CAM [66]	0.1070	0.6382	0.5312	0.1309	0.6528	0.5219	0.2319	0.6218	0.3898
	CexCNN [130]	0.1161	0.6025	0.4864	0.1355	0.6543	0.5188	0.1886	0.6443	0.4557
	Group-CAM [67]	0.1138	0.6218	0.5080	0.1292	0.6545	0.5253	0.1794	0.6904	0.5110
	SUNY-filter (Ours)	0.1019	0.6389	0.5370	0.1258	0.6570	0.5312	0.1425	0.6957	0.5532
	SUNY-feature (Ours)	0.1005	0.6468	0.5462	0.1215	0.6603	0.5388	0.1323	0.6988	0.5665
CUB	Grad-CAM[55]	0.0558	0.7617	0.7059	0.0963	0.7323	0.6360	0.0930	0.6452	0.5522
	Grad-CAM++[56]	0.0589	0.7541	0.6951	0.0950	0.7281	0.6331	0.0972	0.6407	0.5434
	SmoothGrad [65]	0.0594	0.7489	0.6895	0.0977	0.7244	0.6266	0.0974	0.6405	0.5431
	RISE [64]	0.0560	0.7583	0.7023	0.0855	0.7168	0.6314	0.0570	0.6567	0.5996
	Score-CAM[66]	0.0542	0.7575	0.7033	0.0901	0.7326	0.6424	0.0995	0.6351	0.5355
	CexCNN [130]	0.0630	0.7389	0.6760	0.1017	0.7283	0.6267	0.1014	0.6173	0.5159
	Group-CAM [67]	0.0606	0.7521	0.6915	0.0971	0.7290	0.6318	0.0926	0.6458	0.5532
	SUNY-filter (Ours)	0.0544	0.7556	0.7012	0.0853	0.7333	0.6480	0.0682	0.6613	0.5931
	SUNY-feature (Ours)	0.0518	0.7591	0.7073	0.0842	0.7361	0.6519	0.0562	0.6645	0.6083

2.4.3.3 Necessity and Sufficiency Evaluation

Deletion and Insertion. To evaluate the saliency regions' effects on the model's prediction, we conduct the *deletion and insertion* experiments proposed in [64]. According to the saliency map, the *deletion* metric quantify the model's predicted probability upon the removal of more and more important image pixels, while the complementary *insertion* metric measure the same variable as

the image pixels adds back from the most important to unimportant ones. Consistent with [64, 67], we utilize the Area Under the probability Curve (AUC) to quantify the results, with lower *deletion*, higher *insertion*, and higher *overall(insertion-deletion)* indicative of a better explanation. We follow the same experimental settings as [67], including the blurring method and the step size to remove/add pixels. The comparison results are presented in Table 2.1, where we can observe that except for VGG16 on CUB, both *SUNY*-feature and *SUNY*-filter outperform other approaches in terms of *deletion*, *insertion* and *overall* AUC scores. In the case of VGG16 on CUB, *SUNY*-feature still outperforms most of the other explanations, and *SUNY*-filter achieves competitive results.

N-S Quantification. We introduce the *N-S Quantification* metric to quantify the average *N* and *S* degree of a visual explanation method. The *N-S scores* correspond to one input image I are given by $N_{score} = \frac{p_c(I)-p_c(I\odot(1-map))}{p_c(I)\times Size_{map}}$, $S_{score} = \frac{p_c(I\odot(map))}{p_c(I)\times Size_{map}}$, respectively, where *map* is the saliency map provided by the explanation method, $p_c(\cdot)$ is the model's prediction probability w.r.t. the target class c, and $Size_{map} = \frac{\sum_{i,j}(1_{true}map[i,j]\neq 0)}{map.h\times map.w}$. As the highlighted image regions being removed or kept, N_{score} and S_{score} measure how *necessary* or *sufficient* these regions are for the model's decision, with larger scores indicative of higher *N/S*. For every visual explanation method, we conduct the *N-S Quantification* with VGG16 on the correct-predicted images in the CUB validation set, and

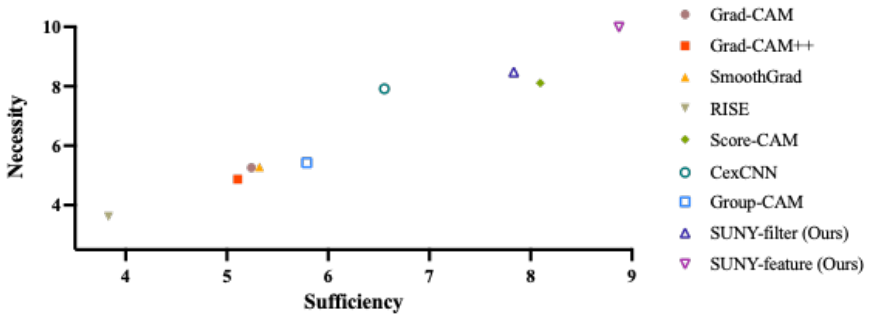


Figure 2.10: Comparison of *SUNY* with seven visual explanation methods in terms of the *N-S Quantification* metric.

calculate the average N_{score} and S_{score} as its *necessary* and *sufficient* level. The result is visualized in Fig. 2.10, which highlights that *SUNY* outperforms most of the seven methods in the ability to identify the most *necessary* and *sufficient* regions corresponding to the model's prediction.

2.4.3.4 Saliency Attack

Researchers have proposed a series of local adversarial attack approaches [166, 167, 168, 169] guided by saliency maps such as CAM [166, 167], Grad-CAM [168], and others [169], which is to fool CNN models by perturbing a small image region highlighted by saliency maps. These methods require the saliency maps to be "minimal and essential" [169], i.e., be capable of capturing image portions that most significantly impact the model output. Inspired by current saliency attack approaches, we propose an evaluation metric, $Attack_{score} = \frac{FlipRate}{AvgAttackSize}$, to validate whether *SUNY* explanations can detect the most important regions w.r.t. the model's decision.

Specifically, we add random Gaussian noise to the saliency regions $I' = (I + noise \bigodot map)$, and then check whether this operation can change the model's decision, i.e., $Flip = 1$ if $argmax(p(I)) \neq argmax(p(I'))$, where $p(\cdot)$ is the model's prediction probability. The $FlipRate$ represents the "essential" level of the saliency regions. To validate whether the region is "minimal", we include *AvgAttackSize*, which is the average size of all saliency maps. For a single saliency map, the calculation of $Size_{map}$ is the same as shown in Sec. 2.4.3.3. Finally, we can obtain the $Attack_{score}$ and the comparison is reported in Table 2.2, which proves that bi-directional explanations are better at highlighting the most important image region corresponding to the model's decision.

2.4.3.5 Localization Evaluation

In this section, we conduct the localization evaluation with the energy-based pointing game [66], a variant of the pointing game [64]. The goal is to measure the localization ability of saliency maps using the ground-truth bounding box of the target class, *bbox*. The input image is binarized

Table 2.2: Comparative evaluation w.r.t. the saliency attack score (higher is better) on the validation sets of ILSVRC and CUB with VGG16, Inception_v3, and ResNet50. The first and second best performances are marked in green and blue, respectively.)

Dataset	Methods	Saliency $Attack_{score}$ ↑		
		VGG16	Inception_v3	ResNet50
ILSVRC	Grad-CAM[55]	0.9615	1.0435	0.7674
	Grad-CAM++[56]	0.9991	0.9821	0.8751
	SmoothGrad[65]	1.0449	0.9675	0.8776
	RISE[64]	0.9928	0.7353	1.0259
	Score-CAM[66]	0.5326	0.9673	0.3378
	CexCNN[130]	1.6341	1.0653	0.6393
	Group-CAM[67]	1.1556	1.0200	0.8020
	SUNY-filter (Ours)	1.6602	1.5725	1.3228
	SUNY-feature (Ours)	2.0452	1.9874	1.5619
CUB	GradCam[55]	0.5969	0.5985	0.4694
	GradCam++[56]	0.6670	0.5950	0.5257
	SmoothGrad[65]	0.7783	0.5929	0.5260
	RISE [64]	0.5063	0.3860	1.1286
	Score-CAM[66]	1.2215	0.5989	0.8027
	CexCNN[130]	1.2673	0.5898	0.4171
	Group-CAM[67]	0.6742	0.5951	0.4991
	SUNY-filter (Ours)	1.3691	0.8867	1.2885
	SUNY-feature (Ours)	2.8111	1.0475	1.7747

Table 2.3: Comparative evaluation w.r.t. the proportion in energy-pointing games (higher is better) on the validation sets of ILSVRC and CUB with VGG16, Inception_v3, and ResNet50. The first and second best performances are marked in green and blue, respectively.

Dataset	Methods	Proportion (%) ↑		
		VGG16	Inception_v3	ResNet50
ILSVRC	Grad-CAM[55]	57.68	66.35	59.84
	Grad-CAM++[56]	61.31	65.93	61.74
	SmoothGrad[65]	62.18	65.78	61.75
	RISE[64]	58.93	59.26	59.48
	Score-CAM[66]	64.25	65.94	66.72
	CexCNN[130]	65.24	66.33	57.39
	Group-CAM[67]	62.70	66.17	60.68
	SUNY-filter (Ours)	65.63	67.02	68.64
	SUNY-feature (Ours)	65.61	66.71	68.02
CUB	Grad-CAM[55]	43.06	40.05	39.02
	Grad-CAM++[56]	45.45	40.45	41.25
	SmoothGrad[65]	47.12	40.34	41.28
	RISE[64]	37.28	34.74	36.32
	Score-CAM[66]	49.68	40.67	47.42
	CexCNN[130]	37.13	41.38	41.22
	Group-CAM[67]	43.53	41.08	40.36
	SUNY-filter (Ours)	51.74	42.03	49.74
	SUNY-feature (Ours)	49.97	41.96	43.21

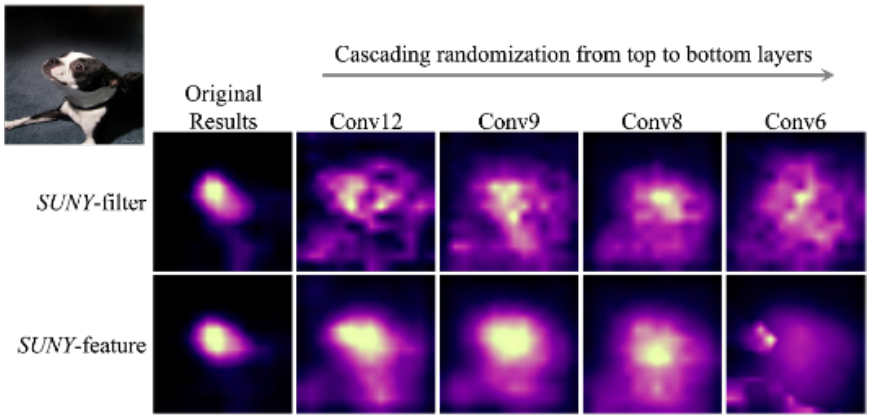


Figure 2.11: Sanity check of *SUNY* explanations for VGG16 by cascading randomization with an example image (shown on the top left) from ILSVRC. Results of *SUNY*-filter and *SUNY*-feature are represented in the first and second row, respectively. The first column is the original saliency maps, and the following columns show results after randomizing specific layers.

with *bbox* by assigning the inside and outside regions with 1 and 0, respectively. Then, we apply the Hadamard product between the binarized input and the saliency map, the summary of which can quantify how much "energy" falls into *bbox*. The performance is measured by $Proportion = \frac{\sum map[i,j]_{(i,j)\in bbox}}{\sum map[i,j]}$. For convenience, we consider all correctly-predicted images with a single *bbox* in the validation set of ILSVRC, CUB, respectively, and report the average proportion value in Table 2.3. The results indicate that both *SUNY*-filter and *SUNY*-feature outperform other methods in terms of localization ability by ranking first and second in the energy-based pointing game.

2.4.3.6 Sanity Check

The sanity check [170] is to validate whether a visual explanation method can be considered as a reliable explanation to reflect the model's behavior. We conduct cascading randomization to the model's weights from the top to the bottom layer successively and generate explanations every time after the randomization. If the saliency maps remain similar across models with widely differing parameters, the method fails the sanity check. Fig. 2.11 presents the sanity check of *SUNY*-filter and *SUNY*-feature, which indicates significant changes for both explanation results after the model

parameter randomization. Therefore, the explanations provided by *SUNY* pass the sanity check.

2.5 Conclusion

This chapter explores how visual intelligence supports the understanding of CNN decision-making from complementary perspectives. Using CNNs as representative deep neural network models in vision, we move beyond aggregate performance metrics to examine model behavior at both system and decision levels.

VAC-CNN provides a visual analytics framework for scalable and flexible comparison of CNN models. By coordinating quantitative measures with multiple visual explanation techniques in an interactive workflow, it enables high-level screening across many architectures as well as detailed inspection of individual models. This system-level perspective facilitates principled reasoning about behavioral similarities and differences in practical multi-model settings.

Complementing this comparative view, SUNY introduces a causality-driven interpretation framework grounded in the notions of *necessity* and *sufficiency*. By modeling CNN classification as a causal mechanism and quantifying feature importance under both perspectives, SUNY produces explanations that are more aligned with human reasoning and more faithful to model behavior.

Together, VAC-CNN and SUNY establish foundational methodologies for understanding model decisions through visual intelligence. This understanding serves as a prerequisite for the subsequent chapters on model and data validation, where reliable and responsible AI systems depend critically on structured insight into model behavior.

Chapter 3

Validating Model Behavior with Human-Aided Analytical Frameworks

Reliable deployment of machine learning models requires systematic validation of model behaviors and efficient mitigation of harmful failure modes. However, identifying problematic data patterns and correcting them often demands substantial metadata annotation, manual inspection, or expensive retraining procedures. This chapter presents human-aided analytical frameworks that support both the discovery and mitigation of model issues. First, we introduce *AttributionScanner*, a visual analytics system for metadata-free data slice finding that helps users identify and interpret subgroups exhibiting systematic model failures. Through interactive visual exploration, the system enables users to uncover common model behaviors and diagnose issues such as spurious correlations or mislabeled data, while facilitating actionable model improvements. Building upon the insights obtained from such analysis, we further introduce *SLIM*, a cost-effective framework for mitigating spurious correlations with minimal human annotation. By leveraging a human-in-the-loop protocol featuring a novel attention labeling mechanism with a constructed attention representation space. *SLIM* significantly reduces the need for exhaustive additional labeling, requiring human input for fewer than 3% of instances. Together, these approaches present a promising path

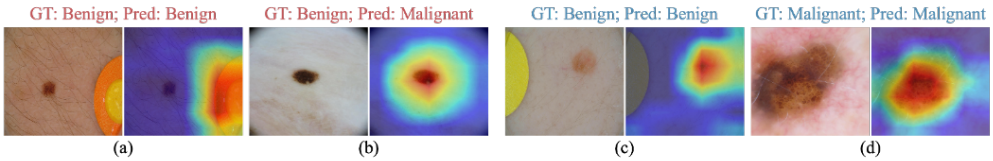
for developing reliable models more efficiently.

3.1 Introduction

Building upon the understanding of model decisions discussed in Chapter 2, the next critical step toward reliable machine learning systems is the validation of model behaviors. As machine learning (ML) technologies are increasingly deployed in real-world applications, detecting model failures and understanding their underlying causes have become essential for improving model accuracy, transparency, and accountability.

A prevalent and harmful issue in ML models is spurious correlation [27], where models rely on irrelevant features rather than task-relevant signals when making predictions. For example, in hair color classification, a model should ideally make decisions according to hair features ("core correlations"), while reliance on unrelated cues such as background or facial attributes constitutes spurious correlations. Such correlations can lead to poor generalization in production environments [171] and raise fairness concerns in AI systems [172]. A notable example is the ISIC skin cancer dataset [173], where a typical spurious correlation occurs when color patches frequently co-occur with benign-labeled data. As shown in Fig. 3.1, a biased model may focus on patches rather than skin lesions to identify melanoma, and fail when such patches are missing, while a desirable case would be the model captures correct skin regions regardless of spurious features' existence. Enhancing model robustness to address such dangerous misdiagnoses is crucial, driving a surge in research towards more effective solutions.

More broadly, ML models often exhibit heterogeneous performance across different subsets of data, making it difficult to comprehensively validate model behavior using aggregate metrics alone [22, 23, 24, 25, 28]. To uncover hidden model failures, data slice finding has emerged as



	Patch	No Patch
Benign	7341	8480
Malignant	0	2011

Figure 3.1: (Left) GradCAM visualizations highlighting model attention on ISIC. (a, b) showcase a model biased towards patches, leading to a correct prediction with wrong reasons in (a) but an incorrect prediction despite correct focus in (b). (c, d) depict a spuriousness-robust model consistently focusing on core features. (Right) The table details the training data distribution by class (benign or malignant) and the presence of color patches, illustrating the imbalance and potential for spurious correlations.

an important paradigm for model validation. A data slice refers to a subgroup of data instances with coherent characteristics that exhibit systematic model behaviors [22, 23, 24, 25, 82, 79]. By identifying slices with unusually poor performance, practitioners can reveal failure modes that would otherwise remain hidden when examining global metrics. However, existing slice discovery methods typically rely on extensive metadata annotations [83, 84, 85] or metadata generated by vision-language models [86]. Collecting such metadata is often expensive and time-consuming, while automatically generated descriptions may be unreliable, particularly in domain-specific scenarios [31]. In addition, many XAI techniques for vision models operate at the instance level [63, 54, 55, 64], producing explanations for individual samples without revealing whether the observed behavior reflects a systematic issue.

Beyond discovering problematic behaviors, addressing the underlying causes of model failures introduces additional challenges. A common strategy is to rebalance the data distribution across different groups. Methods such as GDRO [174] demonstrate that group-balanced training can improve model robustness. However, such approaches require dataset-scale spuriousness labels, which are expensive to obtain. Moreover, human annotations of spurious attributes can be inconsistent in practice, raising concerns about their reliability (see Table 3.9). Other human-in-

Table 3.1: Annotation forms and quantities required by different methods on Waterbirds, highlighting the reduced workload of *SLIM*. (*Att.=Attention, T.=Training set, V.=Validation set*)

Method	Anno. Form	Anno. Amount
RRR[176]	Map	100%× T.
GradIA[177]	Att. Yes-No, Map	100%× T.
Energy Loss[178]	Bounding box	100%× T.
CRAYON[175]	Att. Yes-No	100%× T.
GDRO[174]	Spuriousness label	100%× T.
DFR[25]	Spuriousness label	100%× V.
SLIM (ours)	Att. Yes-No	2.5%×T. or 3%×V.

the-loop approaches require annotators to provide attention correctness feedback or to label important features using maps or bounding boxes [26, 175, 176, 177, 178]. Despite different annotation formats, these methods typically require labeling across the entire dataset, leading to substantial human effort, as summarized in Table 3.1. Meanwhile, approaches that avoid additional annotations, such as JTT [24] and CNC [23], introduce multiple training stages or complex optimization procedures, resulting in increased computational cost.

To address these challenges, this chapter presents two complementary human-aided analytical frameworks for model validation. First, *AttributionScanner* introduces a visual analytics system that supports metadata-free slice discovery for vision models. By leveraging attribution-based representations and interactive visualization, the system enables users to explore high-level model behaviors and identify systematic issues such as spurious correlations or mislabeled data. Through interpretable slice construction and visual summarization (e.g., the *Attribution Mosaic* design), *AttributionScanner* reduces the effort required to analyze large datasets and provides actionable insights for model improvement.

Building upon such validation insights, the chapter further presents *SLIM*, a data-quality-oriented framework for mitigating spurious correlations with minimal human annotation. Instead of requiring dataset-scale spuriousness labels or computationally expensive training procedures, *SLIM* con-

structs an attention representation space that captures locally consistent data features and model attention patterns. By querying attention correctness labels for a small number of representative instances and propagating these labels within the representation space, the framework efficiently curates feature-balanced training subsets that improve model robustness while requiring human supervision for only a small fraction of the dataset.

Together, these approaches demonstrate how structured human involvement can support both the discovery and mitigation of model failures. By combining visual analytics for systematic diagnosis with data-efficient strategies for correcting spurious correlations, this chapter provides practical methodologies for validating and improving machine learning models in real-world deployment scenarios.

3.2 Related Work

3.2.1 Data Slice Finding

Data slice finding is a critical technique for identifying coherent subsets of data where ML models may perform inconsistently. These subsets, or slices, are often defined by similar metadata, model representations, data statistics, or other descriptive information [179, 83, 82, 84]. By analyzing problematic data slices, ML experts can design targeted solutions to address model issues in edge cases, improving the model robustness and reliability [22, 28]. However, existing approaches often rely on expensive metadata to obtain meaningful subgroups, limiting their applicability. To address this, recent work has focused on leveraging feature vectors in the latent space and clustering them to produce data slices [180, 181, 182, 183]. For example, Domino [86] uses a pre-trained vision-language model (VLM) to generate textual descriptions of slices. However, this approach can backfire if there is a significant domain mismatch, yielding less meaningful slices. Our work fills

this gap by introducing a system that provides explainable slices without requiring metadata or pre-trained VLMs, making it more broadly applicable and reliable.

3.2.2 Visual Analytics for Model Validation

Visual analytics has emerged as a powerful tool for both model explanation [50, 184, 185, 186, 187, 188, 189] and validation [190, 191, 192, 193, 194, 195]. For instance, Kahng et al. [196] use pairwise feature combinations to produce data slices and assess model performance. FairVis [197] enables domain users to manually slice data to identify model biases. Similarly, Errudite [198] develops a domain-specific language for slicing textual documents. While manual slicing is useful, it is not scalable due to its labor-intensive nature. To address this, SliceTeller [79] combines an automatic slice-finding algorithm with a visual analytics tool, allowing for iterative refinement of models. However, it still relies on structured metadata to produce meaningful slices. Our approach, *AttributionScanner*, eliminates the need for expensive metadata while incorporating effective human participation, making it a scalable and accessible solution for explainable model validation.

3.2.3 Spuriousness Mitigation

When spuriousness labels are available during training, methods such as class balancing [199] and importance weighting [200] can mitigate spurious correlations by leveraging data groups defined by class and spurious attributes. Following this paradigm, GDRO [174] further improves robustness by focusing training on worst-performing groups. To reduce the reliance on comprehensive spuriousness labeling, several approaches attempt to approximate group information using a reference model [180, 201, 24, 23, 202, 203, 200]. While these techniques alleviate annotation requirements, they often fail to achieve the same performance as methods using explicit spuriousness labels. Other works reduce labeling cost by leveraging spuriousness annotations on smaller

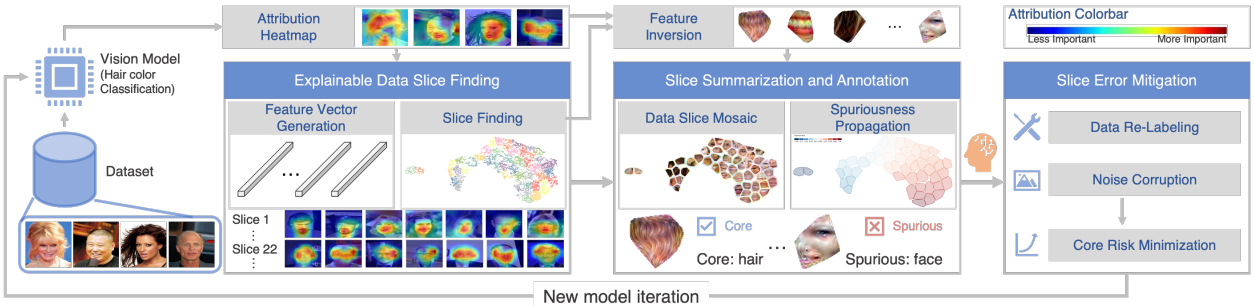


Figure 3.2: *AttributionScanner* workflow involves three phases: *Explainable Data Slice Finding*, *Slice Summarization and Annotation*, and *Slice Error Mitigation*. The first phase: GradCAM is used to assist the generation of feature vectors and then obtain data slices. The second phase: users can identify and annotate slice error types such as core/spurious correlations and noisy labels with the help of *Attribution Mosaic* and Spuriousness propagation. The third phase: the annotation and user-verified Spuriousness are used on the ML side to mitigate slice errors.

validation sets [29, 25]; for example, SSA [29] predicts group labels using semi-supervised learning before applying GDRO. Nevertheless, even reduced-scale labeling remains labor-intensive, and human-defined groups may not align with patterns actually learned by models. Human-in-the-loop approaches therefore incorporate direct human feedback, including attention maps or bounding boxes highlighting core features [176, 204, 178], motivated by concerns about the ambiguity of spuriousness labels [205]. However, these annotations are difficult to produce accurately and typically require labeling across the entire dataset. More recent work simplifies annotation to binary feedback on attention correctness [206, 207], yet still requires feedback for all instances, limiting scalability in practice.

3.3 Discovering Model Issues through Visual Analytics

3.3.1 *AttributionScanner*

In this section, we introduce *AttributionScanner*, a novel approach for interpretable ML model validation with metadata-free, data slice-level analysis. To foster a thorough understanding of our system, we first outline the requirements that shaped our design choices. Then, we delve into the visual components of *AttributionScanner* that enable interpretable model validation. We then

detail our slice computation and summarization method, which allow us to generate data slices with consistent model attribution without additional metadata. Finally, we show how we leverage the insights obtained from our system to enhance the model.

3.3.1.1 Design Requirements

Slice finding has garnered increased attention due to its potential to facilitate a thorough ML model evaluation. However, this task is particularly challenging for vision models, given the inherently unstructured nature of vision data. A pivotal shortcoming we observed in existing approaches is their dependency on metadata, including pre-collected or vision-language model-generated ones, to compute meaningful data slices. However, the acquisition of metadata is resource-intensive, demanding substantial human effort and financial investment. Additionally, the vision-language models are commonly trained on general-purpose datasets, potentially leading to inaccurate or biased results in specific domains. Stemming from these observations, we delineated the subsequent requirements for a system capable of conducting metadata-free, slice-driven model validation:

R1. *Metadata-free*. Our method should yield meaningful data slices without metadata, i.e., neither using metadata as input nor generating or extracting metadata.

R2. *Interpretability*. We should provide sufficient and intuitive information to enhance model transparency, supporting interpretations of a data slice's shared attributes for streamlined exploration and individual instances for insight verification as needed.

R3. *Slice Overview*. Mandatory manual inspection of individual images is impractical for model validation, especially with a high volume of data. To enhance efficiency, our system should provide a visual summary of each data slice, highlighting its key features and attributes.

R4. *Actionable insights*. The system should ensure actionable insights, which requires careful design of visual guidance, issue annotation, and corresponding model or data regularization.

3.3.1.2 System Workflow

AttributionScanner is a human-in-the-loop system, taking as input an image dataset and a trained CNN, such as ResNet [5] for image classification, and yielding interpretable data slices to assist experts in efficient model validation.

A workflow of our approach is presented in Fig. 3.2, where we foster explainable slice finding without metadata (**R1**, **R2**) by leveraging GradCAM [55] and Feature Inversion [58]. The first "explainable data slice finding" phase interpolates model attributions into the latent space to craft attribution-weighted feature vectors, forming an attribution representation space with locally consistent model attributes (**R1**). Then we conduct clustering over this space to generate data slices. Although the produced slices maintain consistent attributions, they still demand significant human effort to identify and understand their shared patterns. The "slice summarization and annotation" phase introduces *Attribution Mosaic*, which visually elucidates each slice's dominant attributions (**R3**), synchronized with instance-level heatmaps for insight verification (**R2**). Upon user annotation of an issue like "spurious correlations", our Spuriousness Propagation method automatically estimates a spuriousness score for each slice to help users uncover other hidden model biases (**R4**). Afterward, our workflow enables the mitigation of detected issues in data or model, corresponding to the "slice error mitigation" phase (**R4**).

3.3.1.3 System Interface

AttributionScanner is comprised of four main components. Fig. 3.3 depicts a use case where our system is used to validate a hair color classification model.

The System Menu (Fig. 3.3 A) provides options for dataset and model selection, visualization layout configuration, and color encoding choice. Two visualization layouts are available: Combination view and Confusion Matrix view, allowing for an overall view of data slices or a more

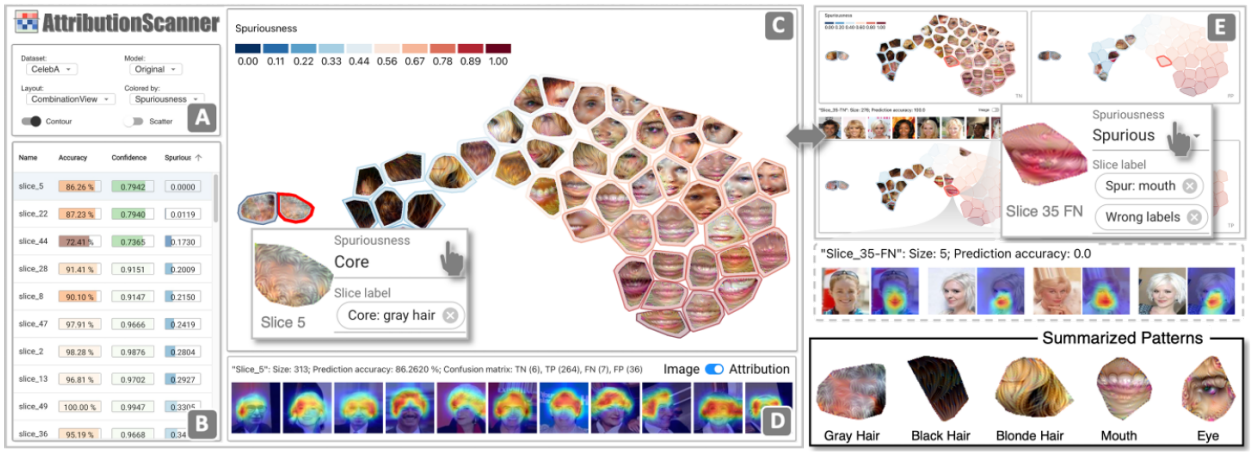


Figure 3.3: *AttributionScanner* applied to the model validation of a hair color classifier trained on the CelebA dataset. (A) System Menu, enabling the selection of dataset, model, and visualization options (Layout, Confusion Matrix, Scatter Plot/*Attribution Mosaic*, and Contour Visibility). (B) Slice Table, showing slice metrics. (C) *Attribution Mosaic*, showing a visual overview of all data slices, which can also be displayed as a confusion matrix view (E). (D) Slice Detail View, showing individual images or Attribution Heatmaps belonging to a selected data slice.

detailed inspection (**R3**). The provided color encoding choices include Slice Name, Slice Accuracy, Slice Confidence, and Spuriousness Probability, adaptable to user requirements.

The Slice Table (Fig. 3.3 B) presents various slice performance metrics to facilitate dataset navigation. Available metrics include Accuracy, Confidence, and Spuriousness Probability, which is produced by the Spuriousness Propagation method upon user annotation (Sec. 3.3.1.5). This table assists in detecting intriguing patterns, like high accuracy with wrong attributions possibly indicative of model biases. On the other hand, low accuracy with correct attributions indicates the potential existence of mislabeled data (**R3**, **R4**).

Our system's third component is the *Attribution Mosaic*, rendering the dominant visual patterns of each slice with respect to the model's decision (**R2**, **R3**), either in a unified form (Fig. 3.3 C) or segregated by the model's confusion matrix (Fig. 3.3 E). The design method for this view is detailed in Sec. 3.3.1.5. Moreover, *Attribution Mosaic* incorporates visualization of user-specified metrics through the coloration of mosaic boundaries, providing a vital guide in pinpointing prob-

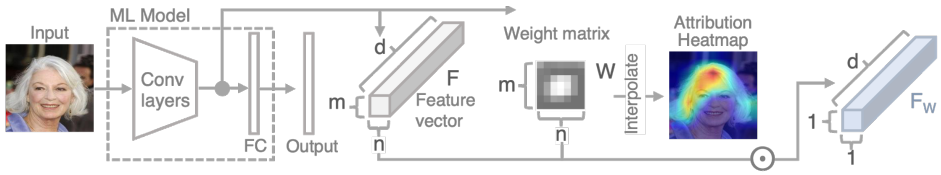


Figure 3.4: Attribution-weighted feature vector generation. An image is forwarded through CNN, where the corresponding feature vector F and weight matrix W can be extracted to calculate the attribution-weighted feature vector F_W.

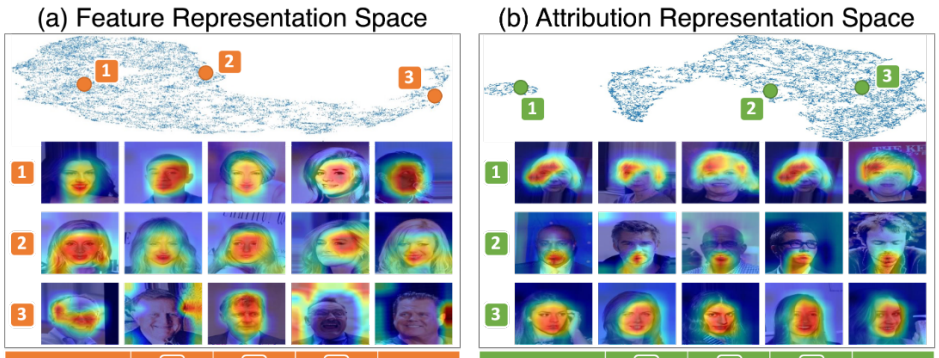
lematic slices (**R3**). Annotation of a slice is facilitated by a double-click action on a mosaic tile, allowing for effortless insight collection (**R4**).

The Slice Detail View (Fig. 3.3 D) showcases individual image samples of the selected data slice, rendering either images or attribution heatmaps. This view enables an in-depth examination of each slice (**R2**).

3.3.1.4 Explainable Data Slice Finding

In this section, we delineate our Explainable Data Slicing method. Our approach presumes a typical CNN model architecture, comprising a sequence of convolutional layers succeeded by a fully connected (FC) layer, a commonality across most CNN models [162, 5, 55].

Attribution-Weighted Feature Vector Generation. In this section, we detail the generation of attribution-weighted feature vectors (**R1**, **R2**). It is notable that there are existing approaches employing attribution-weighted rather than original feature vectors, which aim to provide better model explanation [208, 209, 210] or conduct explainable model enhancement [211, 28]. Considering attribution-weighted feature vectors reserve both data features and model attributes [210, 211, 28], we opt for this design choice to support our explainable data slice construction. Specifically, we leverages GradCAM to obtain model attributes in the latent space, due to its demonstrated efficacy via sanity checks [170] and prevalent usage [211, 182, 212, 213, 214, 30, 175]. In our approach, GradCAM is interchangeable with other XAI techniques such as CAM [54] or RISE [64] (**R2**).



Neighbor Sim	1	2	3	Avg
Data Features	0.9633	0.9748	0.7813	0.9065
Attributions	0.1965	0.2567	0.1132	0.1888

Neighbor Sim	1	2	3	Avg
Data Features	0.9772	0.7549	0.9712	0.9011
Attributions	0.7579	0.8693	1.5706	1.0659

Figure 3.5: Comparison of representation spaces. (a) Feature representation space computed on original feature vectors. (b) Attribution representation space computed on attribution-weighted feature vectors.

As shown in Fig. 3.4, an input image is processed by the model to extract the feature vector F, and GradCAMis performed to produce attributes in the latent space W, which can be linearly interpolated to the image size for attribution heatmap. $W \in \mathbb{R}^{m \times n}$ and is normalized with $\Sigma_{i=1,j=1}^{i=m,j=n} W_{ij} = 1$ to compute the weighted average of F. The resulting attribution-weighted feature vector $F_W \in \mathbb{R}^{1 \times 1 \times d}$ is:

$$F_W = F \odot W = \Sigma_{i=1,j=1}^{i=m,j=n} F_{ij} W_{ij}. \tag{3.1}$$

Data Slice Identification.

Representation Space Construction. To derive meaningful data slices (**R1**, **R2**), we need to construct a representation space with neighbor consistency in terms of model attributes and data features. We first conduct quantitative and qualitative experiments to validate whether attribution-weighted feature vectors help fulfill this requirement—by comparing the feature representation space (derived from the original feature vectors) and the attribution representation space (derived from the attribution-weighted feature vectors). Following the established practices [86, 22, 23] and considering the time efficiency [215, 216], we utilize dimensionality reduction by UMAP [215] to

produce both spaces. At each space, three points are randomly sampled and their nearest neighbors are obtained. We qualitatively examine the consistency of their corresponding attribution heatmaps as shown in Fig. 3.5. Neighboring instances' model attributions on the original space are not similar, while our constructed space has greater local consistency of model attributions.

In addition, we quantitatively compute the neighboring consistency regarding data features and model attributions and report the results in Fig. 3.5. For each point, we compute the average cosine similarities between its top 10 neighbors using their original feature vectors, representing data feature consistency; besides, we compute the average Wasserstein similarities of their attribution mask, where higher values of both scores indicate better consistency. Lastly, we compute the average scores to quantify the overall neighbor consistency of data features and model attributions for each space. As shown in Fig. 3.5, compared to the original feature space, our constructed space has comparable data feature similarity (0.8987) and significantly higher attribution similarity (1.0659), which proves that the attribution-weighted feature vectors help construct a space with satisfactory local consistency in terms of data features and model attributions, providing a solid basis for computing interpretable data slices (**R2**).

Slice Identification. On top of the attribution representation space, we apply K-Means clustering to obtain data slices. We apply over-clustering [86] by incrementing the number of clusters and monitoring the consistency of model attributions within each slice until a coherent grouping of samples is attained. Specifically, the attribution consistency of a data slice is measured as the average cosine similarity of each instance's attribution-weighted feature vector with the slice centroid. We set a threshold at 0.8 to determine slice coherence in our case studies. Note that we allow ML experts to adjust this threshold based on the specific requirements of their datasets and tasks. For rare slices with a small number of data samples featuring abnormal model attributions, this

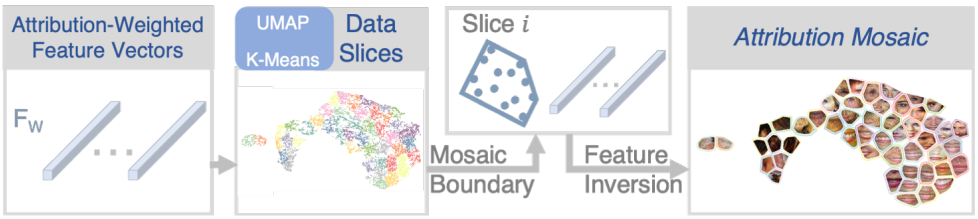


Figure 3.6: *Attribution Mosaic* generation. We compute data slices after acquiring attribution-weighted feature vectors (F_W). Then Feature Inversion is conducted according to F_W and the mosaic boundary of each slice to visualize their common patterns, forming *Attribution Mosaic*.

iterative process, along with our slice mosaic design detailed in Sec. 3.3.1.5, ensures such slices are not overlooked. By combining automated clustering with user-driven refinement, our approach provides a comprehensive and efficient solution for identifying both common and rare slices.

3.3.1.5 Slice Summarization and Annotation

Attribution Mosaic. After obtaining slices with internally consistent model attributions and covering the entire dataset, we design *Attribution Mosaic*, a novel visualization technique, to visually summarize data slices in a mosaic depiction(**R3**). As shown in Fig. 3.6, we first compute the convex hull of each slice based on the projected points in 2D space; then distill each slice's predominant visual patterns via Feature Inversion, integrating the shape constraint of the convex hull. The final landscape featuring slices' attribution summaries is presented as *Attribution Mosaic*, allowing users to easily identify slices with rare, or uncommon model attributions through visual inspection and comparison.

Mosaic Boundary Generation. Given that the convolutional modules take rectangular-shaped input, conventional Feature Inversion produces rectangular visualizations, merely obtaining and displaying its results over the projected cluster centroids leads to overlapped visualizations, hindering effective user exploration. Since K-Means is a centroid-based algorithm that partitions the space into strict non-overlapping Voronoi cells, this guarantees that the convex hulls [217] of our data slices have no overlap. Therefore, we compute the convex hull of each slice on the attribution

Table 3.2: System Design Considerations.

Design Components				System Capabilities			
Slice Table	Slice Detail View	Scatter Plot	Attribution Mosaic	Slice Performance	Slice Attribution	Instance Performance	Instance Attribution
✓	✓			✓			✓
✓	✓	✓		✓		✓	✓
✓	✓		✓	✓	✓	✓	✓

representation space, which provides a mosaic layout for our slice visualization.

Mosaic Drawing with Feature Inversion. We deploy Feature Inversion technique [218, 219, 220] to visualize the significant pattern in each slice (**R3**), which is an optimization-based technique to visualize shared features across a group of feature vectors. Specifically, we first create an image I with size $W \times H$, which is the minimal rectangle encompassing the mosaic boundary of a specific slice denoted by M. The optimization process focuses on updating pixels in I that lie inside M, synthesizing a mosaic-shaped image to visualize the dominant pattern of the feature vectors. To enhance the quality of synthesized images, we initialize them with the average of the original images instead of Gaussian noises [219] and parameterize them in the Fourier space [220], making them better aligned with natural images rather than less-interpretable artificial patterns. The optimization process is defined by:

$$x^* = \arg\max_{x \in M} \left(\ell(\phi(x), \phi_0) + \lambda R(x) \right), \tag{3.2}$$

where x^* denotes the pixels in I bounded by the mosaic shape M, the convex hull of the slice. The values of x^* are updated iteratively through the optimization process to maximize the target function. $\phi : \mathbb{R}^{W \times H \times C} \to \mathbb{R}^d$ is the representation function, i.e., the model's convolutional modules. ϕ_0 is a target feature activation, which is the average of attribution-weighted feature vectors of a data slice in our case. $\ell(*)$ is the loss function where the Euclidean Loss is used here, and $R(x)$ is a Total Variation (TV) regularization term to improve the quality of results [208]. Thus, we obtain

a mosaic-shaped image to visualize the shared attribution patterns within a data slice. Lastly, the visualizations are exhibited at their respective positions within the *Attribution Mosaic*.

Slice Annotation and Spuriousness Propagation. The *Attribution Mosaic* facilitates rapid comprehension of the main content of different slices, supporting the identification of potential spurious correlation issues(**R4**).

We introduce a metric termed Spuriousness probabilities to streamline the detection of spuriousness, ranging from 0 to 1, indicative of the likelihood of spurious correlation existence. We employ the Label Spreading algorithm [221] to propagate user-annotated spurious/core information to other slices. Specifically, a graph is constructed where nodes represent data points, and edge weights encode similarities between points. In our case, we fit the graph to our 2D attribution representation space. Based on this graph structure, the Label Spreading algorithm iteratively updates the labels of unannotated slices, ensuring local and global label consistency: similar slices should exhibit similar spuriousness scores, and the propagated scores should respect the annotated slice spurious/core labels. While only one annotation of "core" or "spurious" is required to initialize the process, the spuriousness scores are dynamically updated as users provide additional annotations. The iteratively refined scores are visualized in the interface to guide further annotation.

The Spuriousness probabilities obtained through the *Attribution Mosaic* offer two benefits. First, they streamline slice exploration by highlighting potential spurious correlations, supporting identifying and assessing problematic slices. Second, after users' verification, these probabilities inform the model-side strategies for issue mitigation (Sec. 3.3.1.7) (**R4**). Since propagated scores are visualized alongside the *Attribution Mosaic*, users can easily validate and override incorrect propagations, ensuring human decisions take precedence over automated computations.

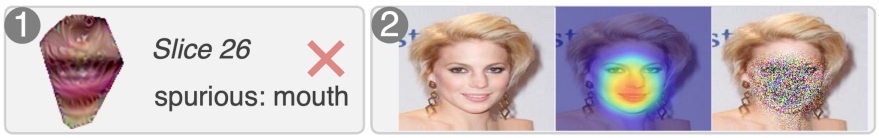


Figure 3.7: Example of spurious correlation mitigation with CoRM for hair color classification: 1 The *Attribution Mosaic* indicates a problem with a spurious feature: mouth. 2 An example of how noise is occupied by CoRM: Original Image (left), GradCAM activations (middle), noise added to the spurious regions of the image (right).

3.3.1.6 System Design Considerations

In designing *AttributionScanner*, various configurations of visual components' combinations have been evaluated to ensure optimal system capabilities to fulfill our design requirements discussed in Sec. 3.3.1.1, as outlined in Table 3.2. Our system should be capable of providing four-faced information: (1) slice performance indicator (**R3**), (2) slice attribution overview (**R2**, **R3**), (3) instance performance indicator (**R2**), and (4) instance attribution inspection (**R2**, **R4**). The notions in "design components" are consistent with Fig. 3.3.

In Table 3.2, the initial configuration, "Slice Table & Slice Detail View," is capable of presenting slice performance indicators and instance attributions. However, the numerical matrices of performance indicators are not sufficient in guiding users to uncover hidden model issues, such as spurious correlations that can happen regardless of high or low performance. A subsequent iteration included a scatter plot showing the attribution representation space. This configuration enables the system's instance performance indicator capability but still misses the slice attribution overview. To the best of our knowledge, there exist few approaches that visually elucidate potential model spuriousness in scale, where our third configuration fulfills this gap — "Slice Table & Slice Detail View & *Attribution Mosaic*, " which comprehensively satisfies all the essential system capabilities and fulfills our design requirements.

Attribution Mosaic, alongside the Slice Table and Slice Detail View, enriched the system's

ability to provide granular insight into both slice and instance-level attributions and performance indicators (**R2**, **R3**). The coordinated Slice Table and *Attribution Mosaic* can guide users to pinpoint hidden issues, and the Slice Detail View enables users to further verify their insights (**R4**). This robust configuration underscores our meticulous design process in assuring that *Attribution-Scanner* is not only capable of furnishing critical insights but also does so in an intuitive and user-friendly manner.

The tabulated design considerations and the ensuing choice of system component configuration aim at maximizing the system's utility and user-centric functionality. With our finalized configuration (the last row of Table 3.2), users are empowered with sufficient support for problem detection, interpretation, and mitigation in the model validation.

3.3.1.7 Slice Error Mitigation for Model Enhancement

One effective approach to mitigating spurious correlations in ML models is through model retraining, which can improve the model's robustness without changing its architecture. We leverage the Core Risk Minimization (CoRM) method introduced in [171]. CoRM corrupts non-core image regions with random Gaussian noise and retrains the model using the noise-corrupted data, which has been shown to be effective in mitigating a model's reliance on spurious features.

We first export slices with high Spuriousness probabilities, indicating undesired correlations. For the images in these slices, the model attribution masks (e.g., GradCAM masks) highlight spurious regions and we utilize such masks to add random Gaussian noise to spurious regions. For a single image, this process can be represented by $\mathbf{x}' = \mathbf{x} + \mathbf{m} \bigodot \mathbf{z}$, where $\mathbf{x}$ is the input image, $\mathbf{m}$ is the attribution mask, and $\mathbf{z}$ is the generated Gaussian noise matrix. All these three variables are of the same size as the input image, and $\bigodot$ denotes the Hadammard product. Fig. 3.7 shows some examples of this operation, with exaggerated noise for presentation purposes. After replacing the

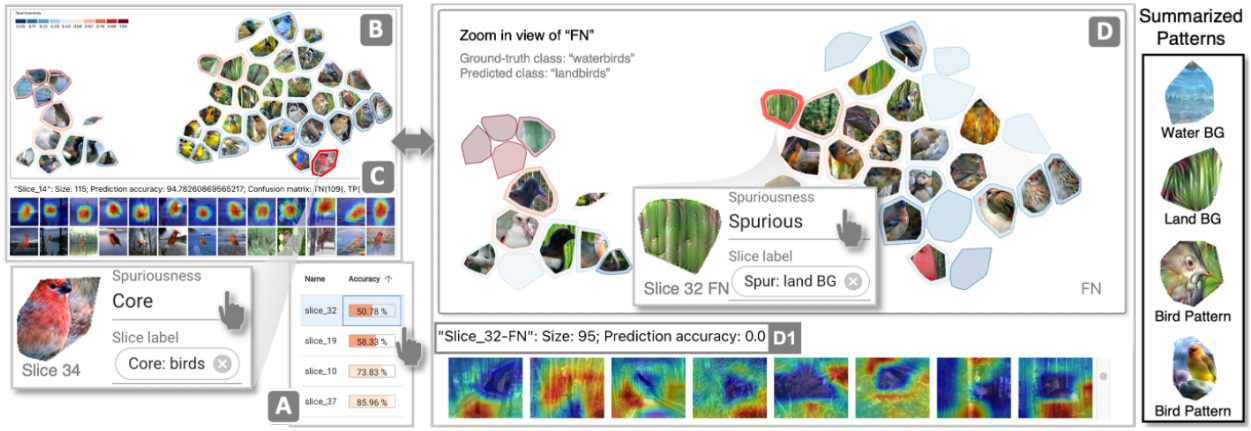


Figure 3.8: *AttributionScanner* applied to the validation of a bird category classification model trained on the Waterbirds dataset. The user finds the correct model behavior (bird patterns) corresponding to a well-performed *Slice 34* and annotates it as "core: birds" B and C. By switching the *Attribution Mosaic* to the confusion matrix view D and investigating the underperformed slices with accuracy sorting A, the user identifies a problematic *Slice 32* that has high false negatives D1, which turns out to use spurious feature of land backgrounds (BG) to predict landbirds.

original data with these noisy-corrupted slices, we retrain the model and evaluate whether spurious correlation has been reduced (**R4**).

3.3.2 Case Studies

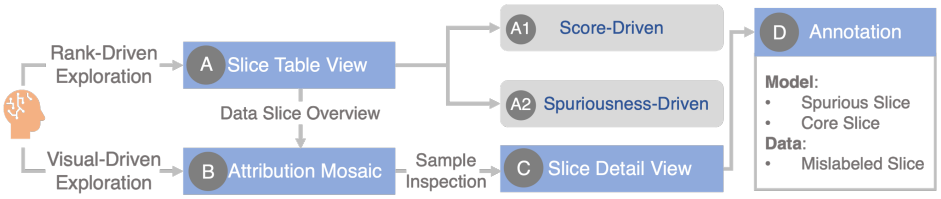


Figure 3.9: Usage patterns for *AttributionScanner* derived from case studies. The analysis can start with A a rank-driven exploration (ranking the slices by A1 score (e.g., model accuracy) or A2 spuriousness), or B a visual-driven exploration. After inspecting slice summary in *Attribution Mosaic* (B), they can inspect individual samples in the Slice Detail View (C). Finally, they can annotate their insights (D).

In this section, we present two case studies with publicly available vision datasets. The primary objective of these case studies is to demonstrate how *AttributionScanner* empowers ML experts and practitioners to detect, evaluate, and interpret potential issues in vision models. Fig. 3.9 illustrates common usage patterns of our system: users can begin the analysis with either a rank-driven

evaluation Ⓐ or a visual-driven evaluation Ⓑ. Once a slice of interest is identified, users can delve deeper into individual samples in the Slice Detail View Ⓒ. Finally, the user can annotate the data slices and continue their analysis Ⓓ.

3.3.2.1 Hair Color Classifier Validation - Finding Edge Cases

• **Overview.** This case study involves the Large-scale CelebFaces Attributes (CelebA) dataset [222] with $202,599$ number of face images. The label of each image is one of {*not gray hair*, *gray hair*}, refer to labels $\{0, 1\}$, respectively. With the train, validation, and test splits of $(8 : 1 : 1)$, we adopt transfer learning to train a ResNet50 [5] binary image classifier. After iteratively fine-tuning hyper-parameters, we obtain a model with 98.03% classification accuracy. The ML experts would then explain and troubleshoot this hair color classifier with *AttributionScanner* based on these settings: *n_neighbors* $= 5$, *min_dist* $= 0.01$, and *n_components* $= 2$ for the UMAP algorithm, and *n_clusters* $= 50$ for K-Means.

• **Does the model behave correctly on well-performing slices?** One basic expected behavior for a hair color classification model is to catch hair features. At first glance of the *Attribution Mosaic* (Fig. 3.3 Ⓒ), ML experts notice *slice_22* and *slice_5* on the left, which lie separately with others on the *Attribution Mosaic*. Their mosaic visualizations indicate gray hair patterns, which means the model uses the correct features, i.e., core features. By verifying the corresponding attribution heatmaps (Fig. 3.3 Ⓓ), they confirm the correctness of this insight and annotate both slices as *core feature* with description *Core: gray hair*. Upon experts saving the annotation, *AttributionScanner* automatically propagates the annotation and provides a Spuriousness probability of each slice. With this guidance, the experts observe many slices lying on the right of *Attribution Mosaic* have higher Spuriousness probabilities (Fig. 3.3 Ⓒ Ⓔ). Their mosaic visualizations do not show any hair patterns, leading to valid doubts that the model does not behave correctly on such slices. Be-

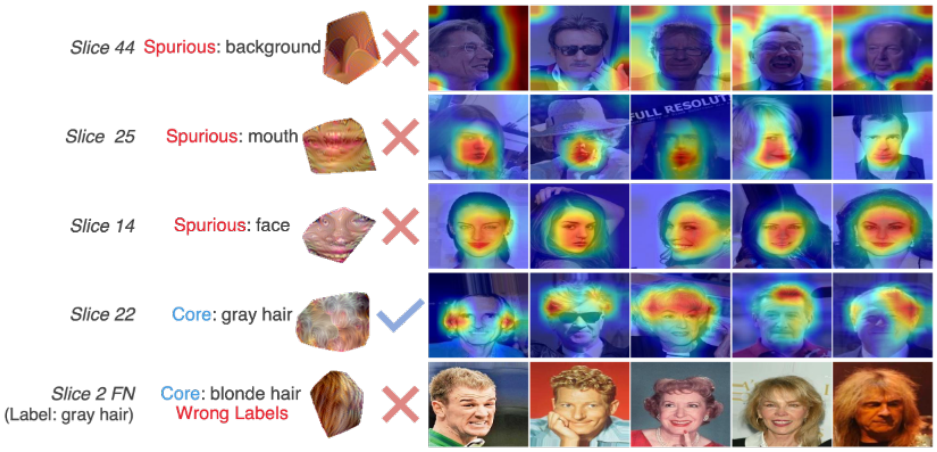


Figure 3.10: Examples of findings when ML experts validate a hair color classifier, where experts discover slices corresponding to the model's core/spurious correlations and wrong labels.

sides, they also notice that the model has correct predictions on these slices (with 100% prediction accuracy). This makes them worry that the model is largely biased by spurious features. Through investigation, they find the model mistakenly utilizes mouth and eyes to predict hair color for those top-performed slices, such as *slice_25*, *slice_14*, and *slice_2* (Fig. 3.10).

• **What underlying factors contribute to unexpected behaviors?** By sorting the Slice Table by descending order of *Spuriousness*, ML experts notice more slices with high Spuriousness, such as *slice_35*, and are interested in understanding why such unexpected model behaviors happen. They switch *Attribution Mosaic* into the confusion matrix form (Fig. 3.3 E) and click on the name of "slice_35" to highlight this slice on the four sub-views and inspect explanations. They notice that images that lie in the "FN" group of this slice have wrong labels — they should be labeled as "not gray hair" rather than "gray hair". They mark this issue as wrong labels and investigate its neighborhood slices on *Attribution Mosaic*, finding *slice_2_FN* also features wrong labels.

• **Why slices underperform?** After fully exploring those well-performed but biased slices, the experts wonder whether the underperformed slices are indeed issue-free or what contributes to their low performance. By sorting the Slice Table (Fig. 3.3 B) by ascending order of accuracy, they

notice the model only achieves 72.41% accuracy on *slice_44* and find that the slice's visualization only shows colorful patterns without any recognizable content. After checking the attribution heatmaps, they find that the model mistakenly uses image backgrounds to make hair color predictions (Fig. 3.10). This spurious correlation issue stands out, and they annotate this slice as "spurious" with the description "Spurious: backgrounds". Similar issues also occur in its neighborhood slices via *Attribution Mosaic*, and *AttributionScanner* automatically propagates them with higher Spuriousness.

• **Insights summary.** Fig. 3.10 presents examples of the detected issues with the current model, which include wrong attributions in underperforming slices, model biases in well-performing slices, and noisy label issues. Through the aforementioned procedure in this case study, we demonstrate how *AttributionScanner* supports users to uncover and interpret potential model issues with visual summaries and other guidance. Based on annotations from ML experts, we employ the CoRM framework to mitigate the detected errors.

3.3.2.2 Bird Category Classifier Validation - Detecting Bias

• **Overview.** To study whether *AttributionScanner* can help ML experts and practitioners find the potential biases and discrimination of models, we design this case study with a biased dataset called Waterbirds [22], which is constructed by cropping out birds from images in the Caltech-UCSD Birds-200-2011 (CUB) dataset [223] and transferring them onto backgrounds from the Places dataset [224]. For each image, the label belongs to one of {*waterbird*, *landbird*}, and the image background belongs to one of {*water background*, *land background*}. The training set is skewed by placing 95% of waterbirds (landbirds) against a water (land) background and the remaining 5% against a land (water) background. Following the same data splitting as [22], the train, validation, and test sets include 4795, 1199, and 5794 images, respectively. After training

and fine-tuning hyper-parameters, the waterbirds/landbirds classification model achieves 85.74% classification accuracy. In the data slice finding, we set *n_neighbors* = 20, *min_dist* = 0.05, and *n_components* = 2 for the UMAP algorithm, and *n_clusters* = 46 for K-Means.

While in this study, ML experts are aware that the model is likely biased by backgrounds — using water (land) background to classify waterbirds (landbirds). However, such priori knowledge is hard to establish in real-world applications because of the scarcity of additional well-labeled metadata. And hence the experts assume such information is unknown and want to validate whether *AttributionScanner* can make the potential model biases stand out by only utilizing the original images and the trained model.

• **Does the model exhibit bias?** To answer this key question, experts start by investigating the underperformed slices. From the Slice Table (Fig. 3.8 A), they sort slices by ascending order of accuracy and select the worst-performed *slice_38*. The coordinated information provided by *Attribution Mosaic* and model attribution heatmaps highlights a spurious correlation problem—the model uses water backgrounds to classify birds (Fig. 3.11). The experts annotate this slice as "spurious", and *AttributionScanner* automatically propagates this annotation. They verify the propagation correctness on neighborhood slices and annotate *slice_41* as "spurious" with "water backgrounds" (Fig. 3.11). Moreover, the experts identify an underperformed *slice_32* that is not clustered with the current ones and is given a high Spuriousness possibility. They investigate it and verify that the model utilizes "Spurious feature: land backgrounds", to predict bird classes. Through the *Attribution Mosaic* in the confusion matrix form (Fig. 3.8 D), they find such spurious correlations result in many false negatives (Fig. 3.8 D1), where the model uses land backgrounds to mistakenly predict many "waterbirds" as "landbirds".

• **Is the detected bias prevalent across all slices? Why or why not?** ML experts are interested

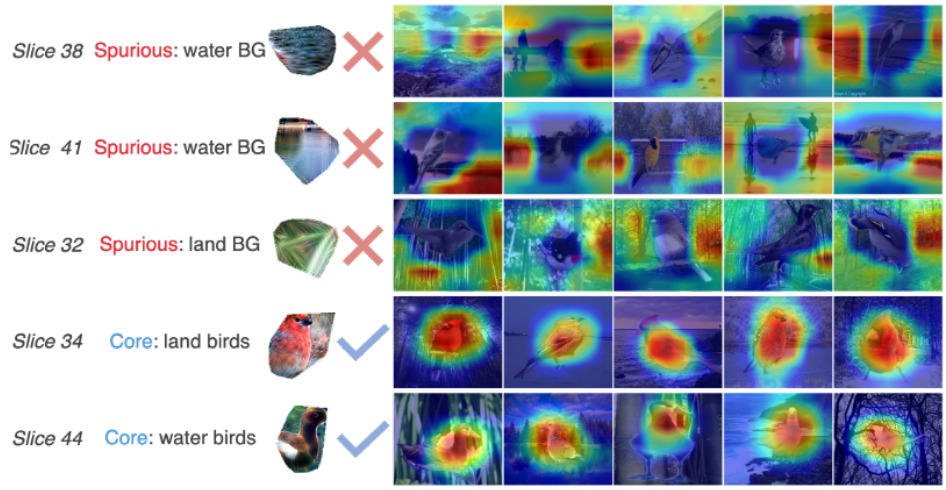


Figure 3.11: Examples of findings when ML experts validate a bird category classifier, where experts identify problematic slices where the model uses backgrounds (BG) to classify birds.

in determining whether the detected bias is pervasive throughout the dataset. By analyzing slices that are distant from the annotated ones in the *Attribution Mosaic* and are assigned with low Spuriousness possibilities, they discovered that the farthest neighbors, namely *slice_34* and *slice_5*, correspond to core features. This suggests that the model can correctly capture bird regions in these slices (Fig. 3.8 B C), which raises a follow-up "why" question. To understand in what circumstances when the model fails. They browse the original images from *slice_32* (spurious feature) and *slice_34* (core feature), respectively, as shown in Fig. 3.8 D and C. They find *slice_32* has very similar land backgrounds and very different birds, while on the other hand, the birds' appearance of *slice_34* is very consistent. This finding explains why this biased model can successfully capture the core features from *slice_34* but fails at *slice_32*—greater similarities in the representation space indicate stronger features. In other words, core features in *slice_34* are strong enough to support the model's robustness against bias. Such insights are helpful in improving model robustness and have been further studied by ML experts [23].

• **Insights summary.** A summary of insights obtained from this case study is provided in Fig. 3.11,

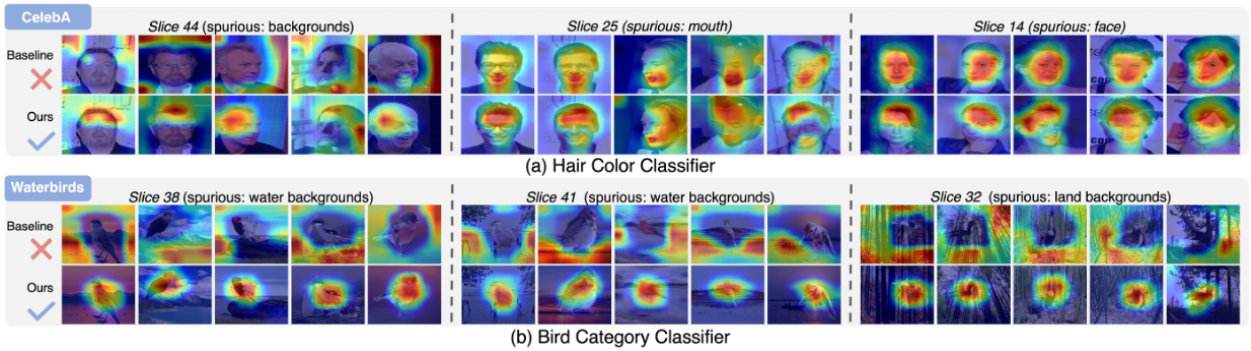


Figure 3.12: Qualitative evaluation with GradCAM attribution heatmaps. We visually compare the original and the *AttributionScanner*-improved models for hair color and bird category classification, respectively. In each sub-figure, the first row refers to the original model (baseline), and the second row refers to the improved model (ours). We can observe *AttributionScanner* suppresses the spurious correlations by using the correct features for predictions.

where ML experts validate the existence of model biases and extract corresponding slices.

3.3.3 Evaluation

3.3.3.1 Quantitative Evaluation

The quantitative evaluation involves four matrices [171], including clean accuracy, core accuracy, spurious accuracy, and relative core sensitivity (RCS). We revisit their definitions as follows:

• **Clean Accuracy.** The original model accuracy, where larger values are indicative of better overall accuracy.

• **Core Accuracy** $acc^{(C)}$**.** The model accuracy when spurious regions are occupied with Gaussian noise, where larger values are indicative of the model's more reliance on core regions.

• **Spurious Accuracy** $acc^{(S)}$**.** The model accuracy when core regions are added with Gaussian noise, where larger values are indicative of the model's more reliance on spurious regions.

• **RCS.** This metric quantifies the model's reliance on core features while controlling for general noise robustness. It is defined as the ratio of the absolute gap between core and spurious accuracy, and the total possible gap for any model between core and spurious accuracy. Represented as $RCS = \frac{acc^{(C)} - acc^{(S)}}{2 \times min(\alpha, 1-\alpha)}$, where $\alpha = \frac{acc^{(C)} + acc^{(S)}}{2}$. RCS ranges from 0 to 1, a higher value indicative of

Table 3.3: Quantitative evaluation of the performance of the hair color classification models trained on the original and *AttributionScanner*(*AS*)-improved CelebA dataset.

Training Procedure	Clean Acc (↑)	Core Acc (↑)	Spurious Acc (↓)	RCS (↑)
Baseline	98.02688	97.21170	97.61917	0.067720
AS (Annotation)	98.02688	98.02185	96.92958	0.216351
AS (Propagation)	**98.08225**	**98.16278**	**96.01852**	**0.368512**

better model performance.

In the two case studies, ML experts annotated five spurious slices in the CelebA dataset, {*slice_25, slice_14, slice_2, slice_44, slice_26*}, and six spurious slices in the Waterbirds dataset, {*slice_38, slice_41, slice_32, slice_29, slice_4, slice_10*}, respectively. For each case study, *AttributionScanner* automatically ran the Label Propagation algorithm and exported both the users' annotation records and the propagated Spuriousness for further investigation.

To thoroughly evaluate our introduced method, we validate three models for each case. The models marked as "baseline" are the original trained models obtained at the beginning of each case study. The models marked as "*AttributionScanner*" are re-trained using the CoRM method after adding noise to "spurious" slices according to results exported from *AttributionScanner*. In particular, "Annotation" indicates that only user-annotated spurious slices were corrupted with noise, while "Propagation" indicates that propagated Spuriousness is used to identify spurious slices to be added with noise.

Table 3.3 and Table 3.4 present the quantitative evaluation results for the two case studies, respectively. Our presented *AttributionScanner* can significantly improve the vision model's overall performance with reduced spurious correlations. Furthermore, our Label Propagation largely reduces human effort by automating the annotation process, and achieves the best performance in this quantified evaluation. Overall, our results demonstrate that *AttributionScanner* is an effective approach for mitigating spurious correlations in machine learning models, and the Label Propagation

Table 3.4: Quantitative evaluation of the performance of the bird category classification models trained on the original and *AttributionScanner*(*AS*)-improved Waterbirds dataset.

Training Procedure	Clean Acc (↑)	Core Acc (↑)	Spurious Acc (↓)	RCS (↑)
Baseline	85.73812	82.98582	82.82901	0.004878
AS (Annotation)	89.57465	86.40534	79.81651	0.195062
AS (Propagation)	**90.40867**	**87.40617**	**77.89825**	**0.274038**

algorithm is a valuable tool for automating the annotation process.

3.3.3.2 Qualitative Evaluation

For further evaluation, we visually compare models' attributions via GradCAM in Fig. 3.12. Refer to Sec. 3.3.2.1, the hair color classifier originally has spurious correlations dominant in {*slice_44, slice_25, and slice_14*}, where the model uses image backgrounds, mouth, or eyes to predict hair color, respectively. With the help of *AttributionScanner*, the model's misattribution is successfully fixed, directing focus towards the correct hair regions (refer to Fig. 3.12(a)). As for the bird category classification model, it originally focuses on spurious features, water/land backgrounds, to decide whether there are water/land birds on the input image {*slice_38, slice_41, and slice_32*} (refer to Sec. 3.3.2.1). In Fig. 3.12(b), we can observe that *AttributionScanner* mitigates these issues by helping the model to focus on the core bird features.

3.3.3.3 Experts Feedback

We conduct two case studies with ten ML experts to evaluate *AttributionScanner*. They had not previously seen *AttributionScanner*. Five are male, and five are female. Six of the experts have over five years of experience in ML, two have between three to five years of experience, and two have between one to three years of experience. All experts need to conduct model validation in their research and have experience working with vision models. Our findings are drawn from their comments during the study, as they follow the "think-aloud" protocol, and additional feedback is gathered through exit discussion and questionnaire.

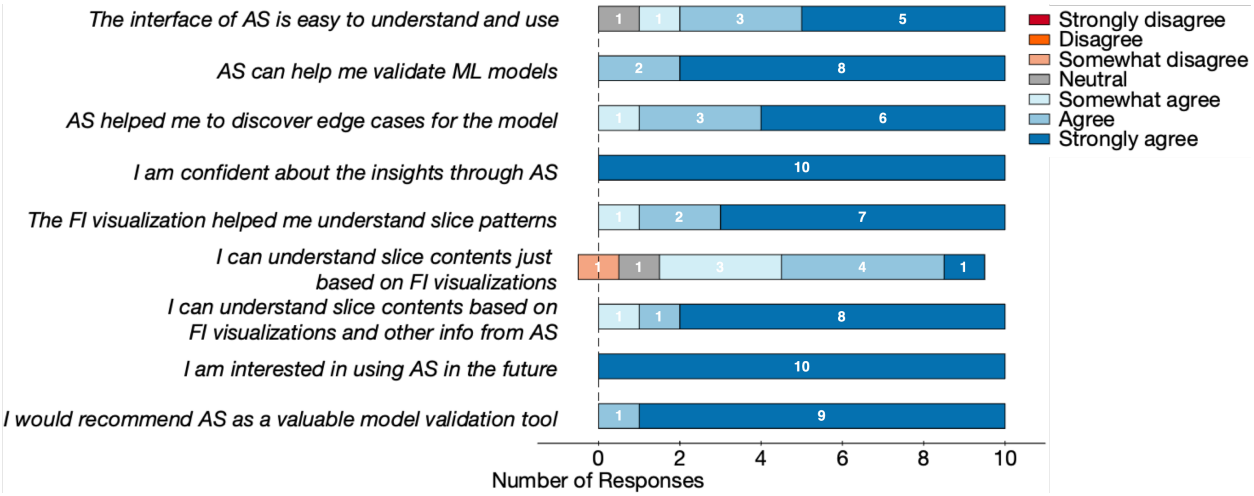


Figure 3.13: Expert perception of the system, according to eight Likert-Type Questions using a 7-point scale. (*AS* = *AttributionScanner*.)

Summary of Likert-Type Questions. The exit questionnaire included nine Likert-type questions on a seven-point scale, aiming to evaluate *AttributionScanner* from various aspects. The feedback, as shown in Fig. 3.13, reflects a positive overall impression. Notably, in the usability evaluation, five experts strongly agreed, and three agreed that our system is user-friendly and understandable. In terms of effectiveness, eight experts strongly agreed that our system aids in validating models. All experts expressed strong confidence in the insights provided. Considering that Feature Inversion (FI) results are synthesized images that might initially be hard to grasp, we included questions to assess this design aspect. Seven experts strongly agreed, and two agreed that FI is helpful for understanding slice patterns. While two experts expressed neutral or somewhat disagreeing opinions about understanding slice contents based solely on FI, eight strongly agreed that *AttributionScanner*'s coordination of FI and other views supports a comprehensive understanding of slice contents. Lastly, all experts expressed interest in using our system in the future and would recommend it as a valuable tool for model validation.

Interpretable Model Validation. All experts acknowledged that *AttributionScanner* "*effectively*

helps ML model validation". They remarked that the complimentary visual summaries and attribution heatmaps help them "*easily understand slice issues*". "*It's often hard to figure out what's going wrong with a model*", one expert said, "*This system guides me to find where I should look and explains the issues intuitively.*" Another expert noted, "*The ability to browse what's happening in different slices at once greatly helped me understand the model.*" They commented that they are confident about their insights because our system "*highlights slice errors and provides supportive evidence*". Three of them are eager to see what issues *AttributionScanner* could uncover for the models they are currently using.

Attribution Mosaic. All participants showed particular interest in the *Attribution Mosaic* and asked about how it was designed in the interview. Seven experts were impressed by its ability to summarize model attributions across data subgroups. Two experts noted that "*it highlights model issues intuitively.*" Another described it as "*impressive and novel*," emphasizing that it "*highlights and explains the model's failures*." Another expert commented, "*This view provides a new angle for me to identify and understand the data/model issues*." All experts quickly adapted to using Spuriousness propagation and utilized the Spuriousness matrix to speed up their annotation. Specifically, five experts found that the Spuriousness propagation "*surprisingly helpful*," and six remarked that it "*really saved my effort*".

3.4 Mitigating Spuriousness with Minimal Annotations

3.4.1 Problem Formulation

We introduce the formal definition of the spuriousness mitigation problem[225]. Assume training data $\mathscr{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n}$ with input features $\boldsymbol{x}_i \in \mathbb{R}^d$ and labels $y_i \in [\mathscr{L}]$, where $\mathscr{L} = \{L_1, \ldots, L_l\}$ are the classes in $\mathscr{D}$. Classification models are trained to minimize the empirical risk:

$$\boldsymbol{\theta}^* \in \arg\min_{\boldsymbol{\theta}} \mathop{\mathbb{E}}_{(\boldsymbol{x_i}, y_i) \in \mathscr{D}} [l(f(\boldsymbol{\theta}, \boldsymbol{x}_i), y_i)], \tag{3.3}$$

where $\boldsymbol{\theta}$ are model parameters, $f(\boldsymbol{\theta}, \boldsymbol{x}_i)$ and $l(*)$ are model outputs and loss function, respectively.

Groups. In spuriousness mitigation, we assume there exists a core feature set $\mathscr{C} = \{c_1, \ldots, c_l\}$ and a spurious feature set $\mathscr{S} = \{s_1, \ldots, s_k\}$ in $\mathscr{D}$. Each class L_i is uniquely identified by a core feature c_i, while spurious features s_j can be found across different classes. We consider an instance in the dataset as having both a core feature c_i and a spurious feature s_j, and each instance belongs to a unique group g_{c_i,s_j}. Core features are reliable class indicators, while spurious features with misleading co-occurrence with class in training may miss such correlations in the testing phase. If the model mistakenly links spurious features with classes in training, it may fail in testing when such spurious features are absent. This can lead to lower accuracy rates, especially for less common group variations that do not exhibit the spurious feature the model has learned to rely on.

Worst-group accuracy. The vital evaluation of spuriousness mitigation focuses on the model's performance in the subgroup where it is least effective: the worst-group accuracy, calculated by:

$$Acc_w = \min_{g \in \mathscr{G}} \left(\frac{1}{|g|} \sum_{(x_i, y_i) \in g} \mathbf{1}[f(x_i) = y_i]\right), \tag{3.4}$$

where $\mathscr{G} = \{\cup_{c_i,s_j} g_{c_i,s_j}\}$ is the union of subgroups defined by core feature c_i and spurious feature s_j, and $f(x_i)$ is the label predicted by the model. This metric is crucial because it directly assesses how effectively a model can generalize across different subgroups, especially those that might be mostly underrepresented because of spurious correlations. It ensures that improvements in model performance are not just reflective of gains on the majority or less challenging subsets of the data.

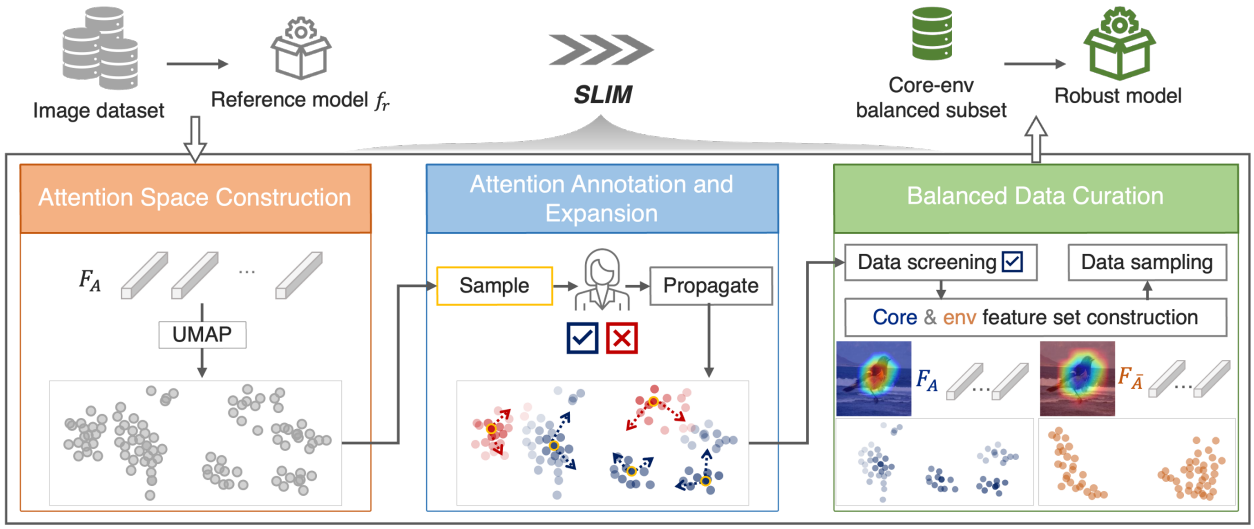


Figure 3.14: Overview of *SLIM* framework, with a data construction pipeline consisting of three phases: (1) **Attention Space Construction**, creating a space with data features and model attention aligned locally; (2) **Attention Annotation and Expansion**, where instances sampled from the attention space are labeled by human for attention correctness, and labels are propagated to neighboring instances; and (3) **Balanced Data Curation**, which filters out instances with incorrect attention, and utilize attention-weighted (F_A) and inverse-attention-weighted ($F_{\bar{A}}$) feature vectors to create core and environment feature sets, forming subgroups to assemble a feature-balanced subset for training a spuriousness-robust model.

3.4.2 Method

In this section, we detail the method behind *SLIM*. In alignment with related works requiring no spuriousness label, our approach also follows a two-stage procedure: inferring groups with an ERM-trained reference model f_r, and then leveraging the inferred groups to train a robust model. As an approach oriented by data quality, *SLIM* framework comprises a three-phase data construction pipeline, as illustrated in Fig. 3.14. Firstly, we build an attention representation space to group instances with both similar data features and model attention, preparing for the data sampling and label propagation. Following this, we sample typical instances for human to evaluate model's attention correctness with yes-no feedback, then estimate neighboring instances' attention labels accordingly. In the last phase, we filter out data with wrong attention and leverage attention matrix to disentangle core and environment features. With constructed feature sets, we

apply consistency-weighted data sampling to curate a feature-balanced subset, with a desired core-environment balanced distribution. Finally, *SLIM* framework is concluded by a training process with the curated subset, resulting in a model exhibiting robustness against spurious correlations.

Considering the benefits of spurious information and drawbacks of the annotation cost, we deploy a binary labeling scheme to simplify descriptive labeling tasks into yes-no feedback on model's attention correctness. Moreover, we aim to further facilitate this process by avoiding the necessity of exhaustive labeling. Our empirical observations indicate that a model would have consistent attention mechanisms on specific data subgroups. If such subgroups can be found, it would largely aid the labeling workload by enabling label estimations among neighbors. Driven by this, we leverage the feature representation space constructed in Sec. 3.3.1.4, where instances with similar attention outcomes (as indicated by highlighted image features) are closely grouped.

Note that for a fair comparison, in alignment with baseline methods, experiments in this work involve the CNN model (ResNet) for classification and GradCAM for obtaining model attention. But our framework is fully applicable to (1) other vision model architectures such as vision transformers (ViT)[226] or Swin transformers[227], and (2) other heatmap-based model visualization techniques, such as RISE[64] or ScoreCAM[228].

3.4.3 Attention Annotation and Expansion

Given that our attribution representation space ensures locally consistent model attention, it allows for annotating a selectively sampled subset of instances and then estimating neighboring labels accordingly, thereby enabling efficient transfer of human expertise. The criteria for selecting these instances are twofold: (1) *diversity*, to encompass a broad covering of various data features and model attention patterns; (2) *typicality*, ensuring they are representative of their neighboring instances. To achieve this, we utilize the k-means clustering algorithm to partition our constructed

attribution representation space into n clusters (*diversity*). We then select n instances that are closest to each cluster's center (*typically*). The choice of n is aimed at optimizing the balance between minimizing intra-cluster variance in data features and model attention patterns, and keeping the annotation budget manageable. Our empirical analysis, detailed in Sec. 3.4.9, reveals that the intra-cluster consistencies increase with n, but this trend levels off upon reaching a specific value of n–the elbow point. Thus, selecting n at this elbow point optimizes consistency without unnecessarily increasing the annotation workload.

We gather binary yes-no feedback from human experts on the correctness of the model's attention for these n instances, feeding this feedback along with our attribution representation space embedding into the label spreading algorithm. This algorithm leverages data similarity and density to extend attention labels to previously unlabeled data[221]. Ultimately, this process yields a probability score for correct attention for each instance across the dataset.

3.4.4 Balanced Data Curation

Following Sec. 3.4.3 that provides attention accuracy scores, we proceed to construct our data subset to support model's unbiased learning, as described below:

Data screening. Inspired by works[229, 230, 231], data receiving correct attention from a reference model are more likely to enhance the learning of core features in a robust model within a more balanced data distribution. Furthermore, our study indicates that even a reference model with inherent biases has the potential to accurately identify core features for a data subset. Such correctly captured features typically exhibit diversity, ensuring good coverage across the spectrum of the core feature set. This understanding informs our approach to curating a data subset that: (1) selects instances based on accurate model attention, and (2) ensures a balanced representation of core features across various environment features. Thus we first apply data screening to filter out

instances with wrong attention according to attention accuracy scores.

Disentangling core and environment features. After screening, the retained data subset mainly comprises instances with correct attention, reflected by attention masks that highlight core features, meaning that the corresponding attribution vector A in the latent space also highlights core features of feature vector F. Moreover, existing studies have found that a model's feature vectors are representative of most input features, regardless of whether they are used for decision-making[25, 232]. Therefore, utilizing A as the weight matrix, we disentangle core and environment features in the latent space by weighting F by attention masks A and inverse attention masks $\bar{A}$. K-means clustering is then applied to partition all attention-weighted feature vectors F_A, and inverse-attention-weighted feature vectors $F_{\bar{A}}$, respectively, facilitating the formation of representative subgroups for the core feature set ($\hat{\mathscr{C}} = \hat{c}_1, \ldots, \hat{c}_N$) and the environment feature set ($\mathscr{E} = e_1, \ldots, e_M$). The number of clusters, determined through the elbow method, culminates in $M \times N$ subgroups $g_{\hat{c}_i, e_j}$, each characterized by a distinct combination of core and environment features $(\hat{c}_i, e_j)$.

Data sampling. Given the unequal distribution of core and environment features, the internal consistency across each constructed subgroup $g_{\hat{c}_i, e_j}$ varies. Our sampling strategy aims to mitigate over-representation of particular features to prevent overfitting, while ensuring broad coverage of diverse core and environment features. We assess cluster consistency as a metric for determining the sample size from each cluster. For a cluster $\hat{c}_i$ in the core feature space, denote the distance between an instance x and the cluster center as $d_{\hat{c}_i}(x)$. The cluster consistency is defined as the reciprocal of the average distance per cluster:

$$\rho_{\hat{c}_i} = \frac{1}{\frac{1}{|\hat{c}_i|} \sum_{x \in \hat{c}_i} d_{\hat{c}_i}(x)}. \tag{3.5}$$

A lower $\rho_{\hat{c}_i}$ signals less consistent features, necessitating a higher sample weight for adequately

representing them. A cluster's sampling weight is defined as $w_{\hat{c}_i} = \frac{1}{\rho_{\hat{c}_i}}$ and the number of samples to be drawn from a core cluster $\hat{c}_i$ is:

$$n_{\hat{c}_i} = N \cdot \frac{1/\rho_{\hat{c}_i}}{\sum_{j=1}^{|\hat{\mathscr{C}}|} 1/\rho_{\hat{c}_j}}, \tag{3.6}$$

where N is the sampling budget. Moving to the environment feature space, the $n_{\hat{c}_i}$ instances from a core feature cluster $\hat{c}_i$ may distribute across multiple environment clusters $\{e_1, \ldots, e_M\}$ and formulate subgroups $\{g_{\hat{c}_i,e_1}, \ldots, g_{\hat{c}_i,e_M}\}$. A similar methodology is employed for a certain group $g_{\hat{c}_i,e_i}$. The average distance for $g_{\hat{c}_i,e_i}$ in the environment representation space is:

$$\rho_{\hat{c}_i,e_i} = \frac{1}{\frac{1}{|\hat{c}_i,e_i|}\sum_{x\in(\hat{c}_i,e_i)} d_{e_i}(x)}. \tag{3.7}$$

Similarly, we have the sampling weight $w_{\hat{c}_i,e_i} = \frac{1}{\rho_{\hat{c}_i,e_i}}$ and the number of samples to be drawn from a certain group $g_{\hat{c}_i,e_i}$ is finally defined by:

$$n_{\hat{c}_i,e_i} = n_{\hat{c}_i} \cdot \frac{w_{\hat{c}_i,e_i}}{\sum_{j=1}^{|\mathscr{E}(\hat{c}_i)|} w_{\hat{c}_j,e_i}} = N \cdot \frac{1/\rho_{\hat{c}_i}}{\sum_{j=1}^{|\hat{\mathscr{C}}|} 1/\rho_{\hat{c}_j}} \cdot \frac{1/\rho_{\hat{c}_i,e_i}}{\sum_{j=1}^{|\mathscr{E}(\hat{c}_i)|} 1/\rho_{\hat{c}_i,e_j}}, \tag{3.8}$$

where $|\hat{\mathscr{C}}|$ denotes the number of core clusters and $|\mathscr{E}(\hat{c}_i)|$ denotes the number of subgroups formulated by instances from cluster $\hat{c}_i$. To summary, our sampling strategy takes feature consistencies and sampling budget into consideration, and ensures fair representation of different core-environment feature combinations across the feature spaces.

3.4.5 Training

In our pursuit to mitigate spurious correlations, we emphasize the pivotal role of data quality by ensuring a balanced distribution of core features across various environmental contexts within our dataset. Given this focus, we employ a basic training approach for our model to learn from this well-prepared dataset.

We apply our data curation pipeline separately to the training ($SLIM_{Tr}$) and validation ($SLIM_{Val}$) sets. For the data constructed from the training set, we adopt the conventional ERM. For the dataset derived from the validation set, we employ the training technique presented by DFR[25]. It involves retraining only the final linear layer of a classification model that was initially trained using ERM. Note that our approach diverges from DFR practices by utilizing our curated dataset instead of a metagroup-balanced dataset.

3.4.6 Experiments

This section assesses *SLIM*'s efficacy by investigating the following questions:

- **Q1: Spuriousness mitigation performance.** How effective is *SLIM* in mitigating spuriousness?
- **Q2: Attention labeling performance.** How does *SLIM*'s annotation efficiency stand against spuriousness labeling, and is there consistency in its representation space to support label propagation?
- **Q3: Ablation study.** How does each component affect *SLIM*'s performance?

3.4.7 Experimental Setup

Datasets. Our study explores a range of datasets with inherent spurious correlations to challenge image classification models, including: (1) **Waterbirds**[174]. Contrasting landbirds and waterbirds against congruent/incongruent backgrounds, this dataset tests models' ability to focus on relevant features for bird type classification. (2) **CelebA**[233]. A celebrity image dataset where the focus is on distinguishing hair color (blond vs. non-blond) to examine biases in associating hair color with gender. (3) **ISIC**[173]. Aiming to differentiate benign from malignant melanoma in dermoscopic images, we target colorful patches as spurious features, aligning with previous studies[211]. (4) **NICO**. Derived from NICO++[234], this dataset features various object categories in shifted contexts to probe spurious correlations. (5) **ImageNet-9**[235]. Comprising 9 super-classes from

ImageNet, this dataset is crafted to test models' robustness against background variations.

Baselines. We benchmark *SLIM* against a spectrum of SOTA spuriousness mitigation techniques, categorized as follows: (1) Techniques not reliant on spuriousness labels: EIIL[202], PGI[203], GEORGE[180], LfF[201], CIM[200], JTT[24], and CNC[23]; (2) Techniques requiring additional information such as predefined spuriousness concepts: DISC[236]; (3) Methods utilizing spuriousness labels from validation sets: SSA[29] and DFR[25]; (4) Approaches utilizing spuriousness labels from training sets: GDRO[174] and PDE[231].

Evaluation. Aligning with baselines, our experiments involve CNN models with ResNet50 architecture and GradCAM for visualizing model attention. For the capability of performance enhancement, we evaluate the worst-group and average accuracy. Additionally, we consider the necessity for extra labels and the additional training cost to evaluate computational efficiency. Furthermore, we examine the attention precision to evaluate whether model attention is corrected. All quantification experiments are based on five independent runs with different random seeds.

3.4.8 Spuriousness Mitigation Performance (Q1)

In this section, we present the performance evaluation by comparing different methods across multiple binary and multiclass classification datasets mentioned in Sec. 3.4.7.

Superior worst-group performance. The metric of worst-group accuracy (Acc_w) is pivotal for spuriousness mitigation methods (refer to Sec. 3.4.1), and the results in Tables 3.5 and 3.6 showcase *SLIM*'s superior capability in this regard. Across various classification tasks, *SLIM* consistantly outperforms competing methods in achieving the highest Acc_w while maintaining high average accuracy. Specifically, in binary classification scenarios (Table 3.5), *SLIM* leads in performance for Waterbirds and ISIC and secures a close second in CelebA. In multi-classification tasks (Table 3.6), $SLIM_{Tr}$ and $SLIM_{Val}$ rank as the top performers across the board, underscoring the effectiveness of

Table 3.5: Worst-group and average accuracy (%) comparison on binary classification tasks, highlighting **best** and second-best performances. ♦ and ♢ indicate methods requiring spuriousness concepts and attention labels, respectively.

Method	Spuriousness label	Add. training cost	Waterbirds		CelebA		ISIC	
			Worst	Avg	Worst	Avg	Worst	Avg
ERM	✗	0×	$62.6_{\pm 0.3}$	$97.3_{\pm 1.0}$	$47.7_{\pm 2.1}$	$94.9_{\pm 0.3}$	$56.7_{\pm 2.2}$	$79.0_{\pm 1.3}$
EIIL	✗	0×	$83.5_{\pm 2.8}$	$94.2_{\pm 1.3}$	$81.7_{\pm 0.8}$	$85.7_{\pm 0.1}$	$72.3_{\pm 3.9}$	$86.4_{\pm 1.7}$
PGI	✗	0×	$79.5_{\pm 1.9}$	$95.5_{\pm 0.8}$	$85.3_{\pm 0.3}$	$87.3_{\pm 0.1}$	$67.4_{\pm 1.9}$	$85.7_{\pm 1.1}$
GEORGE	✗	1×	$76.2_{\pm 2.0}$	$95.7_{\pm 0.5}$	$54.9_{\pm 1.9}$	$94.6_{\pm 0.2}$	$58.8_{\pm 2.1}$	$83.2_{\pm 1.4}$
LfF	✗	1×	$77.3_{\pm 2.3}$	$91.4_{\pm 1.7}$	$76.8_{\pm 1.2}$	$84.5_{\pm 0.7}$	$72.1_{\pm 2.1}$	$85.4_{\pm 0.5}$
CIM	✗	1×	$77.8_{\pm 1.5}$	$94.7_{\pm 1.1}$	$83.8_{\pm 1.0}$	$90.5_{\pm 0.6}$	$71.7_{\pm 1.6}$	$85.9_{\pm 1.1}$
JTT	✗	5×-6×	$83.1_{\pm 3.5}$	$90.6_{\pm 0.3}$	$81.5_{\pm 1.7}$	$88.1_{\pm 0.3}$	$28.1_{\pm 4.3}$	$86.2_{\pm 0.3}$
CNC	✗	2×-12×	$88.5_{\pm 0.3}$	$90.9_{\pm 0.1}$	$88.8_{\pm 0.9}$	$89.9_{\pm 0.5}$	$68.2_{\pm 0.2}$	$84.6_{\pm 0.2}$
DISC	✗♦	1×	$88.7_{\pm 0.7}$	$93.8_{\pm 0.7}$	$85.7_{\pm 1.1}$	$89.9_{\pm 0.7}$	$73.4_{\pm 0.4}$	$84.7_{\pm 0.4}$
SLIM$_{Tr}$	✗♢	0.1×-0.2×	$89.1_{\pm 0.6}$	$91.7_{\pm 0.5}$	$89.4_{\pm 0.3}$	$90.1_{\pm 0.2}$	**79.1**$_{\pm 0.5}$	$88.5_{\pm 0.4}$
SLIM$_{Val}$	✗♢	0.05×-0.3×	**91.0**$_{\pm 0.4}$	$93.8_{\pm 0.4}$	$89.4_{\pm 0.5}$	$90.7_{\pm 0.2}$	$74.0_{\pm 0.2}$	$85.4_{\pm 0.2}$
SSA	validation	1.5×-5×	$89.0_{\pm 0.6}$	$92.2_{\pm 0.9}$	**89.8**$_{\pm 1.3}$	$92.8_{\pm 0.1}$	$61.7_{\pm 0.7}$	$77.7_{\pm 1.0}$
DFR	validation	0.05×-0.3×	$89.6_{\pm 0.7}$	$92.1_{\pm 0.2}$	$87.3_{\pm 1.0}$	$90.2_{\pm 0.8}$	$71.9_{\pm 0.7}$	$84.3_{\pm 0.3}$
GDRO	training	0×	$89.9_{\pm 0.6}$	$92.6_{\pm 0.1}$	$88.9_{\pm 1.3}$	$93.9_{\pm 0.1}$	$66.8_{\pm 3.5}$	$80.5_{\pm 1.9}$
PDE	training	0×	$85.9_{\pm 2.7}$	$90.5_{\pm 1.8}$	$85.0_{\pm 4.0}$	$92.3_{\pm 0.8}$	$66.4_{\pm 2.3}$	$82.2_{\pm 1.3}$

Table 3.6: Worst-group and average accuracy (%) comparison on multi-classification tasks, highlighting **best** and second-best results. ♢ indicates methods requiring attention labels. - represents inapplicable methods because such methods require spuriousness labels while the dataset does not provide them.

Method	Spuriousness label	Add. training cost	NICO		ImageNet9	
			Worst	Avg	Worst	Avg
ERM	✗	0×	$11.9_{\pm 0.2}$	$70.9_{\pm 1.2}$	$55.3_{\pm 2.0}$	$75.6_{\pm 1.1}$
CNC	✗	5×-12×	$72.6_{\pm 1.7}$	$89.5_{\pm 0.6}$	$58.9_{\pm 0.6}$	$74.3_{\pm 0.2}$
SLIM$_{Tr}$	✗♢	0.1×-0.2×	**78.0**$_{\pm 0.8}$	$92.1_{\pm 0.1}$	**68.1**$_{\pm 0.3}$	$74.5_{\pm 0.2}$
SLIM$_{Val}$	✗♢	0.1×-0.2×	$73.8_{\pm 0.4}$	$90.7_{\pm 0.2}$	$64.6_{\pm 0.4}$	$74.2_{\pm 0.3}$
DFR	validation	0.4×	$70.9_{\pm 2.1}$	$85.8_{\pm 0.6}$	-	-
GDRO	training	0×	$73.3_{\pm 1.0}$	$90.3_{\pm 0.5}$	-	-

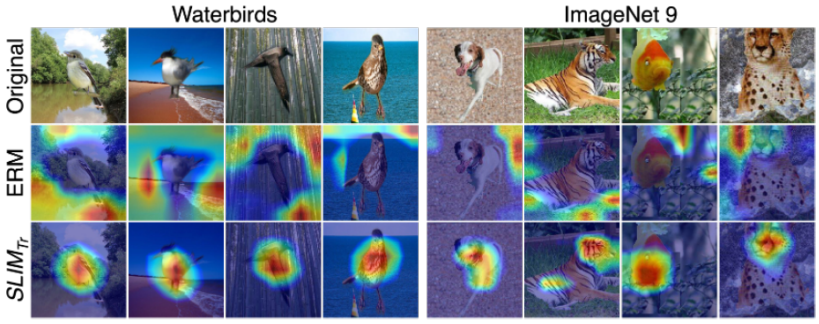
Table 3.7: *SLIM*'s primary cost, including the data amount required for attention annotation *(Att.)*, and the constructed data size for model training *(Training)*.

Method	Waterbird		CelebA		ISIC		NICO		ImageNet 9	
	Att.	Training	Att.	Training	Att.	Training	Att.	Training	Att.	Training
SLIM$_{Tr}$	120 (2.5%)	0.1×	200 (0.12%)	0.1×	120 (0.7%)	0.2×	40 (1.8%)	0.2×	180 (0.4%)	0.1×
SLIM$_{Val}$	40 (3%)	0.1×	70 (0.35%)	0.05×	40 (2%)	0.3×	20 (2.7%)	0.2×	50 (1%)	0.1×

our approach in spuriousness mitigation.

Cost-efficient spuriousness mitigation. Methods in Tables 3.5 and 3.6 are divided into four groups from top to bottom, as per the categorization established in Sec. 3.4.7 according to the

Table 3.8: Attention accuracy evaluation. (Left) GradCAM visualization of model attention. (Right) Worst group and average attention accuracy of selected methods.



Waterbirds	AIoU	
	Worst	Avg
ERM	$.25_{\pm.05}$	$.47_{\pm.03}$
CNC	$.48_{\pm.05}$	$.49_{\pm.04}$
GDRO	$.49_{\pm.03}$	$.52_{\pm.03}$
PDE	$.50_{\pm.02}$	$.52_{\pm.03}$
SLIM$_{Tr}$	$.53_{\pm.02}$	$.61_{\pm.02}$
ImageNet 9	AIoU	
	Worst	Avg
ERM	$.33_{\pm.07}$	$.55_{\pm.03}$
CNC	$.44_{\pm.03}$	$.61_{\pm.03}$
SLIM$_{Tr}$	$.58_{\pm.04}$	$.68_{\pm.03}$

computational cost. The *"Add. training cost"* columns denote the additional training cost for mitigating spuriousness apart from the initial training, such as additional learning phases (CNC, SSA) or data augmentation (JTT). Results in the table substantiate that methods with higher costs (higher *Add. training costs* or more spuriousness labels) tend to outperform others, aligning with prior research findings. Nevertheless, among the well-performed methods, it is worth mentioning that *SLIM* is more cost-efficient: *SLIM*$_{Tr}$ used 0.1-0.2 proportion of training sets and *SLIM*$_{Val}$ used 0.05-0.3 proportion of validation sets for additional training, which is the same as DFR and much smaller than others. Moreover, *SLIM* requires labels for attention correctness, which is more cost-efficient to obtain than spuriousness labels (refer to Sec. 3.4.9 for more details). Table 3.7 presents *SLIM*'s primary cost, including the training data amount for mitigating spuriousness, which is the detailed version of *"Add. training cost"* in Tables 3.5 and 3.6, and the data amount requiring attention labels *(Att.)*. Notably, our approach only requires no more than 3% of the corresponding data split (train or validation) for attention-label annotation, further proving the cost-efficiency of *SLIM*.

Enhanced attention accuracy. Beyond classification accuracy, we further evaluate the perfor-

mance of spuriousness mitigation methods via attention accuracy. Specifically, for models resulting from each method, we randomly sample data instances to visualize model attention, and measure attention accuracy scores in both worst-group and dataset levels, consistent with our evaluation of classification accuracy. Align with the related work[207], we employ the Adjusted Intersection-over-Union (AIoU) score as our attention accuracy measurement, which is an adjusted IoU score with a threshold, aiming at measuring the overlap between model's highest attributed region and the ground-truth bounding box. Table 3.8 presents the evaluation results. The left GradCAM examples showcase *SLIM*'s capability of correcting model's wrong attention, and the right table compares *SLIM* with SOTA methods, where we can find our method secures the best performance in terms of both worst-group and average attention accuracy.

3.4.9 Attention Labeling Performance (Q2)

To compare the processes of annotating spuriousness versus attention correctness, we employed crowdsourcing tasks using the Waterbirds and NICO datasets, as both of which come with ground-truth spuriousness labels. For this study, we selected a random set of 120 images from each dataset and established two separate tasks: one for spuriousness labeling and one for attention correctness labeling. For each task, we recruited 60 workers independently from the Prolific platform to prevent learning biases from cross-task participation. We meticulously recorded the time taken to label each image and reviewed all submissions to exclude any outliers with unusually long or short annotation times. Ensuring ethical research standards, our study avoided collecting personally identifiable information and excluded any potentially offensive content.

Evaluation Consistency. Table 3.9 presents metrics for this study. While the annotation time for Waterbirds remains similar across tasks, attention correctness for NICO is annotated more quickly than the other. This difference may stem from NICO's broader array of spuriousness

Table 3.9: Comparison of crowd-sourced labeling tasks: spuriousness vs. attention correctness.

Waterbirds	Avg. Time	Avg. Consis.
Spuriousness	$4.9_{\pm 2.3}$ sec	$81.9_{\pm 14.8}\%$
Att.	$4.5_{\pm 1.4}$ sec	$94.6_{\pm 10.5}\%$
NICO	Avg. Time	Avg. Consis.
Spuriousness	$7.9_{\pm 3.8}$ sec	$70.1_{\pm 19.1}\%$
Att.	$5.3_{\pm 1.6}$ sec	$95.1_{\pm 6.1}\%$

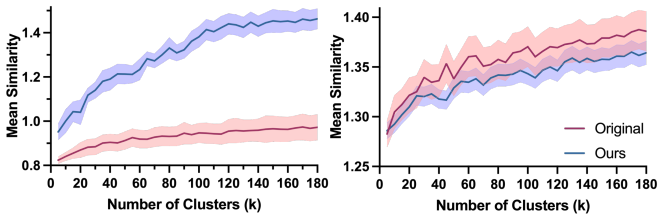


Figure 3.15: Comparison of neighbor consistency over original feature space and our attribution representation space on Waterbirds, w.r.t. model attention (left) and data features (right).

Table 3.10: Performance variation with different attention annotation amounts in *SLIM*. Marks in table: **best** and second best performance; the amounts used in Table 3.5.

Method	Waterbirds			CelebA			ISIC		
	Amount	Worst	Avg	Amount	Worst	Avg	Amount	Worst	Avg
$SLIM_{Tr}$	60	$78.2_{\pm 0.5}$	$86.0_{\pm 0.3}$	100	$86.4_{\pm 0.2}$	$90.2_{\pm 0.1}$	60	$75.3_{\pm 0.2}$	$86.6_{\pm 0.9}$
	90	$85.9_{\pm 0.9}$	$89.3_{\pm 0.3}$	150	$87.8_{\pm 0.4}$	$91.1_{\pm 0.7}$	90	$74.4_{\pm 0.4}$	$87.4_{\pm 0.7}$
	120	$89.1_{\pm 0.6}$	$91.7_{\pm 0.5}$	200	$89.4_{\pm 0.4}$	$90.1_{\pm 0.2}$	120	$79.1_{\pm 0.5}$	$88.5_{\pm 0.4}$
	150	**89.4**$_{\pm 0.6}$	$91.7_{\pm 0.2}$	250	$89.4_{\pm 0.2}$	$90.9_{\pm 0.1}$	150	**79.9**$_{\pm 0.8}$	$88.8_{\pm 0.7}$
	180	$89.3_{\pm 0.3}$	$91.0_{\pm 1.5}$	300	**89.5**$_{\pm 0.1}$	$91.2_{\pm 0.2}$	180	$79.8_{\pm 0.3}$	$88.6_{\pm 0.3}$
$SLIM_{Val}$	10	$78.9_{\pm 8.7}$	$86.5_{\pm 5.1}$	40	$84.7_{\pm 0.4}$	$87.4_{\pm 0.3}$	10	$70.5_{\pm 0.5}$	$83.3_{\pm 1.2}$
	25	$89.2_{\pm 0.6}$	$93.4_{\pm 1.0}$	55	$87.6_{\pm 0.3}$	$90.4_{\pm 0.6}$	25	$72.7_{\pm 0.1}$	$85.2_{\pm 0.8}$
	40	$91.0_{\pm 0.4}$	$93.8_{\pm 0.4}$	70	$89.4_{\pm 0.4}$	$90.7_{\pm 0.2}$	40	$74.0_{\pm 0.2}$	$85.4_{\pm 0.2}$
	55	**91.5**$_{\pm 0.3}$	$93.8_{\pm 0.1}$	85	**89.7**$_{\pm 0.1}$	$90.5_{\pm 0.4}$	55	$74.0_{\pm 0.2}$	$85.9_{\pm 0.5}$
	70	$90.7_{\pm 0.3}$	$94.0_{\pm 0.3}$	100	$89.7_{\pm 0.2}$	$90.5_{\pm 0.2}$	70	**74.1**$_{\pm 0.2}$	$85.8_{\pm 0.2}$

choices versus the binary attention annotations. To measure the annotation consistency—whether a single receives divergent labels—we use the percentage of the most frequent label as a consensus measure. A higher percentage suggests greater consistency. Our findings indicate that attention correctness annotations exhibit a notably higher consistency, suggesting that it is less ambiguous to determine correct attention than to identify spurious features.

Attention Consistency. *SLIM* requires spreading labels in a space where attention patterns are consistently similar in nearby areas. To evaluate its feasibility, we compared how similar neighbors are in the original feature space versus the attribution representation space created by *SLIM*. We used k-means clustering with varying numbers of clusters to measure the intra-cluster similarity for both the model's attention matrices and the feature vectors. A higher similarity score means

Table 3.11: Performance variation with different training sizes. $SLIM_{Tr}$ and $SLIM_{Val}$ are trained on different proportions (Prop) of training sets and validation sets, respectively.

Method	Waterbirds			CelebA			ISIC		
	Prop%	Worst	Avg	Prop%	Worst	Avg	Prop%	Worst	Avg
$SLIM_{Tr}$	5	$82.4_{\pm 1.0}$	$83.6_{\pm 1.3}$	5	$88.9_{\pm 0.1}$	$90.0_{\pm 0.1}$	15	$77.6_{\pm 2.1}$	$87.1_{\pm 1.2}$
	10	$\mathbf{89.1}_{\pm 0.6}$	$91.7_{\pm 0.5}$	10	$\mathbf{89.4}_{\pm 0.3}$	$90.1_{\pm 0.2}$	20	$\mathbf{79.1}_{\pm 0.5}$	$88.5_{\pm 0.4}$
	12	$82.3_{\pm 1.3}$	$87.8_{\pm 0.3}$	15	$89.4_{\pm 0.4}$	$91.0_{\pm 0.4}$	25	$77.4_{\pm 1.5}$	$88.2_{\pm 1.2}$
	15	$83.5_{\pm 2.7}$	$89.6_{\pm 0.5}$	20	$88.9_{\pm 0.1}$	$90.8_{\pm 0.3}$	30	$78.3_{\pm 1.5}$	$88.3_{\pm 0.9}$
$SLIM_{Val}$	5	$84.2_{\pm 3.6}$	$89.2_{\pm 2.6}$	3	$85.7_{\pm 1.3}$	$88.5_{\pm 1.1}$	25	$68.2_{\pm 2.1}$	$81.9_{\pm 0.8}$
	10	$\mathbf{91.0}_{\pm 0.4}$	$93.8_{\pm 0.4}$	5	$\mathbf{89.4}_{\pm 0.4}$	$90.7_{\pm 0.2}$	30	$\mathbf{74.0}_{\pm 0.2}$	$85.4_{\pm 0.2}$
	12	$87.6_{\pm 0.6}$	$93.9_{\pm 1.3}$	10	$88.3_{\pm 1.4}$	$90.3_{\pm 0.5}$	35	$72.5_{\pm 0.5}$	$85.3_{\pm 0.1}$
	15	$86.8_{\pm 2.0}$	$94.6_{\pm 0.5}$	15	$87.2_{\pm 0.4}$	$90.3_{\pm 0.4}$	40	$70.9_{\pm 1.2}$	$85.4_{\pm 3.5}$

the space has more consistent patterns, which is essential for the accurate spreading of labels. As Fig. 3.15 shows, our attribution representation space significantly improves consistency in the model's attention, while also maintaining a good level of data feature consistency. Increasing the number of clusters generally increases consistency, but there is a point where adding more clusters does not lead to further gains, as shown by the curve flattening at k=120 in the left plot.

In summary, we have demonstrated that the annotation module of *SLIM*is less time-consuming and yields more consistent labeling feedback. Additionally, the attribution representation space engineered by *SLIM* establishes a solid foundation for propagating attention labels, considering the consistent model attention observed among neighboring instances.

3.4.10 Ablation Study (Q3)

Influence of data amount for attention annotations. We change the data amount for attention annotations to investigate their effect on performance, keeping other variables constant. Table 3.10 illustrates that both the worst-group and average accuracy tend to rise with more annotations, while such incremental benefit stabilizes after reaching a certain point, helping us identify an optimal number of annotations that balance performance with annotation cost.

Influence of training set size. This size refers to the size of the data subset constructed in *SLIM*'s

final phase for training a robust model. By only varying this parameter (Table 3.11), we find that the optimal size is inherently linked to the size of our constructed subgroups, $|g_{\hat{c}_i,e_i}|$. An optimal size is reached when data from various groups are sufficiently sampled without redundant over-sampling. These findings guide the hyperparameter selection in this phase, ensuring an effective cost allocation.

3.5 Conclusion

This chapter presents human-aided analytical frameworks for validating and improving machine learning systems. We first introduce *AttributionScanner*, a visual analytics system that enables metadata-free data slice discovery to systematically reveal model issues such as spurious correlations and mislabeled data through interpretable, human-in-the-loop analysis. Building upon these insights, we further present *SLIM*, a data-quality-oriented framework that mitigates spurious correlations using minimal human annotations by constructing feature-balanced training subsets. Together, these approaches demonstrate that structured human involvement can effectively support both the discovery and mitigation of model failures, highlighting the potential of human-AI collaboration and data-centric strategies for developing more reliable and trustworthy AI systems.

Chapter 4

Validating Multimodal Data Quality through Visual Analytics

Building upon the previous chapters that focus on understanding model decisions and validating model behavior, this chapter investigates the problem of validating data quality in modern multimodal AI systems. Recent advances in multimodal foundation models (FMs), such as CLIP and LLaVA, have enabled large-scale automatic data annotation, greatly expanding the availability of labeled data for downstream tasks such as open-vocabulary object detection and segmentation. However, while existing research largely emphasizes increasing data quantity, the quality of FM-generated labels remains relatively underexplored. In practice, validating large volumes of automatically labeled data without ground-truth annotations is highly challenging. Existing approaches often rely on limited statistical metrics or manual inspection of a small subset of samples, making it difficult to comprehensively diagnose systematic issues in large multimodal datasets. To address this challenge, this chapter presents *VISTA*, a visual analytics framework for scalable multimodal data validation. By integrating multi-stage data validation strategies with human-centered visual analysis, the framework enables users to identify, interpret, and correct hidden issues in FM-generated labels. Case studies on benchmark datasets and expert evaluations demonstrate the

effectiveness of the proposed approach in improving the reliability of multimodal data used in downstream AI systems.

4.1 Introduction

Recent advances in foundation models (FMs) have significantly expanded the capabilities of modern AI systems, enabling them to interpret complex visual and linguistic environments [10, 35]. In particular, multimodal foundation models can automatically generate textual descriptions for large collections of visual data, leading to a growing trend of incorporating FMs into data pipelines to increase the scale and diversity of training datasets [91, 237]. Such data generation pipelines support a wide range of vision-language tasks, including image captioning [238], phrase grounding [237], and OVS [91]. In these settings, models such as SAM [10], LLaVA [238], and CLIP [35] are commonly used to generate image-text annotations at large scale.

While these approaches substantially increase the quantity of training data [237, 238], the quality of automatically generated annotations has received comparatively less attention. Recent studies suggest that models trained on smaller but carefully curated datasets can outperform models trained on larger datasets of unverified quality [32, 33]. However, improving the quality of FM-generated multimodal data remains challenging. Such datasets are often produced through automated pipelines without verified ground truth, making systematic quality assessment difficult. Existing efforts typically fall into two categories: automated filtering based on model scores or dataset statistics [34, 35, 36, 32], and human-assisted inspection or correction [10, 33]. However, automated methods often capture only limited aspects of data quality, while human validation is typically applied to only a small subset of data due to the high cost of manual inspection. Consequently, these approaches struggle to provide comprehensive and scalable validation for large

multimodal datasets.

To address these challenges, this chapter presents *VISTA*, a visual analytics framework designed to support scalable validation and improvement of FM-generated multimodal data. The framework focuses on the challenging task of open-vocabulary image segmentation, where complex data generation pipelines introduce multiple potential sources of error. By combining automated quality analysis with interactive visual exploration, *VISTA* enables users to identify, interpret, and correct hidden issues in multimodal datasets. The system organizes potential data issues using multi-faceted quality metrics and provides visual interfaces that support efficient issue detection, validation, and correction through human-AI collaboration.

The main contributions of this chapter include:

- An automated issue detection and summarization approach that organizes multimodal data using multiple quality metrics, enabling large-scale identification of potential data issues.
- A visual analytics framework, *VISTA*, that integrates algorithmic analysis with interactive visualization to support human-centered validation and improvement of FM-generated data.
- Empirical evaluations through case studies and expert analysis demonstrating the effectiveness of the proposed framework in improving both the quality of multimodal datasets and the performance of downstream OVS models.

This chapter illustrates how visual analytics and human-centered data validation can complement automated data pipelines, enabling more reliable multimodal learning systems.

4.2 Related Work

4.2.1 Foundation Model-Involved Data Pipeline.

FMs' capability to understand the diverse multi-modal world facilitates their applications in auto-labeling, which automatically generates textual descriptions of images/videos [237, 238]. This has spurred many efforts to leverage FMs in data pipelines, such as general-purpose image-language understanding [238], referring between noun phrases and bounding boxes [237], open-vocabulary image segmentation [91]. While extending data volumes has contributed to performance improvement, there are increased quality concerns. For instance, Dai et al. [32] prove that improving the quality of only a small data amount significantly boosts image generation performance. Touvron et al. [33] demonstrates that high-quality data acquired by supervised fine-tuning (SFT) can notably improve model performance. However, the efficiency of data quality enhancement remains a persistent challenge—Data validated by humans has verified quality with considerable costs, thus existing work often opts for validating a sampled subset [33, 10]. On the other hand, automatic methods utilize quality metrics to filter out unsatisfactory data [34, 36], which may miss the big picture and overlook many crucial issues [193]. *VISTA* aims to fill the existing gap with effective human-AI collaboration—bypassing the disadvantages including high manual labeling costs and limited capability of static filtering.

4.2.2 Human-in-the-loop Data Labeling and Validation

Visual analytics (VA) plays a crucial role in labeling and validating expansive datasets [239, 240, 241, 50], providing data exploration capabilities to expedite the costly annotation process [242, 243]. VA tools for label creation develop visualization techniques [244, 245, 246, 247] or automated algorithms [248, 249] to facilitate label collection, which still demands substantial human

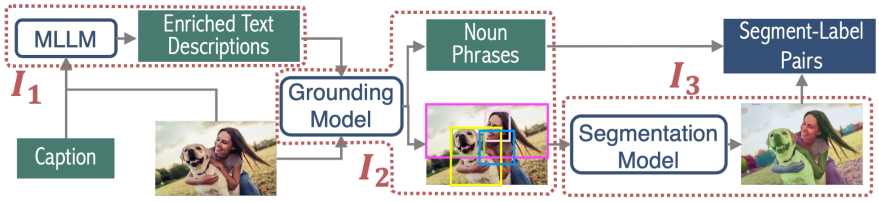


Figure 4.1: OVS data generation pipeline, involving FMs to construct segment-label pairs. Red outlines indicate potential stages with data quality concerns.

involvement or fail to handle more complex multi-modal data. Label or model validation systems assume labels have already been produced and focus on validating their quality [250, 188, 251, 252, 253, 254, 255, 256, 257, 258, 26, 48]. Techniques such as dimensionality reduction and uncertainty estimation have been employed to verify crowd-sourced image datasets and detect problematic samples [259, 260, 195, 28], which can be resource-intensive. Hoque et al. [261] introduced Visual Concept Programming that uses semantic segmentation models and allows experts to infuse domain knowledge, scaling up data labeling through automated functions. Therefore, facing multi-modal data that are often more complex due to varied modalities, heterogeneous in structure, and massive in volume, we introduce *VISTA* to efficiently address data quality concerns with humans in the loop.

4.2.3 Open-vocabulary Image Segmentation

OVS marks a significant evolution in autolabeling, enabling models to recognize fine-grained semantics in the dynamic real world. Two primary approaches define the current landscape of this domain. Training-based approaches [262, 263, 43, 264, 97, 265] use a web-scale dataset to train or fine-tune a sophisticated model, which can produce good results but are computationally intensive. The performance of these approaches is affected by multiple factors including model architectures, training data, loss functions, and optimization strategies [265, 43, 264, 97]. On the other hand, training-free OVS either explores the attention mechanisms of FMs [266, 267, 268] or constructs a

reference set to perform semantic segmentation via information retrieval [269, 270, 271, 272, 273], while retrieval-based methods provide a valuable evaluation for our framework, whose performance is mainly determined by the quality of reference sets.

4.3 Motivation and *VISTA* Framework

In this section, we overview the OVS data pipeline with FMs, discuss challenges and requirements for quality enhancement, and introduce *VISTA* framework. Note that *VISTA* is developed in close collaboration with two ML experts working in the industry.

4.3.1 OVS and Data Pipeline with FMs

Fig. 4.1 shows a typical data generation pipeline of OVS. A multi-modal large language model (MLLM) processes the input image to produce enriched text descriptions. A grounding model then links noun phrases from descriptions to image regions using bounding boxes, according to which a segmentation model crafts segment masks.

Data quality considerations. In Fig. 4.1, three outlined stages can potentially introduce quality issues, as detailed below.

*I*1. **Label issues.** The label generation process with MLLMs can produce contextually irrelevant labels (Fig. 4.1 (I_1)).

*I*2. **Alignment issues.** Grounding process pairs noun phrases with image regions (Fig. 4.1 (I_2)). However, contextual information can be mixed in its embeddings, leading to inaccurate alignments.

*I*3. **Image segmentation issues.** The image segmentation process (Fig. 4.1 (I_3)) may generate low-quality masks, such as those without accurate boundaries or meaningful content.

Our discussions with ML experts indicate that not all issues are equally significant. *I*3 have become less frequent due to technology advancements [10], while the other two remain signifi-

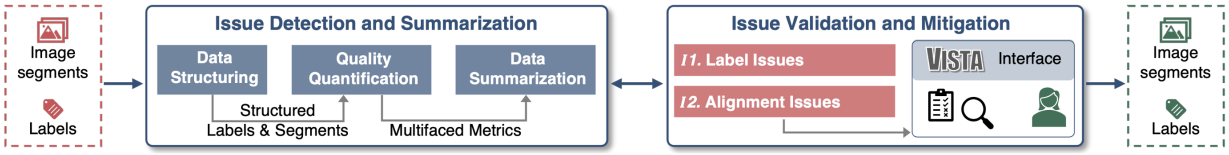


Figure 4.2: *VISTA* data quality enhancement framework, comprising the method phase for issue detection and summarization and the visual analytic interface for issue validation and mitigation. With *VISTA*, we enhance foundation model-generated data quality with human assistance.

cant contributors to data quality problems and require close attention. Consequently, our research focuses on addressing the most critical challenges related to *I*1 and *I*2.

4.3.2 Challenges for Quality Enhancement

C1. Lack of ground-truth data. As discussed in Sec. 4.1, FMs' involvement in data pipelines results in data without verifiable ground truth [34, 36, 10]. However, identifying wrong data is foundational for quality validation, and the absence of ground truth makes it challenging to spotlight problematic data, thereby impeding subsequent remediation efforts.

C2. Complexity of multi-modal information. While we categorize the main data concerns into three types, their interconnected nature prevents them from being viewed in isolation. However, tackling them simultaneously is also not feasible. Such complexity of the multi-modal landscape poses additional challenges, leading to many manual efforts in prior works [10, 33], thereby necessitating a nuanced approach to properly disentangling the underlying problems.

C3. Large volume of noisy data. Significant noise may be present in the large volume of FM-generated data. Sorting through and enhancing them is a formidable task. For instance, considering human has open-world knowledge to validate data quality, the sheer amount of potential errors necessitates substantial effort, which is impractical without a strategically developed support system.

4.3.3 Requirements for Quality Enhancement

To overcome the identified challenges, combined with the expertise of our collaborated ML experts, we develop the essential requirements that our framework must meet:

R1. Issue detection from multiple perspectives. To facilitate human involvement (**C3**), we need an algorithmic strategy to discover data issues. Considering the lack of ground truth (**C1**), analyzing from multiple angles can provide a more comprehensive error coverage. Moreover, the designed metrics should align closely with the data generation (Fig. 4.1), where the issues are produced, and disentangle multi-modal data (**C2**), considering different issue types (*I*1, *I*2).

R2. Multimodel alignment analysis at various granularities. Analyzing image-text alignment is the cornerstone of multi-modal problems, inherently involving various image granularities (**C2**), i.e., "*image—segment's surrounding region (bounding box)—segment*". Despite introducing noise to visual embeddings, such granularities have positive contributions to our framework by providing additional relationships to analysis, which can serve as additional evaluation perspectives to address our lack of ground truth challenge (**C1**).

R3. Systematically organize, explain, and summarize data. Our framework should systematically organize, explain, and summarize multi-modal data to facilitate validators' understanding (**C2**, **C3**). Our framework should identify similar labels, segments, and misalignments, streamlining the detection of different types of issues. In addition, explaining the formed groups is essential to support understanding.

R4. Usability-driven design to support human involvement. Human expertise, while invaluable, is finite and can have declining performance, especially when exhausted. Our target users are ML experts with limited visualization/VA knowledge. Therefore, our interface design should pri-

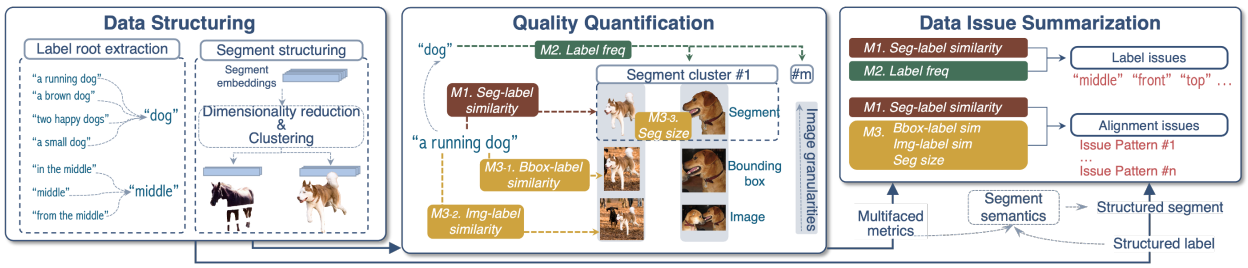


Figure 4.3: Method in *VISTA* for data issue detection and summarization, with three interlocked phases: label and segment data structuring, quality quantification for different issues, and data issue summarization for forming subgroups with coherent patterns or semantics.

oritize usability to avoid overwhelming users. Specifically, we should: (1) provide straightforward rather than sophisticated visualizations; (2) design interactive views to present related information simultaneously; (3) present a summary view to enable memory retrieval with a simple glance.

4.3.4 *VISTA* Framework

To address the challenges and requirements, we introduce *VISTA*, a framework to enhance FM-generated data quality. An overview of *VISTA* is shown in Fig. 4.2. Our issue detection and summarization uses statistical or computational strategies (**R1**) to deconstruct this complex problem into coherent subgroups with clear summarization (**R2**, **R3**). Grounded on the summarized issues, the visual analytics system enables human validators to quickly locate, understand, and solve different issues to enhance data quality (**R3**, **R4**).

4.4 Method

In this section, we discuss the method in *VISTA*;s issue detection and summarization backbone, including three phases as shown in Fig. 4.3. Driven by design requirements, the data structuring phase aims at organizing data from each modality into structured subgroups (**R3**). The quality quantification phase focuses on the multiple granularities of data alignments, measuring their quality with multifaced quantification metrics (**R1**, **R2**). Then, the data issue summarization phase

coordinates findings to further structure and explain data with concise semantics or coherent error patterns (**R1**, **R2**, **R3**).

4.4.1 Data Structuring

Structuring of labels. Recent research finds that labels with descriptions encapsulate richer semantics, and thus improve models' open-vocabulary recognition capability [274]. Therefore, noun phrases become the common type of textual label, in the format of *"description + noun"*. The involvement of FMs further increases their diversity, making it hard for effective exploration. To fulfill **R3**, we study this process and find the invariant here is the root of labels—the "*noun*". For example, for an image with dogs, we can have multiple labels like *"a running dog"*, *"pretty dogs"*, and *"a brown dog"*. Regardless of various descriptions, the root *"dog"* is unchanged. Therefore, by extracting the root from noun phrases, we categorize various labels according to their root and form consistent label groups, as depicted in the "label root extraction" process in Fig. 4.3. Specifically, we deploy SpaCy [275] for the root extraction, and record the mapping between each root and their corresponding labels for seamless information retrieval in our analytical framework.

Structuring of segments. Image segments embody a more complex context compared with labels, demanding a meticulous design for organizing them (**R3**). As shown in the "segment structuring" module in Fig. 4.3, our method involves a two-step clustering process, comprising the reduction of high-dimensional segment embeddings to a 2D space using UMAP [215] and the clustering with HDBSCAN [276] to identify clusters. The necessity for this intermediate dimensionality reduction arises from the challenges posed by the large amount and dimension of segments. Our experiments with ML experts indicate that directly applying HDBSCAN or other clustering methods on high-dimensional embeddings results in suboptimal clusters. Through results assessment with silhouette scores of different combinations of dimensionality reduction [277] and clustering [278], this two-

step process has emerged as the most effective strategy for our data format and scale, ensuring the creation of meaningful segment groups.

4.4.2 Quality Quantification

Let I denote an image, S denote an image segment, B denote the bounding box of the segment, and L denote a textual label describing the segment, with root r. A multi-modal model, such as *clip_surgery* [279], including an image encoder and a text encoder, encodes them into embeddings with the same dimension, $\boldsymbol{i}$, $\boldsymbol{s}$, $\boldsymbol{b}$, $\boldsymbol{l} \in \mathbb{R}^n$, respectively. Besides, we use $\frac{\langle x,y \rangle}{\|x\| \cdot \|y\|}$ to represent cosine similarity between x and y. Now, we delve into our designed data quality metrics.

M1 *Segment-label similarity*

This metric quantifies the alignment between segment and label embeddings, serving as a primary data issue indicator. It is the cosine similarity between $\boldsymbol{s}$ and $\boldsymbol{l}$, represented by:

$$sim(\boldsymbol{s}, \boldsymbol{l}) = \frac{\langle \boldsymbol{s}, \boldsymbol{l} \rangle}{\|\boldsymbol{s}\| \cdot \|\boldsymbol{l}\|}, \tag{4.1}$$

where a higher $sim(\boldsymbol{s}, \boldsymbol{l})$ indicates a stronger correlation.

M2 *Label frequency*

Building upon the data structuring established in Sec. 4.4.1, we use $C(r)$ to represent the count of unique segment groups to which a unique root r is assigned. This metric supports the detection of label issues (*I*1), represented by:

$$freq(l) = C(root(l)) = C(r). \tag{4.2}$$

Labels with the same root should not be capable of describing numerous varieties of segments. Therefore, a higher $freq(l)$ is correlated with ambiguous terms or other potential mistakes.

M3 *Multi-granularity image-label alignment*

This metric quantifies alignment between image and text (*I2*), with a specific focus on image granularities. As discussed in *I2*, additional context information outside a segmentation mask can be mixed in segment embeddings, resulting in misalignment where a label is paired with the spatial context of a segment. There are three sub-metrics in this fold:

$$sim(\boldsymbol{b},\boldsymbol{l}) = \frac{\langle \boldsymbol{b},\boldsymbol{l} \rangle}{\|\boldsymbol{b}\| \cdot \|\boldsymbol{l}\|}; sim(\boldsymbol{i},\boldsymbol{l}) = \frac{\langle \boldsymbol{i},\boldsymbol{l} \rangle}{\|\boldsymbol{i}\| \cdot \|\boldsymbol{l}\|}; size(S) = \frac{w_S * h_S}{w_I * h_I}, \tag{4.3}$$

where the first and second are *Bounding box-label similarity* and *Image-label similarity*; the third is *Segment size* that measures the proportion of an image spanned by a segment; w_* and f_* represent the width and height in the image space. Specifically, w_S and h_S are defined as the dimensions of the smallest bounding box that can fully enclose the segment S.

In the quality quantification phase in Fig. 4.3, we depict how these metrics assess different perspectives of this pairing relationship in multi-modality (**R1**, **R2**).

4.4.3 Data Issue Summarization

In this phase, our goal is to further sort through data, providing concise explanations for coherent groups and identifying consistent issue patterns from our initial findings (**R3**), making the information ready to be presented at our visual analytics interface (**R4**). Aligned with previous works [35, 34], we leverage segment-label similarity (*M1*) as the foundation to identify problematic data, as it represents the kernel of this pairing problem. Additionally, with the consideration discussed in **R1**, other data issue perspectives are not overlooked.

4.4.3.1 Segment semantics discovery

In Sec. 4.4.1, we have structured segments and labels. However, unlike labels in natural language, the interpretability of image information is not as direct [280]. To offer quick insights to human

Algorithm 4 Discovering semantics for a segment cluster

Input: Segment set $\boldsymbol{S}$, set of the corresponding labels $\boldsymbol{L}$, segment and label embeddings $\boldsymbol{s}, \boldsymbol{l} \in \mathbb{R}^n$, segment 2D embeddings $\boldsymbol{c_s} \in \mathbb{R}^2$.
Output: Concise semantics of the segment cluster $\boldsymbol{S}$.

1: **for** each $S_i \in \boldsymbol{S}$ and $L_j \in \boldsymbol{L}$ **do**
2: $sim(\boldsymbol{s}_i, \boldsymbol{l}_j) = \frac{\langle \boldsymbol{s}_i, \boldsymbol{l}_j \rangle}{\|\boldsymbol{s}_i\| \cdot \|\boldsymbol{l}_j\|}$ // Cosine similarity
3: $kde(\boldsymbol{c}_{\boldsymbol{s}_i}) = \frac{1}{Nh^2} \sum_{j=1}^{N} K\left(\frac{\boldsymbol{c}_{\boldsymbol{s}_i} - \boldsymbol{c}_{\boldsymbol{s}_j}}{h}\right)$ // KDE
4: $r \leftarrow [\,]$ // Initialize ranking list
5: **for** each segment-label pair $(S_i, L_j) \in (\boldsymbol{S}, \boldsymbol{L})$ **do**
6: $score \leftarrow kde(\boldsymbol{c}_{\boldsymbol{s}_i}) \times sim(\boldsymbol{s}_i, \boldsymbol{l}_j)$ // ranking score
7: r.append$((S_i, L_j, score))$
8: $count = \lceil p\% \cdot |r| \rceil$ // top $p\%$
9: $L_{top} \leftarrow r[argsort(-r.score)[:count]].L$ // labels in top $p\%$
10: **return** $L_{top}.root.unique()$ // unique roots

validators into the key content of each segment group, we endow them with semantic descriptions through our semantics discovery derived through Alg. 4.

Given a clustered set of segments $\boldsymbol{S}$, and the set of corresponding labels for each segment $\boldsymbol{L}$. Each segment-label pair has a similarity quantification *M1* (line 2 of Alg. 4). To avoid the impact of clusters' outliers, we focus more on segments densely located within the cluster according to Kernel Density Estimation (KDE), representing the density of neighboring segments within the 2-dimensional embedding space (line 3 of Alg. 4). The product of the segment-label similarity and the KDE value for each segment yields a ranking score, identifying top similar labels to the central of densely clustered segments, i.e., the cluster's semantics (**R3**).

4.4.3.2 Label issue summarization

To spotlight the most dominant label issues that happen to groups of labels (i.e., *root* (**R4**), we leverage segment-label similarity (*M1*) and label frequency (*M2*) in this step (**R1**). *M2* is computed for each label group while *M1* is based on a single segment-label pair. To make both describe group-level issues, per the recorded mapping between root and labels (refer to Sec. 4.4.1), we ob-

tain the median of segment-label similarities corresponding to each root. Therefore, for each label group (root), we have two metrics, representing its problematic level (*M1*) and whether they are sketchy or ambiguous to be assigned to a wide variety of segments (*M2*), respectively. To provide a holistic analysis of this focal issue, we decided to employ a dual-metric approach in our interface without merging them, which will be discussed in detail in Sec. 4.5.2.

4.4.3.3 Alignment issue summarization

The alignment issue (*I2*) concurrently elucidated by the segment-label similarity (*M1*) and multi-granularity image-label alignment (*M3*) (**R1**). Specifically, *M3* comprises three sub-metrics for a thorough validation across various granularities (**R2**). Nevertheless, integrating these sub-metrics with the segment-label similarity poses a challenge—How can we distill them into a clear summary, enabling straightforward error pattern identification?

To tackle this challenge, we employ DivExplorer [82], a tool designed to find common attributions associated with particular outcomes. For each segment-label pair, we regard *M1* as the indicator of misalignment. Consistent with common set-ups [34, 36], we set its global median as the default value of the threshold—If the value of *M1* falls below it, the pair is flagged for misalignment. We then convert the continuous values of each sub-metric in *M3* into three discrete categories with the same size following common practices—based on their percentiles [82, 79, 281], 33%, 66% are used in our case. Note that such thresholds are adjustable by users. DivExplorer leverages binary misalignment indicator (outcome) and categorized metrics (attributions) to identify subgroups exhibiting consistent error patterns as per *M3*. This algorithmic approach offers a solid foundation for human validators to pinpoint and comprehend the primary misalignment patterns (**R4**).

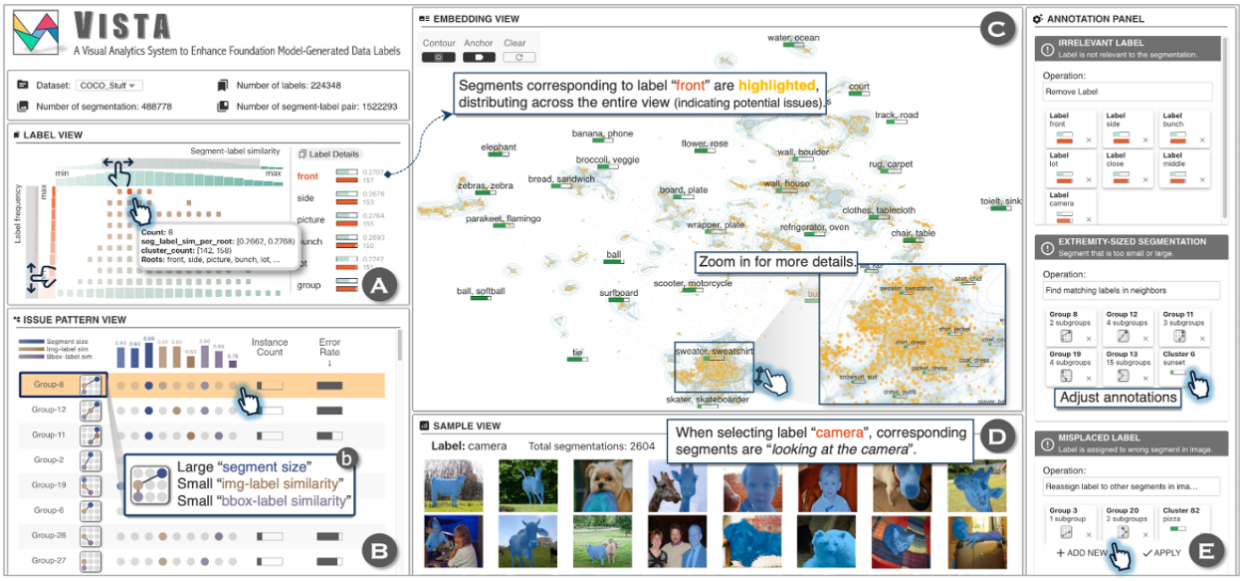


Figure 4.4: The system interface of *VISTA* supports the quality enhancement of foundation-model generated labels, integrating: Ⓐ *Label View* presents groups of labels in a grid-based layout, allowing for label issue validation; Ⓑ *Issue Pattern View* organizes segment-label pairs into data groups based on their common patterns, streamlining the identification and handling of alignment issues; Ⓒ *Embedding View* is a 2D projection of segment embeddings where users can browse segment clusters or data instances; Ⓓ *Sample View* provides a closer look at the selected samples, allowing for in-depth analysis; Ⓔ *Annotation Panel* is a utility pane enabling human validators to annotate and correct identified issues, which also provides a summary of the annotation status.

4.4.4 Online Evaluation

To quantitatively evaluate the capability of *VISTA* in enhancing the quality of datasets with segment-label pairs, we adopt a training-free OVS method [282] that performs semantic segmentation by retrieving a reference set with segment-label pairs, enabling real-time evaluation of data quality. Specifically, we compare the OVS performance when using the original and *VISTA*-produced data, respectively.

Let (S_{train}, L_{train}) denote segment-label pairs generated by FMs or enhanced by *VISTA*, with N unique segments. (S_{test}, L_{test}) be the ground-truth segment-label pairs in the test set, with M labels. Firstly, we find similar segments and similar labels between $test$ and $train$ using similarity quantifications sim_S and sim_L. We obtain the similarity between S_{test} and S_{train} by:

$$sim_S = softmax((\boldsymbol{S}_{test} \cdot \boldsymbol{S}_{train}^T) \times \beta_1), \tag{4.4}$$

where $\boldsymbol{S}_{test}$, $\boldsymbol{S}_{train}$ are embeddings of S_{test}, S_{train}; $sim_S \in \mathbb{R}^{1\times N}$. For all unique labels $l_{train} \in L_{train}$, we obtain their similarity with L_{test} labels by:

$$sim_L^{(i)} = softmax((\frac{\boldsymbol{o}_{train}^i}{\|\boldsymbol{o}_{train}^i\|} \cdot \boldsymbol{L}_{train} \cdot \boldsymbol{L}_{test}^T) \times \beta_2), i \in (0,N), \tag{4.5}$$

where $\boldsymbol{o}_{train}^i$ is the binary vector of labels for the i-th segment s_{train}^i, $\boldsymbol{L}_{train}$, $\boldsymbol{L}_{test}$ are embeddings of unique L_{train}, L_{test}, respectively. We have $sim_L \in \mathbb{R}^{N\times M}$. Note that β_1 and β_2 are temperatures for softmax, with a lower value considering more neighbors and a higher value leading to the focus on the top similar ones. An ablation study of parameters β_1 and β_2 is provided in the supplementary. Lastly, the predicted labels for s_{test} is computed by:

$$\hat{l}_{test} = argmax_{l_{test}}(sim_S \times sim_L). \tag{4.6}$$

This way, we obtain OVS results by this referring process. Consequently, we can measure its performance with mIoU and pixel accuracy, the common OVS validation metrics, by:

$$mIoU = \frac{1}{N}\sum_{i=1}^{N} \frac{Area\ of\ Overlap}{Area\ of\ Union} \tag{4.7}$$

$$pixel_acc = \frac{\#\ of\ Correctly\ Classified\ Pixels}{Total\ \#\ of\ Pixels} \tag{4.8}$$

4.5 Visual Interface

4.5.1 Interface Overview

VISTA's interface, as shown in Fig. 4.4, is designed data quality enhancement and consists of five coordinated views:

Ⓐ ***Label View***: This view facilitates the interactive exploration of summarized label issues (*I1*), helping users identify problematic label groups. As shown in Fig. 4.4 Ⓐ, we design a grid visualization showing the 2-dimensional information provided by segment-label similarity (*M1*, x-axis) and label frequency (*M2*, y-axis), the two key metrics for pinpointing crucial label issues (refer to Sec. 4.4.3.2) (**R1**, **R3**). Each rectangular cell represents one or more label groups with size indicating the group count, while as smaller (*M1*) and larger (*M2*) indicate more troublesome and dominant labels, they are spotlighted by cells closer to the top left of this label grid (**R3**, **R4**). Validators can brush histograms at each axis to select and check a label group (represented by the root, such as "front" in Fig. 4.4) to investigate further information.

Ⓑ ***Issue Pattern View***: Presenting summarized alignment issues (*I2*), this view allows users to validate predominant error patterns in the alignment between image segment and labels, corresponding to *M1* and *M3* (**R2**, **R3**). As shown in Fig. 4.4 Ⓑ, we visualize alignment issue summarization (refer to Sec. 4.4.3.3) with an UpSet visualization [283]. The discretized sub-metrics of *M3* are integrated with a matrix layout with nine columns, with the top bars showing the frequency of each attribution. For example, as shown in Fig. 4.4, 0.93 of "small segment size" means 93% of samples with "small segment size" are flagged as "wrong labels". Each row represents one data group with coherent issues, with the corresponding attribution highlighted with colored circles. As *M1* is a misalignment (error) indicator, we visualize the calculated "error rate" as horizontal bars for each data group.

With the UpSet visualization, we guide validators to explore and detect predominant misalignment patterns by making all columns rankable to prioritize more problematic groups (**R4**). In addition, we further observe that the matrix layout is insufficient for quick pattern identification—

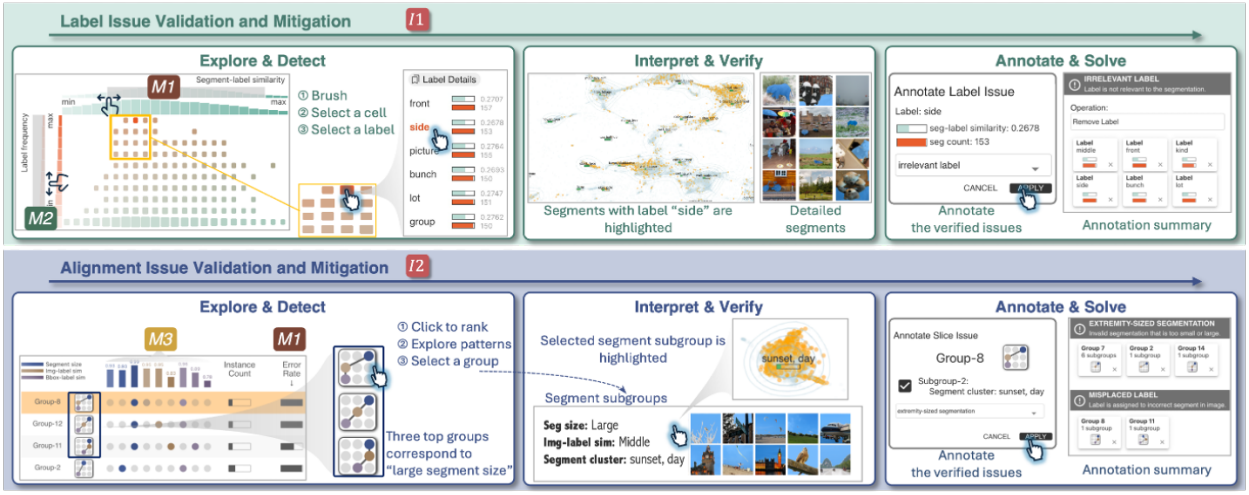


Figure 4.5: Analytical workflow for human validators to conduct issue validation and mitigation with *VISTA*'s visual interface, with the top and bottom phases addressing *I*1 and *I*2, respectively. Each phase comprises three stages. Humans interact with the *Label View* or the *Issue Pattern View* to *"Explore & Detect"* issues; *"Interpret & Verify"* errors by inspecting corresponding segment distributions or data samples through *Embedding View* and *Sample View*; and lastly, *"Annotate & Solve"* verified issue with the *Annotation Panel*, which summarizes annotation records under different categories.

users may need interactions to confirm related attributions. To facilitate user interpretation, we introduce the "Pattern Dice" as displayed in Fig. 4.4 (b), which abstracts a group's error pattern into a 3x3 grid of dots, where each row corresponds to one sub-metric of *M3* and each column from left to right corresponds to a discretized value range. This ingenious summary provides a distilled visual cue, ensuring validators can swiftly identify issues for debugging (**R4**).

(C) ***Embedding View***: This view provides the 2D distribution of image segments, generated by the segment structuring as discussed in Sec. 4.4.1. As shown in Fig. 4.4 (C), each point represents one image segment, and the textual semantics of segment groups are displayed on top of each group center, providing straightforward information about the key content of each group (**R3**), with a horizontal bar indicating the group's overall segment-label similarity (*M1*). Despite addressing *I*3 is not the main focus of *VISTA*, in the situation where users conduct in-depth validation, this view also allows users to investigate segment issues by highlighting, checking, and annotating single

or groups of segments with issues (**R1**, **R4**). In response to selections in other views, this view highlights the related segments simultaneously, showing complementary information to assist in the interpretation of data issues (**R4**).

Ⓓ ***Sample View***: In concert with Ⓒ, this view updates in sync with the other views to show images of corresponding segments, which coordinates with Ⓒ for highlighting distributions during users' interaction (Fig. 4.4 Ⓓ). This view allows for in-depth and intuitive exploration, supporting the verification of issues with direct examples (**R4**).

Ⓔ ***Annotation Panel***: As shown in Fig. 4.4 Ⓔ, this view provides a semantic summarization of each annotated issue and offers a suite of tools for issue mitigation, including different annotation options and solutions, buttons for adding new options, shortcuts for the annotation history, and options for modifying previous annotations (**R4**). It enables validators to seamlessly interact with the dataset, enhancing its quality and ensuring efficient correction of errors. Note that once an issue has been annotated, the corresponding label or group will be displayed in gray as an indication, making it straightforward for human validators to identify unexplored data issues.

4.5.2 Analytical Workflow

The design of *VISTA*'s interface consolidates a user-centric analytical workflow for efficacy-driven data issue validation and mitigation (**R4**), where each view works coordinately to integrate human expertise in resolving critical label (*I*1) and alignment issues (*I*2). As illustrated in Fig. 4.5, for addressing issues fall into each category (*I*1 or *I*2), our workflow is structured into three stages: *Explore & Detect*, *Interpret & Verify*, and *Annotate & Solve*, each meticulously crafted to guide users through the data quality enhancement process. In the *Explore & Detect* stage, validators investigate the label grid or UpSet visualization to identify issues requiring validation. Progressing

to the *Interpret & Verify* stage, following users' selection, our interface displays corresponding distribution overviews or in-depth details, providing the context needed to make informed decisions. Upon verification of an issue, the *Annotate & Solve* stage provides intuitive interfaces for annotating them and proposing resolutions, ensuring a thorough and effective data refinement. Following these stages, validators can tackle different types of issues systematically, thereby upholding the integrity of the data throughout the validation and mitigation process. Next, we delve into the specifics of how *VISTA* assists validators in resolving label issues and alignment issues, respectively.

Label Issue Validation and Mitigation. (*I*1)

As depicted in the top phase at Fig. 4.5, the workflow starts from the *Explore & Detect* stage with the *Label View*, where human validators can brush the histograms to zoom in specific regions, select a potentially wrong cell, and then investigate the label problems grouped by each root. Upon selection, our visual guidance will lead users to the *Interpret & Verify* stage, where the *Embedding View* and *Sample View* update segment-corresponded points and example images for the verification of flagged issues. Then at the *Annotate & Solve* stage, solutions can be provided with a few clicks, which will be automatically recorded and presented in the *Annotation Panel*. Finally, when validators confirm the annotated issues and solutions, they are applied to the data set for quality enhancement.

Alignment Issue Validation and Mitigation. (*I*2)

As depicted in the bottom phase at Fig. 4.5, the workflow starts from the *Explore & Detect* stage with the *Issue Pattern View*, where human validators can sort data groups according to specific attribute, explore their patterns, and select one group for detailed inspection. When a group is selected, in the *Interpret & Verify* stage, *Embedding View* and *Sample View* will be updated accord-

ingly, showing semantic descriptions of the selected data group and highlighting their distributions, which support detailed investigation of individual subgroups or segment-label pairs to verify data issues. Similarly, at the *Annotate & Solve* stage, the verified alignment issues can be annotated and applied following validators' confirmation.

4.5.3 Data Issue Resolution

When human validators confirm and apply their annotations at the end of the analytical workflow, *VISTA* meticulously documents them to apply human-proposed solutions for data issue correction. In our pursuit to improve the quality of data generated by FMs, we consciously refrain from utilizing these models at the error-addressing stage. Instead, we have collaborated with ML experts to devise three principal remedial strategies: (1) removing irrelevant labels, (2) removing problematic segments, and (3) seeking alternative matching segments for misaligned labels. These are provided in our interface as candidate solutions. If humans select from these categories, they are automatically executed to rectify the specified issues. In situations where the solutions fall outside, our specialists step in to review and address them accordingly.

As a prototype framework for addressing this challenging problem, our emphasis is on exploring the efficacy of human intervention and validating its significance in data quality enhancement. This approach allows us to establish a closed-loop system with manageable solutions, underscoring the merit of human contribution. While our focus is on lightweight interventions, we remain receptive to the adoption of more advanced techniques in future iterations of our framework.

4.6 Evaluation

In this section, we demonstrate the effectiveness of *VISTA* with two use cases and an expert study. With two datasets in the general and autonomous driving domains, respectively, in use cases, we

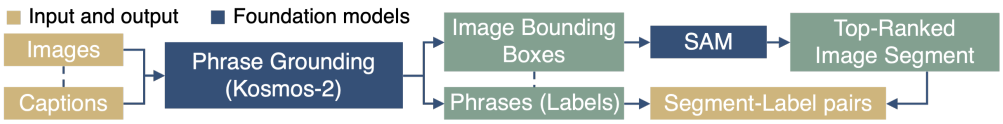


Figure 4.6: Data pipeline for getting segment-label pairs in our use cases.

demonstrate how *VISTA* supports users to conduct data enhancement in different scenarios. Additionally, we invited ten external ML experts who have not seen or used *VISTA* before in our expert study to evaluate our solution.

Data Pipeline. In our study, a data generation pipeline was employed to acquire image segment-label data, as depicted in Fig. 4.6. The process begins with the Kosmos-2 performing phrase grounding on paired images and captions, which results in preliminary labels grounded with bounding boxes, prompting SAM to generate corresponding segment masks. This pipeline represents a typical approach for generating segment-label pairs with FMs, serving as the basis for OVS. With the generated data, we deploy our *VISTA* for two cases.

4.6.1 Use Case I: Comprehensive Scene Parsing

Alice is a senior-year PhD student studying machine learning. She is working on an OVS project with the COCO Stuff dataset [284] and is willing to explore how data quality impacts its performance. Therefore, she conducted a performance-driven analysis with *VISTA*, comparing the OVS performance based on the original and *VISTA* processed data.

Investigating the validity of labels. With the current data pipeline, labels are directly generated by the phase grounding model. Alice does not completely trust the model, doubting if the labels themselves are troublesome. Following the analytical workflow, Alice interacts with the grid visualization of labels in view Ⓐ of *VISTA*, as shown in Fig. 4.4. She brushed the top histogram to check labels with low segment-label similarity, where she found many cells corresponding to high label frequency, indicating both low-performed and dominant labels. By selecting them and

Table 4.1: OVS performance comparison when using FM-generated and *VISTA*-enhanced data.

Data	Use Case I		Use Case II	
	mIoU	*Pixel Acc*	*mIoU*	*Pixel Acc*
Original	32.39	52.01	35.55	32.63
Enhanced	41.18 (↑ 8.79)	62.37 (↑ 10.36)	49.90 (↑ 14.35)	44.83 (↑ 12.20)

investigating views C and D, Alice found many labels are not even nouns and should not be assigned to any segment, such as "front", which is assigned to 14,350 image segments. Somehow many such labels were extracted and paired with segments by the FMs. In addition, Alice noticed a meaningful noun, "camera", being spotlighted in a neighbor cell as well. By checking the corresponding segments, she found most of them have no camera included, with content ranging from animals to humans, but they are all "looking at the camera" (as shown in Fig. 4.4). After exploring neighboring cells, Alice verified 18 labels with similar issues, annotated them as "irrelevant labels", and decided to "remove label" (refer to Fig. 4.5 for an additional example). By annotating them with a few clicks, Alice moved to other views for further data validation.

Analyzing segments and label pairing errors. Given that the data originated from a singular pipeline rather than multiple resources, Alice speculated that there might be other recurring errors. Observing the view B *Issue Pattern View* (refer to Fig. 4.4), she first noticed the three groups of bars, where "large segment size" is the highest. She also found the "error rate" indicator and thought this was important for error detection. Alice sorted the data by "error rate" while grouping by "large segment size", and found three groups at the top (refer to Fig. 4.4), characterized by "large segment size" with "high error rate". By investigating further details with views D, she found "Group-8" with "large segment size", "small image-label similarity", and "small bounding box-label similarity" includes segments highlighting background information, where their labels are pretty noisy (refer to Fig. 4.4). In addition, upon examining the segment distributions in view

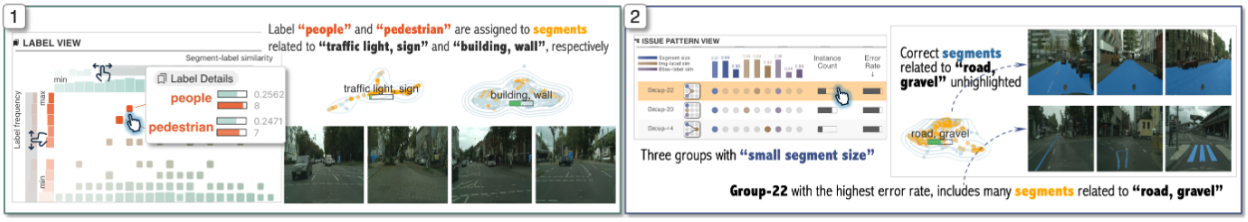


Figure 4.7: The use case of traffic scene understanding, where the user identified: (1) Labels "people", "pedestrian" mistakenly assigned to vague image segments related to "traffic light, sign" and "building, wall", respectively. (2) The problematic subgroup, "Group-22", had "small segment size" but included many "road, gravel" segments.

Ⓒ, she observed that the labels of the segments' neighbors, which were not highlighted, were accurate. Consequently, she marked this group's issue as "extremity-sized segmentation" and the solution as "find matching labels in neighbors". Additionally, for "Group-12" with "large segment size", "middle image-label similarity", and "low bounding box-label similarity", she found the segments frequently depicted background elements, with labels correctly describing the image but not the segments. For instance, an image of a bus station had its ground segment erroneously labeled as "a station", indicating an issue of "misplaced label". Alice designated the resolution to "re-assign label to other segments in image" for such cases. Continuing this methodical approach, Alice addressed other top-ranked groups, annotating and resolving verified issues.

Performance enhancement. In total, Alice annotated 27 problematic labels and 20 misaligned groups. By checking the report generated by *VISTA*, Alice was impressed that errors corresponding to 1.1M segment-label pairs have been fixed, which indicates a significant speed-up over manual data curation. After issue correction, Alice conducted the online evaluation(refer to Sec. 4.4.4) to compare OVS performance with *mIoU* and *pixel accuracy*. As shown in Tab. 4.1, the model demonstrated improved performance with *VISTA*-enhanced data, confirming the premise that high-quality data is essential for superior model performance.

4.6.2 Use Case II: Traffic Scene Understanding

Kevin is a researcher developing computer vision (CV) methods for autonomous vehicles. He is working on the Cityscapes dataset[285], which includes traffic scenarios with vehicles, pedestrians, and street signs. Kevin focused on discovering the hidden failures of auto-labeling, particularly those involving critical elements in the driving scene.

Pedestrian identification accuracy. Correctly identifying pedestrians is paramount for autonomous driving. As illustrated in Fig. 4.7, Kevin investigated the *Label View* to check whether there are problematic labels related to pedestrians. Sifting through the "segment-label similarity" histogram, he discovered that "people" and "pedestrian" lied in two neighboring cells, showing low similarity scores. Diving into the corresponding distributions in the *Embedding View*, he found frequent errors for assigning the label "people" to "traffic light, sign" segments, or assigning the label "pedestrian" to "building, wall" segments. Curious about the underlying causes, Kevin then checked related samples in the *Sample View* in detail, which led to significant insights—(1) Misalignments involving the label "people" often occurred with small, indistinct traffic signs or traffic sign-shaped segments; and (2) For the misalignment between the label "pedestrian" and the segments "building, wall", there were recurring situations where part of a building or wall is close to pedestrians. A common observation for both is the corresponding segments are vague, due to distance or lighting factors. After verification through the interface, Kevin categorized them as "vague segments" and "misplaced labels", respectively. Moreover, a review of neighboring cells at the *Label View* exposed that the label "person" was also associated with low "segment-label similarity" scores. This led Kevin to conclude that human-related segments were frequently susceptible to mislabeling, warranting a more thorough investigation.

Other common traffic labeling failures. Kevin proceeded to the *Issue Pattern View* to check the existence of other segment-label misalignment. He noticed that when ranking the groups decreasingly according to the error rate, many groups corresponding to "small segment size" stayed on the top. Initially, he thought this might correspond to the pedestrian identification issue due to the typically small segment size. However, by investigating the *Embedding View*, Kevin found a large portion of the segment cluster "road, gravel" was highlighted, which was an unexpected pattern, contradicting with the common sense that "road, gravel" should have large sizes. Upon examining the cases, Kevin found that the model only captured edges of roads or traffic lines rather than the complete road, yet labeled such edges as "road". He decided to visit the *Embedding View* to check the entire cluster with semantics "road", where he confirmed that for those not highlighted by our framework, the segments were correctly associated with the entire road. He annotated them with "correct labeling with road" and "wrong labeling with road", respectively.

Iterative data enhancement and performance gains. Because of the safety concerns, Kevin decided to conduct iterative data enhancement. After mitigating the detected errors, he loaded the updated data back to *VISTA* for the next-round data validation, repeating this process until reaching a satisfactory stage. Lastly, he validated the resulting data with *VISTA* and verified that they had the desired quality, supported by two observations: (1) the overall distribution of "segment-label similarity" in the *Label View* was much higher; (2) he had to increase the threshold to 0.8 to flag data with "potential issues", indicating a good alignment between segments and labels in general. By checking the automatically generated report, he found that *VISTA* helped him correct auto-labeling issues of $15k$ samples. He also conducted the quantitative evaluation to compare training-free OVS performance with the original and enhanced data as shown in Tab. 4.1, which confirmed the efficacy of *VISTA* in boosting model performance. He collected the final annotation record and

decided to validate other auto-labeling technologies using *VISTA*.

4.6.3 Expert Study

As mentioned at the beginning of this section, ten external ML experts were invited to our study, referred to as $E_0 \dots E_9$.

Expert demographics. Before the study, We collected detailed demographics of each expert, to contextualize their feedback and ensure that they belong to our target user—ML experts who do not necessarily have visualization/VA knowledge. Among them, 4 are PhD students and 6 have obtained PhD degrees. All experts major in Computer Science for their PhD studies. We asked them to self-report their expertise in two areas on a scale of $[0, 10]$ (0 for "no knowledge" and 10 for "expert") and report the statistics as follows:

- Expertise in ML/CV: Md = 9.0, IQR = 1.7;
- Expertise in Visualization/VA: Md = 0.0, IQR = 1.0.

For topics they are mostly familiar with, all experts have either used or conducted research on FMs, with other research focuses spanned among: object detection/segmentation (E_0-E_4, E_8), generative AI (E_2, E_5-E_7, E_9), medical/biomedical topics (E_1, E_3), autonomous driving (E_0, E_4, E_6), and 3D reconstruction (E_1, E_2, E_5-E_9).

Study setup. Each expert was invited to join a one-on-one session, lasting around 60 minutes. In the session, we started with a 5-minute introduction of the background, tasks, and interface. Then experts were asked to explore *VISTA* with the public domain case, COCO Stuff, following a think-aloud protocol and exploring as many functionalities as possible. We recorded the study process and noted their comments, paying particular attention to how well the design requirements were met. At the end, we gathered each expert's feedback through a questionnaire with Likert-type

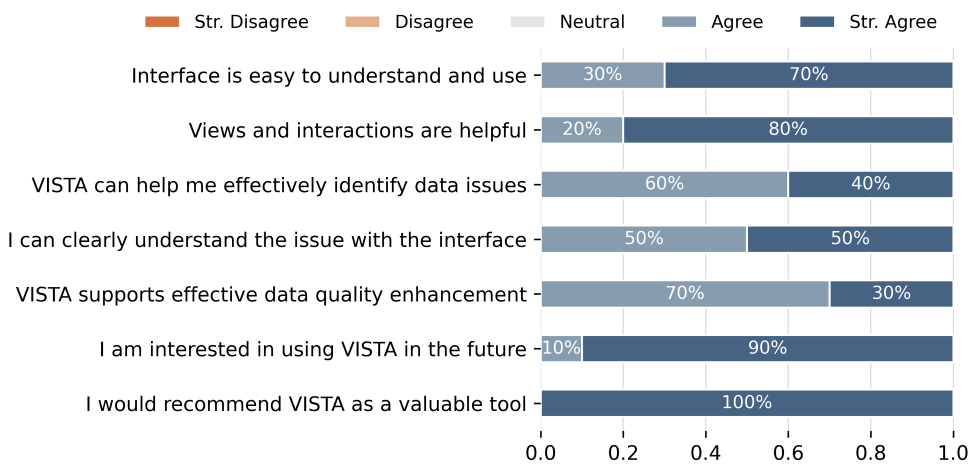


Figure 4.8: Expert perception of the system according to Likert-type questions.

questions and an interview.

Fig.4.8 provides the results of our post-questionnaire, which indicates all experts positively rated the usability and usefulness of *VISTA*. From the interview, we obtained more detailed feedback, and the summarization is provided below.

Detection of hidden errors (**R1**, **R2**). All experts successfully identified a range of data errors with *VISTA*. E_0 and E_1 commented positively on *VISTA* for its analytical depth in dissecting complex multi-modal data. E_1 appreciated the framework's contribution to improving model robustness, saying *"The key concern of adapting FMs in the medical domain is its uncontrolled nature and you helped mitigate this gap"*. E_0 shared the same view—*"The thorough validation from different angles increases my confidence in the data,"* highlighting the importance of multi-perspective data validation as *"crucial but often neglected."*

Interpretation and verification of data issues (**R3**). E_0 commented *VISTA* for providing sufficient information to make data issues interpretable. In addition, all experts agreed that our coordinated views facilitated their understanding, with E_1 and E_7 particularly enjoying the interactive feature. E_7 mentioned *"The interface simultaneously showed corresponding information upon my*

interactions. This greatly helped my task." $E9$ talked about the potential of *VISTA* for synthetic data, indicating that our *Embedding* and *Sample Views* can help them validate whether their generated data was realistic.

Usability of *VISTA* (R4). Following our brief introduction, all experts navigated the system smoothly. Regarding our designed analytical workflow, E_1, E_5, and E_8 described it as *"a clear guidance"*. E_0 mentioned the sorting function, *"it allows me to group various features for a closer examination."* E_2 and E_4 commended our "Pattern Dice" design as *"compact and helpful"*, and E_4 said *"it makes identifying patterns super easy."* We also noticed that although both E_5 and E_{10} haven't used any VA system, they promptly finished the data enhancement tasks, proving the usability of *VISTA*.

Other comments. The experts also offered constructive feedback. E_0 expressed interest in metric customization. While praising the system's capability in summarizing prevalent issues, she proposed that *"in later exploring stages, allowing users to define new metrics might uncover trivial problems that are less apparent."* E_9 discussed the possibilities of applying *VISTA* in natural language validation. "I would be interested to see how to validate LLM-generated data, " she mentioned. Despite some suggestions might be out-of-scope of this work, we plan to explore these invaluable comments in the future.

4.7 Conclusion

In this chapter, we presented *VISTA*, a visual analytics framework for improving the quality of multimodal data generated by foundation models, with a focus on the challenging task of open-vocabulary image segmentation. By integrating automated data analysis and human-aided visual exploration, our framework successfully supports users in detecting hidden data issues and then

performing targeted improvements with minimal effort. Empirical studies on benchmark datasets and expert evaluations demonstrate the framework's effectiveness in improving data quality and measurably enhancing downstream model performance. *VISTA* highlights the importance of data-centric validation in modern AI pipelines, which suggests that integrating automated analysis with human expertise through visual analytics provides a practical pathway toward building more reliable multimodal AI systems.

Chapter 5

Scaling AI Systems in Open-World Settings via Collaborative Models

Recent advances in foundation models have significantly expanded the capabilities of modern AI systems. However, extending these models to new concepts and tasks in open-world environments often requires costly model retraining or large-scale manual annotation. One representative problem in this context is OVS, which aims to segment images according to arbitrary textual categories without task-specific fine-tuning. Existing training-free approaches typically rely on attention mechanisms of pre-trained vision-language models such as CLIP or construct synthetic reference data with complex retrieval pipelines. Nevertheless, their performance is often limited by the intrinsic perception capability of the underlying models or the suboptimal quality of the reference sets used for retrieval. This chapter investigates the role of data quality in enabling training-free OVS and shows that a well-constructed reference set can substantially improve segmentation performance. Based on this observation, we develop a data-centric framework that constructs high-quality segment-text representations and performs similarity-based retrieval to support open-vocabulary segmentation without additional model training. Extensive experiments on multiple benchmark datasets demonstrate that improving the quality and structure of reference data pro-

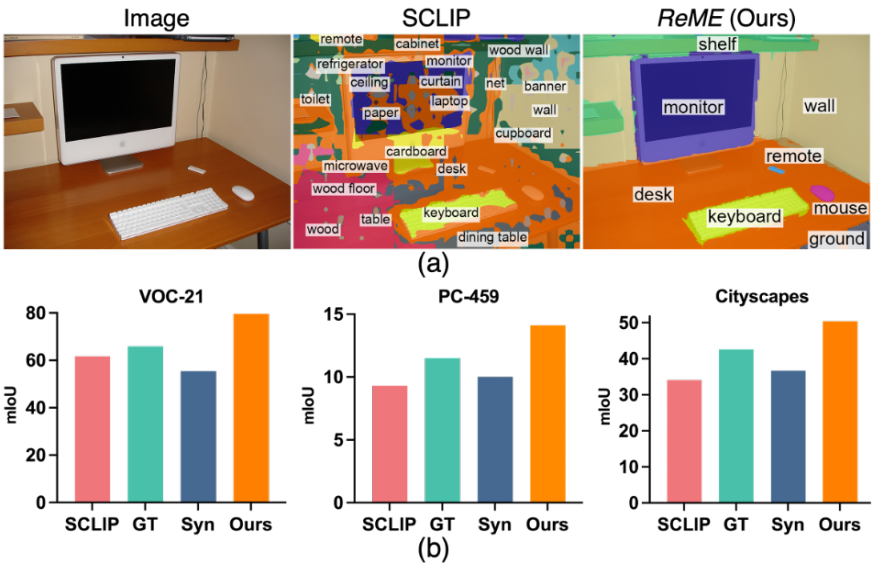


Figure 5.1: **(a)** Qualitatively compare *ReME* with SCLIP [286] (a representative CLIP-attention-based method) on an example from ADE20K [287] with 150 classes. Both methods have no mask post-processing. *ReME* produces higher-quality masks with a simple segmentation algorithm, yet SCLIP generates noisy masks paired with many irrelevant classes, such as *"curtain"* and *"microwave"*, etc. **(b)** Performance comparison between SCLIP and retrieval-based OVS using different reference sets. **GT**: ground-truth segment-label of COCO Stuff [288]; **Syn**: synthetic reference set from FreeDA [289]; **Ours**: our reference set from real images with quality refinement.

vides an effective pathway for extending the capability of vision-language models in open-world settings.

5.1 Introduction

OVS aims to segment images according to a set of arbitrary textual categories, enabling flexible scene understanding beyond predefined label spaces. This problem represents an important step toward extending the capabilities of visual recognition systems to operate in open-world environments. Conventional approaches to OVS typically rely on training or fine-tuning large models using densely annotated segment-text [97, 98, 99, 100, 101, 290, 291, 40] or image-text datasets [43, 292, 293, 44, 294, 295, 296, 45]. While effective, these approaches require substantial annotation effort and computational resources. Recent advances in vision-language models (VLMs), such as CLIP [35], have therefore inspired growing interest in *training-free* OVS meth-

ods that attempt to leverage pre-trained models without additional fine-tuning.

Many existing training-free approaches attempt to extend the pixel-level localization ability of VLMs by modifying their internal attention mechanisms [266, 268, 297, 298, 267, 299, 286]. These methods assume that VLMs already possess strong semantic recognition capabilities and that appropriate manipulation of attention maps can recover spatial localization information. However, VLMs are typically trained under weak image-text supervision [279], and post-hoc modifications of their attention mechanisms are fundamentally constrained by the coarse supervisory signals used during training [300]. As a result, such approaches often produce noisy segmentation masks and limited localization accuracy. For example, as illustrated in Fig. 5.1(a), attention-based methods can struggle to produce segmentation masks that match even the quality of classical class-agnostic segmentation algorithms [301], which generate perceptually coherent image regions despite lacking semantic supervision.

These limitations suggest that improving OVS performance may require complementary sources of information beyond the internal representations of VLMs alone. One promising direction arises from the *retrieval paradigm*, which augments model predictions by referencing external knowledge bases. From a data-centric perspective, this raises a natural question: if a high-quality reference set of segment-text pairs were available, could retrieval-based reasoning unlock stronger OVS capabilities without additional training?

To investigate this question, we first examine a simplified scenario in which ground-truth (GT) segment-text annotations from COCO Stuff [288] are used as a high-quality reference set. Using a class-agnostic segmentation algorithm [301] to generate candidate regions for test images, a simple similarity-based retrieval strategy (Sec. 5.3.3) can be applied to transfer semantic labels from the reference set. Notably, this basic retrieval procedure does not involve complex models or learning

algorithms. Nevertheless, as shown in Fig. 5.1(b), retrieval using GT annotations consistently surpasses SCLIP [286], a strong attention-based baseline. This observation highlights the significant potential of high-quality reference data for enhancing OVS performance.

In practice, however, constructing such reference sets using ground-truth annotations is extremely labor-intensive. To address this challenge, some recent works generate synthetic reference data using diffusion models (DMs) [302], producing image-text pairs or attention masks to populate retrieval databases [270, 289, 271, 273]. Although this strategy alleviates annotation costs, synthetic data often lacks the realism and diversity of real-world images [303, 304, 26]. As illustrated in Fig. 5.1(b), retrieval results based on synthetic datasets such as FreeDA [289] remain substantially weaker than those obtained from real annotated data. Other approaches attempt to compensate for data limitations by introducing more sophisticated retrieval algorithms [273], yet such solutions primarily address the symptoms of the data bottleneck rather than its underlying cause [28].

Motivated by these observations, this chapter investigates how high-quality multimodal data can be constructed to support training-free OVS. In particular, it explores the possibility of building a well-aligned and semantically rich reference set of segment-text pairs directly from real images, without relying on expensive manual annotations. To achieve this goal, the chapter introduces *ReME*, a data-centric framework that refines multi-modal embeddings to improve the quality of reference data for retrieval-based OVS.

The framework leverages an important observation about vision-language models: although individual predictions may be noisy, these models often exhibit useful patterns when analyzing data collectively [34, 305, 306, 307, 96, 195, 49]. Starting from images alone, the proposed data pipeline first constructs a semantically rich base set by pairing image segments with textual labels generated

by VLMs. By analyzing collective patterns in same-modal feature spaces, the framework then refines these initial pairings to reduce cross-modal misalignment introduced by VLM predictions. The resulting reference set provides improved semantic richness and alignment quality. During inference, class-agnostic segmentation is applied to test images, and the predicted regions are aligned with the refined reference set through a simple similarity-based retrieval process.

Experimental results across ten OVS benchmarks demonstrate that improving the quality of the reference set can substantially enhance training-free segmentation performance. These results highlight the importance of data-centric design for extending the capabilities of foundation models without costly model retraining.

The main contributions of this chapter are summarized as follows:

• A data-centric framework (*ReME*) that constructs high-quality segment-text representations from real images, enabling retrieval-based OVS without model fine-tuning.

• A multimodal data refinement pipeline that improves semantic richness and cross-modal alignment by leveraging collective patterns in intra-modal feature spaces.

• Extensive experiments across ten benchmark datasets demonstrating consistent improvements over existing training-free OVS methods.

5.2 Related Work

Training-Free OVS. Many training-free OVS methods analyze attention mechanisms [308, 48, 50] of VLMs or diffusion models to produce segmentation results [266, 267, 298, 268, 309, 310, 311, 312]. However, their reliant models trained without pixel-level supervision may produce inferior results, constraining the upper bound of these approaches. To improve the situation, methods such as CaR [297] leverage multiple CLIP models to iteratively refine segmentation results, which are

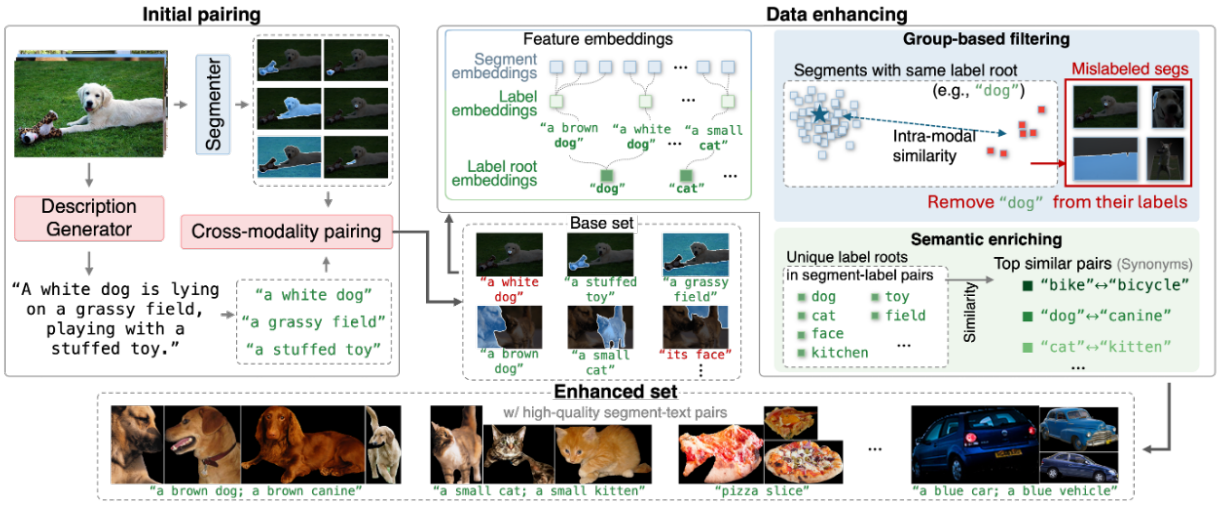


Figure 5.2: **The data pipeline of *ReME* to construct a high-quality reference set.** The pipeline includes two phases: ***initial pairing*** that produces pre-matched segment-text pairs using images as input; and ***data enhancing*** that performs group-based filtering according to more discriminative intra-modal similarity among segment embeddings and semantic enriching with similar labels.

less practical due to significant inference costs. Another line of methods [269, 270, 289, 271, 272, 273] constructs a reference set with extensive segment-text pairs, which serves as reference knowledge to support better alignment between test segments and text. Most of such methods either use synthetic data [270, 289, 271, 273] that falls short of diversity and realism [303, 304], or opt for sophisticated retrieval strategy or shortcut like adding GT classes to bypass the data quality issues [273, 313]. To fill the gap, our framework investigates the importance of data for this challenging task.

Multi-Modal Datasets Curation. High-quality datasets have driven remarkable progress in multi-modal models [91, 314, 315, 293, 237, 316, 31]. Recent techniques of image-text data curation typically involve two branches: *data filtering* [317, 318] that removes noisy data, and *data improvement* [319, 96] that refines multi-modal alignment. Despite training better VLMs is often the objective of data curation, these models themselves are also used for curating better data. For instance, CLIP is frequently used for data filtering with CLIP score (i.e., similarity between visual

and text features) [34, 305, 306], and some methods use BLIP-2 to reduce semantic noise [307, 96]. As both *data filtering* and *data improvement* address essential aspects of data quality, we consider both of them in our approach for more comprehensive data enhancement. In addition, multi-modality data curation in fine-grain such as segment-text pairs remains underexplored, requiring further research to support more precise, context-aware applications. Beyond OVS, our proposed data cleaning methodology has the potential for broader applicability across other contexts.

Pre-Training Models. Pre-training models often have high reusability and adaptability, which can be easily reused in various approaches without further fine-tuning [269, 312, 270, 320, 81]. For instance, *Vision-Language Models (VLMs)* like CLIP [35] and ALIGN [321] link text with images for zero-shot image classification. *Multimodal Large Language Models (MLLMs)*, such as LLaVA [103], BLIP-2 [322], and GPT-Vision [323], support diverse tasks such as captioning, reasoning, and visual chat. Vision models like DINO [324] and DINOv2 [325] provide universal visual feature representations. Despite their success, such models are not almighty solutions, having various limitations for specified tasks [326]. In this work, we study how to effectively harness the value of data to compensate for the shortcomings of such models in a training-free manner.

5.3 Our Approach

The goal of an OVS method is to segment an input image given an arbitrary set of textual labels. Our training-free framework *ReME* consists of a data pipeline with two key phases, ***initial pairing*** and ***data enhancing***, to construct a high-quality reference set of segment-text pairs (Fig. 5.2); followed by a simple ***similarity based retrieval*** process to unveil the performance gains enabled by our data (Fig. 5.4).

5.3.1 Initial Pairing

Our data pipeline starts by constructing a diverse base set with segment-text pairs, where we aim to ensure broad coverage of various objects and contextual elements.

As shown in Fig. 5.2, for each input image, a segmenter generates class-agnostic segment masks, and an image description generator produces semantically rich descriptions. Following [237, 316], we extract noun phrases (i.e., nouns with descriptive modifiers such as *"a lovely white rabbit"*) from the descriptions to produce candidate textual labels, which reserve richer semantics than pure nouns. Then, we pair the segments and labels using a capable VLM. Despite the noises, CLIP still provides a certain degree of correctness for segment labeling, as evidenced by CLIP-based OVS methods [266, 268, 297, 267, 272]. We thus leverage CLIP to perform initial pairing. Given an image, we extract CLIP embeddings of its segments and noun phrases, $\boldsymbol{S} \in \mathbb{R}^{m \times d}$ and $\boldsymbol{L} \in \mathbb{R}^{n \times d}$, respectively, where m, n is the number of segments and labels, and d is the embedding dimension. Note that all embeddings across the chapter are L2 normalized so that $X \cdot Y^T$ is equivalent to their cosine similarity $\langle X, Y \rangle$. We then compute $sim = \boldsymbol{S} \cdot \boldsymbol{L}^T \in \mathbb{R}^{m \times n}$ to pair each label with its closest segment, where sim_{ij} is the similarity between $\boldsymbol{S}_i$ and $\boldsymbol{L}_j$. The top-matching S_{i^*} of L_j is identified by: $i^* = \arg\max_i sim_{ij}$. The unpaired segments are dropped to reduce the redundancy in class-agnostic image segments.

Despite including diverse segments paired with noun phrase labels, the base set still has unaddressed quality concerns, which is our major focus in the following phase.

5.3.2 Data Enhancing

As shown in Fig. 5.2, through initial pairing, we obtain a base set with paired segments and labels. Various factors negatively affect its quality: (1) irrelevant textual labels caused by object halluci-

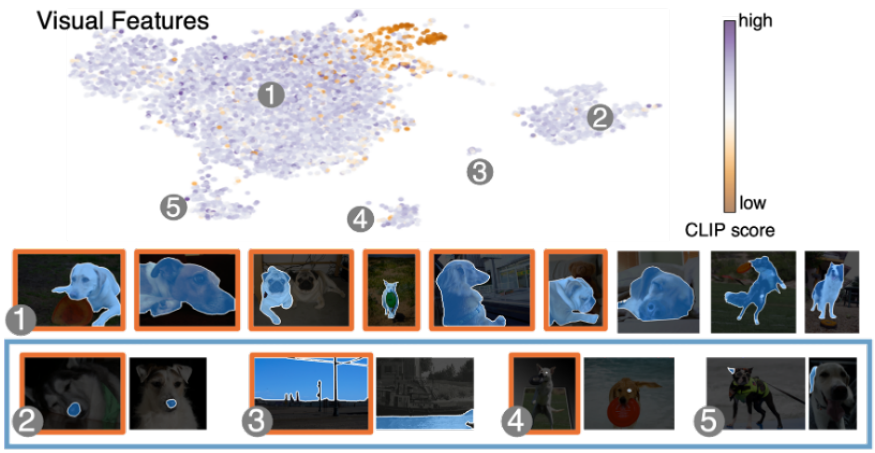


Figure 5.3: **The superiority of intra-modality over cross-modality for data issue detection.** The plot provides the UMAP projection of segment embeddings in the base set labeled as "dog", colored by cross-modal similarity scores (CLIP scores). **Blue boxes** highlight misalignments detected by our filtering; **orange boxes** are those detected by low CLIP scores, which remove correct pairings while leaving many misalignments unaddressed.

nations of the description generator, (2) meaningless or partial segments from over-segmentation, and (3) segment-text pairing errors. These challenges are essentially introduced by the ambiguity in cross-modality. We opt for navigating them with intra-modality operations—which we find to be more discriminative to effectively clean and enrich such data.

Unlike the common practice of dropping segment-label pairs with low cross-modal similarity scores (i.e., CLIP scores) [34, 305, 306], we reexamine data features and test whether CLIP scores can favorably clean noisy data. As shown in Fig. 5.3, our observation brings two key insights: (1) CLIP scores tend to mistakenly remove correct pairs while leaving many misalignments unaddressed; (2) Visual features, i.e., embeddings of image segments, have favorable discriminative power for data issue detection. (Please refer to Sec. 5.4.3 for extensive discussions.)

Group-Based Filtering. Guided by our observation, in this phase, we detect misalignments by leveraging the discriminativeness of intra-modal features. Specifically, we aim to leverage the visual features of segments to identify outliers in their own modality. First, for our labels in the

format of noun phrases, we view those with the same "root" noun as an identical label. Next, we group each single segment-label pair according to identical labels. For example, segments labeled by *"a small dog"* or *"an adorable dog"* belong to the same group denoted by the root *"dog."* Note that we reserve all descriptive modifiers in the phrases to keep semantic richness; "roots" are just used for grouping.

This is an important step for detecting data misalignments. Since each group of segments corresponds to an identical label, their visual features should be inherently consistent, and misaligned data would be automatically highlighted as outliers. As shown in Fig. 5.3, the majority of segments marked by ❶ accurately depict dogs, while others are misalignments located as outliers in the projection, indicating their distinctive features.

In detail, we define the segment center S_{center} of each group, computed by the median of all visual features. S_{center} denotes the representative features for correctly labeled segments. We then compute the intra-modal similarity between each segment in the current group and S_{center}, indicating the possibility of a segment being mistakenly labeled. δ_{filter} percent of segments with the lowest scores are considered mistakenly paired, where we remove the corresponding noun phrases from their labels. By performing this filtering guided by intra-modality feature similarity, we effectively reduce misalignment in the base set.

Semantic Enriching. By clearing outliers in each group, we have significantly mitigated segment-text misalignments. However, data enhancement remains incomplete in terms of semantic diversity – While statistical analysis of the entire label corpus of our base set reveals the textual modality includes different words for the same concept (e.g., *"bike"* and *"bicycle"*), such alternatives are missing in individual segments. This results in a lack of semantic richness, as we expect our method to recognize concepts in various representations. Consistently, we design an intra-

modality-based approach to diversify our labels. As shown in Fig. 5.2, we collect embeddings of identical labels, i.e., the "root" noun of our noun phrase labels, denoted as $\{L_1, \ldots, L_n\}$. Then, we compute their pair-wise cosine similarity $\langle L_i, l_j \rangle$, and identify k top-similar pairs to be treated as synonyms. Next, label semantics are enriched by adding these synonyms. For instance, *"cat"-"kitten"* is a pair of synonyms, so, if a segment has a label *"a small cat"*, we add *"a small kitten"* to become its label as well. Those synonym-enriched noun phrases are our finalized labels. This process further diversifies our textual descriptions by leveraging the global semantic richness present within the data. Note that, an LLM could produce even better semantic enriching, while we opt for a lightweight-yet-effective solution to avoid the additional overhead of large models.

5.3.3 Similarity-based Retrieval

Following Sec. 5.3.2, we have constructed a reference set with well-paired segments and labels and stored their embeddings. At inference time, given a test image and a set of target classes, our goal is to assign pixel-level labels to the test image by referring to the reference set. Inspired by Tip-Adapter [327], we perform a simple retrieval strategy grounded in feature similarity, where labels of test segments are estimated by their closest matches in the reference set.

As shown in Fig. 5.4, the reference set includes embeddings of m segments $\boldsymbol{S}_{\text{ref}} \in \mathbb{R}^{m \times d_1}$ and n labels $\boldsymbol{L}_{\text{ref}} \in \mathbb{R}^{n \times d_2}$. The label assignments are encoded by a binary matrix $\boldsymbol{O}_{\text{ref}} \in \mathbb{R}^{m \times n}$, with an entry 0 indicating the corresponding label is absent in the segment and 1 otherwise.

Given a test image, we segment it into k class-agnostic masks, denoted by $\boldsymbol{M}_{\text{seg}} \in \mathbb{R}^{k \times h \times w}$, where $h \times w$ is the image size. We extract the segment and test class embeddings $\boldsymbol{S}_{\text{test}} \in \mathbb{R}^{k \times d_1}$ and $\boldsymbol{L}_{\text{test}} \in \mathbb{R}^{c \times d_2}$, using the same encoders applied to the reference set, where d_1 and d_2 are the embedding dimensions of the image and text modalities, respectively. The affinity between test

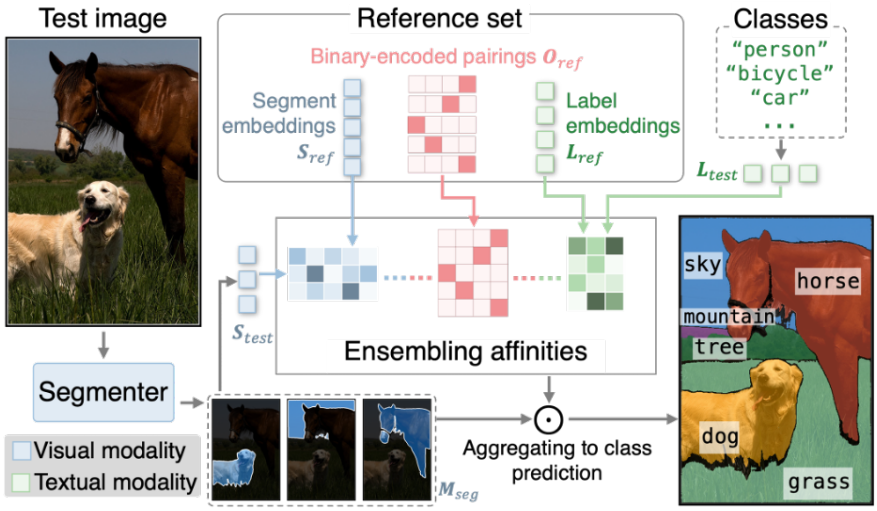


Figure 5.4: The process of similarity-based retrieval from the reference set given a test image and classes. Note that the same modality embeddings (i.e., reference & test segments; reference & test labels) are encoded by the same feature encoder, respectively.

segments and reference set labels is computed as:

$$A_1 = \text{Softmax}(\boldsymbol{S}_{\text{test}} \cdot \boldsymbol{S}_{\text{ref}}^T) \cdot \boldsymbol{O}_{\text{ref}}, \tag{5.1}$$

where $A_1 \in \mathbb{R}^{k \times n}$, and Softmax normalizes the affinities to formulate the probability distribution of labels for each test segment. Similarly, the affinity matrix between the reference labels and test classes is represented as:

$$A_2 = \text{Softmax}(\boldsymbol{L}_{\text{ref}} \cdot \boldsymbol{L}_{\text{test}}^T), \tag{5.2}$$

where $A_2 \in \mathbb{R}^{n \times c}$. Ensembling the affinities, for k test segments, the predicted logit is defined as: $\boldsymbol{P}_{\text{seg}} = A_1 \cdot A_2, \boldsymbol{P}_{\text{seg}} \in \mathbb{R}^{k \times c}$. Then, we calculate pixel-wise label probabilities $\boldsymbol{P}_{\text{test}} \in \mathbb{R}^{h \times w \times c}$ by aggregating ensembled affinities according to the segment masks $\boldsymbol{M}_{seg}$, where the probability of class j at coordinates (x, y) is:

$$\boldsymbol{P}_{\text{test}}^{(x,y,j)} = \sum_{i=1}^{k} \boldsymbol{P}_{\text{seg}}^{(i,j)} \cdot \boldsymbol{M}_{\text{seg}}^{(i,x,y)}. \tag{5.3}$$

Lastly, the predicted mask $\hat{l} \in \mathbb{R}^{h \times w}$ is determined by: $\hat{l}^{(x,y)} = \arg\max_j \boldsymbol{P}_{\text{test}}^{(x,y,j)}$, where $\hat{l}^{(x,y)} \in [0, c-1]$ indicates class predictions, which conclude our retrieval phase.

5.4 Experiments

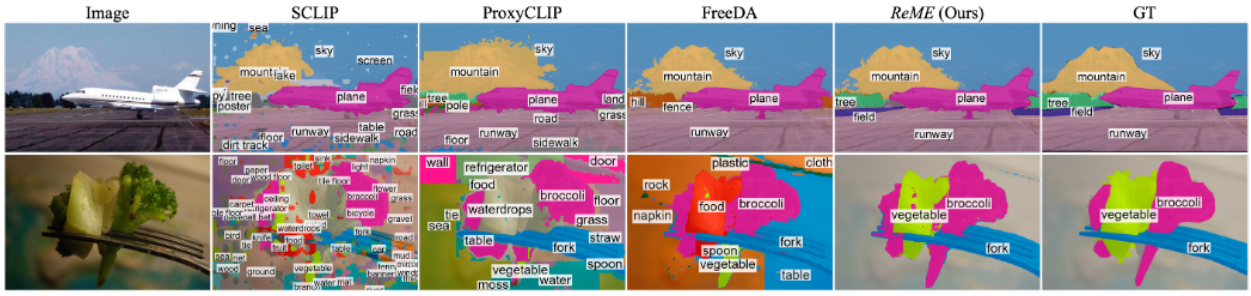


Figure 5.5: **Qualitative results of *ReME* in comparison with other training-free OVS methods.** Examples are from ADE20K [287] (w/ 150 classes) and COCO Stuff [288] (w/ 171 classes), respectively. SCLIP is based on CLIP attention; ProxyCLIP enhances CLIP attention with DINO features; FreeDA and *ReME* are retrieval-based methods, adopting the same superpixel-algorithm [301] for class-agnostic segmentation. We observe increasing quality of OVS results from left to right, with less noise in both masks and assigned labels.

5.4.1 Experimental Setup

Datasets for standard benchmarks. We conducted experiments on 10 widely-used OVS benchmarks to evaluate *ReME*, including validation splits of Pascal VOC (VOC) [329], Pascal Context (PC) [330], COCO Object (Object) [331], COCO Stuff (Stuff) [288], Cityscapes (City) [285], and ADE20K [287]. Specifically, VOC has 20 object classes (VOC-20). PC involves PC-459 with 459 classes and PC-59 with 59 frequent classes. Object and Stuff provide COCO-2017 image annotations of 81 and 171 classes, respectively. City captures street scenes with 19 classes. ADE20K includes A-150 and A-849 with 150 and 847 classes. For VOC-20 and PC-59, we consider pixels not belonging to any class as *"background"*, represented by VOC-21 and PC-60, respectively. We use the standard mean Intersection-over-Union (mIoU) to measure OVS performance.

Implementation. We obtain textual descriptions of input images with LLaVA-1.5 [103]. For class-agnostic segmenter, we deploy Felzenszwalb's algorithm by default, which is a lightweight superpixel-based algorithm, following the same settings as [289]. The initial pairing is performed by CLIP [35] with ViT-B. In the data-enhancing and retrieval phases, we use CLIP-encoded text

Table 5.1: **Comparison to state-of-the-art training-free OVS approaches.** The best results are **bolded**, with the second-best underlined. We also analyze data robustness by varying the image resources from the default COCO-2017 to VOC and ADE, respectively (w/o SAM version), where leading performances over SAM-free baselines are **bolded**.

Methods	Post-processing	mIoU									
		VOC-20	VOC-21	City	PC-59	PC-60	Object	Stuff	A-150	PC-459	A-847
Training-free Methods without SAM											
GEM [310]	✗	46.2	24.7	-	32.6	21.2	-	15.1	10.1	4.6	3.7
MaskCLIP [266]	✓	74.9	38.8	12.6	25.5	23.6	20.6	14.6	9.8	-	-
ReCo [269]	✓	62.4	27.2	23.2	24.7	21.9	17.3	16.3	12.4	-	-
SCLIP [286]	✓	83.5	61.7	34.1	36.1	31.5	32.1	23.9	17.8	9.3	6.1
CaR [297]	✓	91.4	67.6	15.1	39.5	30.5	36.6	11.2	17.7	11.5	5.0
NACLIP [328]	✓	83.0	64.1	38.3	38.4	35.0	36.2	25.7	19.1	9.0	6.5
CLIPtrase [268]	✓	81.2	53.0	21.1	34.9	30.8	39.6	24.1	17.0	9.9	5.9
PnP [312]	✓	79.1	51.3	19.3	31.0	28.0	36.2	17.9	14.2	5.5	4.2
FreeDA [289]	✓	87.9	55.4	36.7	43.5	38.3	37.4	28.8	22.4	10.2	5.3
ProxyCLIP [267]	✗	83.2	60.6	40.1	37.7	34.5	39.2	25.6	22.6	11.2	6.7
DiffSegmenter [298]	✓	71.4	60.1	-	27.5	25.1	37.9	-	-	-	-
OVDiff [270]	✓	80.9	68.4	23.4	32.9	31.2	36.2	20.3	14.1	12.0	6.6
***ReME* (Ours)**	✗	**92.3**	**79.6**	**50.4**	**44.9**	**41.6**	**45.5**	**33.1**	**26.1**	**14.1**	**8.4**
***ReME* (Ours - VOC)**	✗	84.7	**75.0**	**43.9**	40.9	**38.7**	**40.8**	22.6	**25.2**	12.8	**8.3**
***ReME* (Ours - ADE)**	✗	84.3	**72.3**	**42.1**	**44.0**	**39.7**	35.8	27.0	**26.0**	**13.2**	**8.6**
Training-free Methods with SAM											
RIM [273]	✗	77.8	-	-	34.3	-	44.5	-	-	-	-
CaR w/ SAM [297]	✗	-	70.2	16.9	40.5	31.1	37.6	12.4	17.9	11.8	5.7
CLIPtrase w/ SAM [268]	✗	82.3	57.1	-	36.4	32.0	44.2	24.8	17.2	10.6	6.0
ProxyCLIP w/ SAM [267]	✗	80.4	59.3	37.0	37.0	33.6	35.4	25.0	19.1	6.9	4.8
CorrCLIP [320]	✗	91.6	74.1	47.7	45.5	40.3	43.6	30.6	-	-	-
***ReME* w/ SAM (Ours)**	✗	**93.2**	**82.2**	**59.0**	**53.1**	**44.6**	**48.2**	**33.3**	**28.2**	**15.8**	**8.8**

features, and employ DINOv2 with ViT-L as the default visual feature encoder. Hyperparameters δ_{filter}, k_{sim} are both set to 30. To compare with SAM-involved OVS baselines, we optionally use SAM as the segmenter, where we prompt a 32×32 point grid to obtain masks. More implementation details are in supplementary.

Our reference set construction requires only images as input—no GT captions, classes, or masks, and we default to using COCO-2017 [331, 288] images, which depict everyday scenes with objects in their natural contexts.

5.4.2 Comparison to the State-of-the-arts

Baselines. We compare *ReME* with 14 training-free OVS approaches, including ReCo [269], MaskCLIP [266], SCLIP [286], NACLIP [328], GEM [310], PnP [312], FreeDA [289], RIM [273],

OVDiff [270], CLIPtrase [268], DiffSegmenter [298], CaR [297], ProxyCLIP [267], and CorrCLIP [320]. Since post-processing techniques for mask refinements, such as DenseCRF [332], enhance OVS performance but introduce computational overhead, we indicate whether each method applies such refinements. Moreover, as SAM is widely regarded as a strong backbone for segment-related tasks, we conduct our comparison across both SAM-free and SAM-involved approaches.

Comparison. Table 5.1 presents the quantitative results, where we observe that *ReME* consistently outperforms training-free OVS baselines across ten evaluated benchmarks. Notably, *ReME* achieves superior performance in challenging benchmarks with complex scenarios and large numbers of classes, including A-150, PC-459, and A-847. On Cityscapes, an increase of 8.4 mIoU points (11.3 mIoU increase for SAM-involved comparison) further indicates that our approach excels in domain-specific scenarios. We also provide qualitative results of *ReME* in Fig. 5.5.

Data robustness. To evaluate the data robustness of *ReME*, we vary image resources of our reference set, replacing the default COCO-2017 images with those from VOC and ADE. As shown in Table 5.1, SAM-free *ReME* maintains strong performance over baselines. Across 10 benchmarks, *ReME* achieves the highest mIoU in 6 benchmarks with VOC and 7 with ADE. These results highlight the effectiveness and flexibility of our approach in diverse settings.

5.4.3 Analysis, Ablation, and Discussion

Hyperparameters. Our data-enhancement process involves two key hyperparameters: *the drop ratio per group* (δ_{filter}) in *group-based filtering* (i), and *the number of top-similar label pairs* (k_{sim}) in *semantic enriching* (ii). To determine their values, we perform a grid search on $1k$ (1%) randomly sampled images from the training split of COCO Stuff (with no overlap with evaluation data), as shown in Fig. 5.6. Initially, increasing both parameters improves performance, as more misaligned data is filtered out and label diversity is enhanced. However, beyond a certain point,

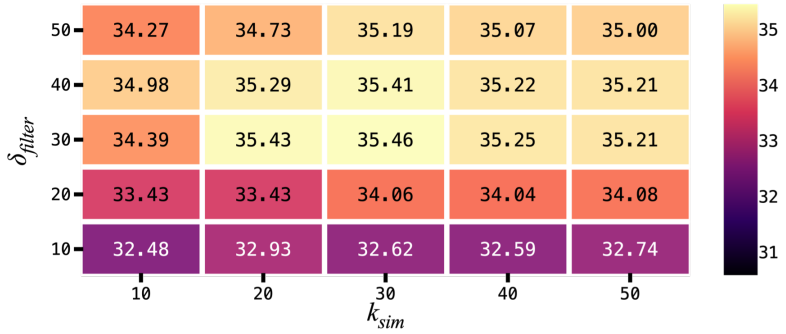


Figure 5.6: Hyperparameter analysis. δ_{filter} is the drop ratio in *group-based filtering*; k_{sim} is the number of top-similar pairs in *semantic enriching*.

Table 5.2: Impact of data enhancement components.

Components	mIoU			
	VOC-20	PC-59	Object	A-150
Base set (no enhancement)	70.03	35.42	39.38	22.03
w/ (i) Group-based filtering	91.10	40.66	42.48	24.09
w/ (ii) Semantic enriching	79.50	36.69	39.92	23.41
w/ (i) and (ii)	**92.34**	**44.89**	**45.50**	**26.13**

further increases cause a decline in performance: a high drop ratio removes correct pairings and reduces data diversity, while an excessive k_{sim} introduces noise by including less-similar labels. We use 30 as the default value for both parameters in our implementation.

Data enhancing component analysis. We analyze the effects of our data enhancement components: (i) group-based filtering and (ii) semantic enriching. As shown in Table 5.2, the base set without enhancement yields relatively low performance, highlighting the critical role of data quality. We also observe higher individual effectiveness of (i), while enriching labels alone provides modest gains due to data noises. Notably, the performance improves significantly when both are integrated, demonstrating the complementary benefits of our data enhancement components.

Analysis of different data-cleaning approaches. We apply group-based filtering to remove $\delta_{filter}\%$ of outliers from each label-identified group (refer to Sec. 5.3.2). This experiment examines the impact of different data-cleaning strategies by comparing three methods: (a) the common

Table 5.3: Analysis of different data filtering approaches.

Data cleaning	mIoU			
	VOC-20	PC-59	Object	A-150
Global CLIP score$^{(a)}$	79.34	39.79	40.79	21.18
Group-based CLIP score$^{(b)}$	80.05	41.84	41.81	22.79
Ours $^{(c)}$	**92.34**	**44.89**	**45.50**	**26.13**

Table 5.4: Analysis of visual encoder variations.

Feature encoder	mIoU			
	VOC-20	PC-59	Object	A-150
CLIP$_B$	91.61	36.51	39.82	24.72
CLIP$_L$	92.16	37.63	40.52	25.56
DINOv2$_B$	91.72	43.65	44.71	25.29
DINOv2$_L$	**92.34**	**44.89**	**45.50**	**26.13**

approach of dropping segment-label pairs with the globally lowest cross-modal similarity scores (CLIP scores); (b) a variant that also uses CLIP scores, removing segment-label pairs within each constructed groups; (c) our group-based filtering based on intra-modal feature similarity. By only changing this module, we evaluate OVS performance as shown in Table 5.3, where we keep the drop ratio fixed at 30%. The results indicate that (a) and (b) perform much worse than (c), suggesting that filtering based on CLIP scores is less effective, likely due to its weaker alignment with actual pairing quality.

Analysis of different visual encoders. We investigate the impact of different visual feature encoders, including CLIP and DINOv2 with ViT-B and ViT-L architectures, denoted by CLIP$_B$, CLIP$_L$, DINOv2$_B$, and DINOv2$_L$, respectively. As shown in Table 5.4, upgrading from the base to large variants within the same encoder family results in slight performance improvements. When comparing CLIP and DINOv2 at the same scale, DINOv2 achieves better performance as expected, given its specialization in visual representation learning. Notably, by using CLIP alone and removing DINOv2 from our framework, we still achieve performance that is comparable to or better than

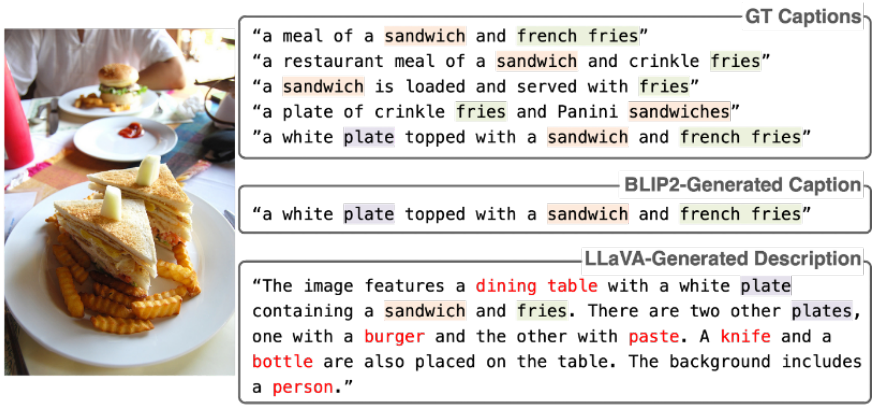


Figure 5.7: Image descriptions from different resources. Red text highlights concepts uniquely present in the LLaVA description.

Table 5.5: Ablation study of the image description generator.

Description generator	mIoU			
	VOC-20	PC-59	Object	A-150
LLaVA [103]	**92.34**	**44.89**	**45.50**	**26.13**
BLIP-2 [322]	89.41	40.85	37.64	24.31
GT Caption	89.02	40.15	37.77	24.09

Table 5.6: Ablation study of the segmenter.

Segmenter	mIoU			
	VOC-20	PC-59	Object	A-150
Superpixel [301]	92.34	44.89	45.50	26.13
SAM [10]	93.15	**53.10**	48.21	**28.21**
SAM2 [93]	**93.18**	52.04	**48.40**	**28.21**

baselines using similar backbones. This suggests that our approach is robust to variations in visual encoders.

Analysis of the description generator. Providing semantically rich image descriptions is important for our data construction. However, not all descriptions are equally informative. Fig. 5.7 presents an image from the COCO-2017 [288] dataset alongside its descriptions from three resources: (1) GT captions, (2) BLIP-2 [322], and (3) LLaVA [103], where we highlight the detected visual concepts. We can observe common concepts presented in all three, "sandwich", "fries", and "plate". However, concepts such as "dining table", "burger", "knife", "bottle", and "person" are

missing from both (1) and (2). To further study the impact of using BLIP-2 and LLaVA as description generators, we conduct a quantitative evaluation. As shown in Table 5.5, LLaVA demonstrates superior performance, achieving significantly higher mIoU compared to BLIP-2, where BLIP-2 captions yield only marginal improvements over GT captions. These results highlight LLaVA's ability to generate richer, more detailed descriptions and quantitatively confirm its effectiveness in enhancing training-free OVS with retrieval.

Analysis of the segmenter. We also investigate the impact of different segmenters in Table 5.6 by replacing our default superpixel algorithm-based segmenter with advanced segmentation models, SAM [10] and SAM2 [93]. While a more advanced segmentation model leads to slightly better performance, our data-quality-enhancement pipeline enables even a simple segmentation algorithm to achieve strong results compared to baseline methods.

Analysis of data effectiveness. We evaluate data effectiveness by comparing *ReME* with FreeDA [289], a representative retrieval-based method that also utilizes COCO-2017 dataset [331, 288]. Unlike our approach, which takes images as input and generates textual descriptions, FreeDA uses GT captions to generate synthetic images for constructing its reference set. As shown in Table 5.7, despite FreeDA incorporating five times more input samples, its reference set contains fewer unique labels than *ReME*, due to the limited semantic diversity of GT captions (refer to Fig. 5.7). Our data-enhancement strategy improves data quality while reducing dataset size, resulting in a reference set with over a million fewer segment-text pairs than FreeDA. Using this considerably smaller reference set, *ReME* achieves superior performance even with a much simpler retrieval mechanism, highlighting the effectiveness of high-quality, semantically rich data over quantity.

Analysis of inference time. Inference time for training-free OVS varies due to factors such as the number of classes, image resolution, and inference strategies. To evaluate this, we compare

Table 5.7: Comparing data effectiveness with FreeDA, a retrieval-based OVS baseline that uses a synthetic reference set.

Methods	Input data		Reference set		
	Format	Size	#labels	#pairs	Storage
Ours	Images	118k	41k	1,023k	4GB
FreeDA [289]	Captions	591k	19k	2,167k	17GB

Table 5.8: Inference time comparison (seconds/image).

Methods	PC-59	A-150	PC-459	A-847
ProxyCLIP [267]	0.17	0.27	0.21	0.32
SCLIP [286]	0.20	0.34	0.57	0.53
CaR [297]	5.06	14.67	33.46	75.09
FreeDA [289]	0.89	0.89	4.89	5.01
Ours	0.29	0.34	0.31	0.34

our method with four representative approaches: (1) ProxyCLIP [267], which augments CLIP attention with DINO features, (2) SCLIP [286], which exploits CLIP self-attention followed by PAMR post-processing [333], (3) CaR [297], which iteratively queries two CLIP models for mask proposal and classification, and (4) FreeDA [289], a retrieval-based method. All experiments are conducted on two NVIDIA 4090 GPUs, measuring total inference time per dataset to compute the average. As shown in Table 5.8, (3) is the most time-consuming due to its iterative processing. Our method surpasses the retrieval-based baseline (4), benefiting from a smaller reference set and streamlined retrieval design, while remaining competitive with (1) and (2), which do not rely on reference sets.

5.5 Conclusion

In this chapter, we presented *ReME*, a data-centric framework for training-free open-vocabulary semantic segmentation by refining multimodal embeddings. The study highlighted the often-overlooked role of data quality in dense scene understanding tasks and demonstrated how carefully constructed reference data can substantially improve segmentation performance. Through

the proposed data pipeline, a reference set with well-aligned, semantically rich, and contextually relevant segment-text pairs can be constructed from real images. Experimental results show that improving the quality of reference data can be more effective than relying solely on complex retrieval mechanisms or model-specific architectural modifications. These findings suggest that data-centric design offers a promising pathway for extending the capabilities of vision-language models in training-free open-vocabulary segmentation.

Chapter 6

Conclusion

In summary, this dissertation examines how collaborative intelligence among humans, models, and data can enable the development of more interpretable, reliable, and scalable AI systems. As machine learning models continue to grow in complexity and data pipelines expand to unprecedented scales, the challenges faced by modern AI systems are no longer limited to improving prediction accuracy. Instead, understanding model behavior, diagnosing hidden failure modes, validating large-scale multimodal data, and extending AI capabilities efficiently have become more critical for the real-world deployment of AI.

The works presented in this dissertation addressed these challenges through a series of complementary approaches, which integrate model analysis, human insight, and data-centric design across different stages of the machine learning lifecycle, improving both the transparency and robustness of AI systems. Rather than focusing solely on designing new model architectures or increasing training data, the dissertation research emphasizes that many challenges in modern AI arise from the lack of effective mechanisms for understanding, validating, and coordinating existing models and data resources. By introducing visual analytics frameworks, human-aided validation techniques, and collaborative model mechanisms, this dissertation demonstrates how systematic anal-

ysis of model behaviors and data representations can reveal hidden patterns, correct systematic issues, and enable new capabilities in existing AI systems.

The first part of the dissertation focuses on improving the understanding of model decision-making. Modern models often operate as highly complex structures whose internal reasoning processes are difficult to interpret directly. Chapter 2 addresses this challenge by developing visual analytics techniques that expose patterns in model predictions and learned representations. Through interactive visualization of attribution signals and representation structures, the proposed approaches allow analysts to examine how neural networks associate input features with predicted outcomes and how different models encode semantic information in their feature spaces. By transforming high-dimensional model behaviors into interpretable visual structures, these methods provide a practical way to analyze the internal mechanisms of neural networks and to identify systematic patterns that would otherwise remain hidden behind aggregate performance metrics.

Building on this understanding, the dissertation then investigated the validation of model behavior. Chapter 3 shows that many model failures originate from systematic patterns embedded within specific subsets of data. To address this problem, the work introduces human-aided analytical frameworks that enable the discovery of such patterns through representation-level analysis and visual exploration. These techniques allow practitioners to identify coherent data slices that exhibit problematic model behaviors, such as spurious correlations or unintended labeling issues. Furthermore, the work demonstrates that targeted human feedback can be propagated through learned representation spaces to correct such issues with minimal annotation effort. This approach highlights how combining machine learning representations with structured human supervision can significantly improve model robustness without requiring extensive additional labeling.

The dissertation further extends the validation perspective from models to the data. As founda-

tion models increasingly enable the automatic generation of large-scale multimodal datasets, ensuring the quality of such data has become a critical challenge. Chapter 4 presents a visual analytics framework for inspecting multimodal data pipelines that rely on foundation models for large-scale annotation. The system integrates automated data profiling with interactive visualization to help analysts detect labeling inconsistencies, distributional anomalies, and other hidden issues. By combining algorithmic analysis with human-in-the-loop validation, the framework demonstrates how large-scale data pipelines can be systematically inspected and refined. These results underscore the importance of treating data validation as a first-class component of modern AI development rather than an afterthought following model training.

Finally, the dissertation investigates how AI capabilities can be efficiently extended to handle more challenging tasks in open-world environments. Traditional approaches to scaling AI capabilities often rely on scaling model size or collecting increasingly large annotated datasets, both of which can incur significant computational and annotation costs. Chapter 5 explores an alternative perspective in which AI capabilities are extended through collaboration among existing models and data sources. By constructing high-quality multimodal reference representations and enabling retrieval-based reasoning, the proposed framework demonstrates that the fine-grained perception capabilities of vision-language models can be significantly enhanced without additional model training. These findings suggest that many limitations of current AI systems may stem not from insufficient model capacity but from the lack of mechanisms for effectively coordinating the knowledge embedded in existing models and datasets.

Taken together, the research results presented throughout this dissertation suggest a broader perspective on the development of modern AI systems. Understanding model behavior, validating models and datasets, and extending AI capabilities are deeply interconnected challenges, yet

they are often studied in isolation. Interpretability techniques help reveal how models behave, validation frameworks identify systematic issues in both models and data, and collaborative model mechanisms enable the knowledge embedded in different components to be combined effectively. Through this perspective, collaborative intelligence emerges as a unifying principle for designing AI systems that are both powerful and trustworthy.

Future Directions. Looking ahead, several promising directions emerge from the findings of this dissertation. One important opportunity lies in developing systematic frameworks for data curation in the era of foundation models. As multimodal models and generative systems increasingly automate the creation of synthetic datasets, ensuring the quality of such data will become a central challenge for machine learning research. Future work may explore principled metrics for multimodal data quality, automated pipelines for detecting inconsistencies in large datasets, and hybrid systems that combine algorithmic data profiling with interactive human analysis.

Another important avenue involves the development of collaborative model ecosystems. Rather than relying on monolithic models that attempt to solve all tasks independently, future AI systems may consist of multiple specialized models that interact dynamically to solve complex problems. Designing mechanisms for coordinating these models, aligning their representations, and composing their capabilities efficiently could enable AI systems to integrate perception, reasoning, and generation in more flexible ways. Such collaborative systems may provide a scalable alternative to simply increasing model size or training data.

Finally, as AI technologies continue to be integrated into real-world decision-making processes, ensuring transparency and accountability will remain essential. Future research may therefore explore system-level monitoring frameworks that combine explainability, validation, and human oversight to ensure that increasingly complex AI systems remain reliable and trustworthy. Devel-

oping such mechanisms will be critical for enabling responsible deployment of AI technologies across scientific, industrial, and societal applications.

To conclude, by integrating visual analytics, human-aided validation, data-centric learning, and collaborative model mechanisms, the work presented in this dissertation provides a framework for improving the understanding, reliability, and scalability of AI systems. As AI continues to evolve and expand into increasingly complex environments, approaches that combine computational intelligence with human insight will be essential for building intelligent systems that are not only powerful, but also interpretable, reliable, and adaptable in real-world settings.

REFERENCES

[1] M. Awais, M. Naseer, S. Khan, R. M. Anwer, H. Cholakkal, M. Shah, M.-H. Yang, and F. S. Khan, "Foundation models defining a new era in vision: a survey and outlook," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 47, no. 4, pp. 2245–2264, 2025.

[2] C. Zhou, Q. Li, C. Li, J. Yu, Y. Liu, G. Wang, K. Zhang, C. Ji, Q. Yan, L. He *et al.*, "A comprehensive survey on pretrained foundation models: A history from bert to chatgpt," *International Journal of Machine Learning and Cybernetics*, vol. 16, no. 12, pp. 9851–9915, 2025.

[3] W. Peebles and S. Xie, "Scalable diffusion models with transformers," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2023, pp. 4195–4205.

[4] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[5] 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*, 2016, pp. 770–778.

[6] Y. Li, K. Zhang, J. Liang, J. Cao, C. Liu, R. Gong, Y. Zhang, H. Tang, Y. Liu, D. Demandolx *et al.*, "Lsdir: A large scale dataset for image restoration," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 1775–1787.

[7] X. Yu, M. Xu, Y. Zhang, H. Liu, C. Ye, Y. Wu, Z. Yan, C. Zhu, Z. Xiong, T. Liang *et al.*, "Mvimgnet: A large-scale dataset of multi-view images," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 9150–9161.

[8] Z. He, T. Liang, J. Xu, Q. Liu, X. Chen, Y. Wang, L. Song, D. Yu, Z. Liang, W. Wang *et al.*, "Deepmath-103k: A large-scale, challenging, decontaminated, and verifiable mathematical dataset for advancing reasoning," *arXiv preprint arXiv:2504.11456*, 2025.

[9] OpenAI, "Gpt-4 technical report," *ArXiv*, vol. abs/2303.08774, 2023.

[10] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo *et al.*, "Segment anything," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 4015–4026.

[11] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby *et al.*, "DINOv2: Learning robust visual features without supervision," *arXiv preprint arXiv:2304.07193*, 2023.

[12] Y. Huang, Y. Chen, and Z. Li, "Applications of large scale foundation models for autonomous driving," *arXiv preprint arXiv:2311.12144*, 2023.

[13] M. Moor, O. Banerjee, Z. S. H. Abad, H. M. Krumholz, J. Leskovec, E. J. Topol, and P. Rajpurkar, "Foundation models for generalist medical artificial intelligence," *Nature*, vol. 616, no. 7956, pp. 259–265, 2023.

[14] H. Wang, T. Fu, Y. Du, W. Gao, K. Huang, Z. Liu, P. Chandak, S. Liu, P. Van Katwyk, A. Deac *et al.*, "Scientific discovery in the age of artificial intelligence," *Nature*, vol. 620, no. 7972, pp. 47–60, 2023.

[15] L. Ren, J. Dong, S. Liu, L. Zhang, and L. Wang, "Embodied intelligence toward future smart manufacturing in the era of ai foundation model," *IEEE/ASME transactions on mechatronics*, vol. 30, no. 4, pp. 2632–2642, 2024.

[16] X. Li, H. Xiong, X. Li, X. Wu, X. Zhang, J. Liu, J. Bian, and D. Dou, "Interpretable deep learning: Interpretation, interpretability, trustworthiness, and beyond," *Knowledge and Information Systems*, vol. 64, no. 12, pp. 3197–3234, 2022.

[17] F. Doshi-Velez and B. Kim, "Towards a rigorous science of interpretable machine learning," *arXiv preprint arXiv:1702.08608*, 2017.

[18] Z. C. Lipton, "The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery." *Queue*, vol. 16, no. 3, pp. 31–57, 2018.

[19] D. Gunning, "Explainable artificial intelligence (xai)," *Defense advanced research projects agency (DARPA), nd Web*, vol. 2, no. 2, p. 1, 2017.

[20] D. Kaur, S. Uslu, K. J. Rittichier, and A. Durresi, "Trustworthy artificial intelligence: a review," *ACM computing surveys (CSUR)*, vol. 55, no. 2, pp. 1–38, 2022.

[21] T. Zack, E. Lehman, M. Suzgun, J. A. Rodriguez, L. A. Celi, J. Gichoya, D. Jurafsky, P. Szolovits, D. W. Bates, R.-E. E. Abdulnour *et al.*, "Assessing the potential of gpt-4 to perpetuate racial and gender biases in health care: a model evaluation study," *The Lancet Digital Health*, vol. 6, no. 1, pp. e12–e22, 2024.

[22] S. Sagawa, P. W. Koh, T. B. Hashimoto, and P. Liang, "Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization," *arXiv preprint arXiv:1911.08731*, 2019.

[23] M. Zhang, N. S. Sohoni, H. R. Zhang, C. Finn, and C. Ré, "Correct-n-contrast: A contrastive approach for improving robustness to spurious correlations," *arXiv preprint arXiv:2203.01517*, 2022.

[24] E. Z. Liu, B. Haghgoo, A. S. Chen, A. Raghunathan, P. W. Koh, S. Sagawa, P. Liang, and C. Finn, "Just train twice: Improving group robustness without training group information," in *International Conference on Machine Learning*. PMLR, 2021, pp. 6781–6792.

[25] P. Kirichenko, P. Izmailov, and A. G. Wilson, "Last layer re-training is sufficient for robustness to spurious correlations," *arXiv preprint arXiv:2204.02937*, 2022.

[26] X. Xuan, J. P. Ono, L. Gou, K.-L. Ma, and L. Ren, "AttributionScanner: A visual analytics system for model validation with metadata-free slice finding," *arXiv preprint arXiv:2401.06462*, 2024.

[27] H. A. Simon, "Spurious correlation: A causal interpretation," *Journal of the American statistical Association*, vol. 49, no. 267, pp. 467–479, 1954.

[28] X. Xuan, Z. Deng, H.-T. Lin, and K.-L. Ma, "SLIM: Spuriousness mitigation with minimal human annotations," in *European Conference on Computer Vision*. Springer, 2024, pp. 215–231.

[29] J. Nam, J. Kim, J. Lee, and J. Shin, "Spread spurious attribute: Improving worst-group accuracy with spurious attribute estimation," *arXiv preprint arXiv:2204.02070*, 2022.

[30] S. Wu, M. Yuksekgonul, L. Zhang, and J. Zou, "Discover and cure: Concept-aware mitigation of spurious correlation," *arXiv preprint arXiv:2305.00650*, 2023.

[31] X. Xuan, X. Wang, W. He, J. P. Ono, L. Gou, K.-L. Ma, and L. Ren, "VISTA: A visual analytics framework to enhance foundation model-generated data labels," *IEEE Transactions on Visualization and Computer Graphics*, 2025.

[32] X. Dai, J. Hou, C.-Y. Ma, S. Tsai, J. Wang, R. Wang, P. Zhang, S. Vandenhende, X. Wang, A. Dubey *et al.*, "Emu: Enhancing image generation models using photogenic needles in a haystack," *arXiv preprint arXiv:2309.15807*, 2023.

[33] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale *et al.*, "Llama 2: Open foundation and fine-tuned chat models," *arXiv preprint arXiv:2307.09288*, 2023.

[34] C. Schuhmann, R. Vencu, R. Beaumont, R. Kaczmarczyk, C. Mullis, A. Katta, T. Coombes, J. Jitsev, and A. Komatsuzaki, "Laion-400m: Open dataset of clip-filtered 400 million image-text pairs," *arXiv preprint arXiv:2111.02114*, 2021.

[35] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*. PMLR, 2021, pp. 8748–8763.

[36] H. Xu, S. Xie, X. E. Tan, P.-Y. Huang, R. Howes, V. Sharma, S.-W. Li, G. Ghosh, L. Zettlemoyer, and C. Feichtenhofer, "Demystifying clip data," *arXiv preprint arXiv:2309.16671*, 2023.

[37] Z. Wan, X. Wang, C. Liu, S. Alam, Y. Zheng, J. Liu, Z. Qu, S. Yan, Y. Zhu, Q. Zhang *et al.*, "Efficient large language models: A survey," *arXiv preprint arXiv:2312.03863*, 2023.

[38] M. Xu, W. Yin, D. Cai, R. Yi, D. Xu, Q. Wang, B. Wu, Y. Zhao, C. Yang, S. Wang *et al.*, "A survey of resource-efficient llm and multimodal foundation models," *arXiv preprint arXiv:2401.08092*, 2024.

[39] Y. Yang, X. Wu, T. He, H. Zhao, and X. Liu, "SAM3D: Segment anything in 3d scenes," *arXiv preprint arXiv:2306.03908*, 2023.

[40] S. Cho, H. Shin, S. Hong, A. Arnab, P. H. Seo, and S. Kim, "Cat-seg: Cost aggregation for open-vocabulary semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 4113–4123.

[41] J. Yang, S. Yang, A. W. Gupta, R. Han, L. Fei-Fei, and S. Xie, "Thinking in space: How multimodal large language models see, remember, and recall spaces," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 10 632–10 643.

[42] B. Yin, Q. Wang, P. Zhang, J. Zhang, K. Wang, Z. Wang, J. Zhang, K. Chandrasegaran, H. Liu, R. Krishna *et al.*, "Spatial mental modeling from limited views," in *Structural Priors for Vision Workshop at ICCV'25*, 2025.

[43] J. Xu, S. De Mello, S. Liu, W. Byeon, T. Breuel, J. Kautz, and X. Wang, "Groupvit: Semantic segmentation emerges from text supervision," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 18 134–18 144.

[44] Y. Xing, J. Kang, A. Xiao, J. Nie, L. Shao, and S. Lu, "Rewrite caption semantics: Bridging semantic gaps for language-supervised semantic segmentation," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[45] H. Wang, P. K. A. Vasu, F. Faghri, R. Vemulapalli, M. Farajtabar, S. Mehta, M. Rastegari, O. Tuzel, and H. Pouransari, "Sam-clip: Merging vision foundation models towards semantic and spatial understanding," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 3635–3647.

[46] D. Malik, X. Xuan, and K.-L. Ma, "Towards interactive 3d surgical scene reconstruction: An incremental training and monitoring framework," in *2025 IEEE 22nd International Symposium on Biomedical Imaging (ISBI)*. IEEE, 2025, pp. 1–4.

[47] L. Cai, G. Xu, Q. Zhang, Z. Song, and W. Zhang, "Deep learning based channel estimation for deep-space communications," *IEEE Transactions on Vehicular Technology*, 2025.

[48] X. Xuan, Z. Deng, H.-T. Lin, Z. Kong, and K.-L. Ma, "SUNY: A visual interpretation framework for convolutional neural networks from a necessary and sufficient perspective," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 8371–8376.

[49] Z. Deng, X. Xuan, K.-L. Ma, and Z. Kong, "A reliable framework for human-in-the-loop anomaly detection in time series," *arXiv preprint arXiv:2405.03234*, 2024.

[50] X. Xuan, X. Zhang, O.-H. Kwon, and K.-L. Ma, "VAC-CNN: A visual analytics system for comparative studies of deep convolutional neural networks," *IEEE Transactions on Visualization and Computer Graphics*, vol. 28, no. 6, pp. 2326–2337, 2022.

[51] A. Adadi and M. Berrada, "Peeking inside the black-box: a survey on explainable artificial intelligence (xai)," *IEEE access*, vol. 6, pp. 52 138–52 160, 2018.

[52] J. Heer, M. Bostock, and V. Ogievetsky, "A tour through the visualization zoo," *Communications of the ACM*, vol. 53, no. 6, pp. 59–67, 2010.

[53] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek, "On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation," *PloS one*, vol. 10, no. 7, p. e0130140, 2015.

[54] 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*, 2016, pp. 2921–2929.

[55] 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*, 2017, pp. 618–626.

[56] 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)*. IEEE, 2018, pp. 839–847.

[57] D. Omeiza, S. Speakman, C. Cintas, and K. Weldermariam, "Smooth Grad-CAM++: An enhanced inference level visualization technique for deep convolutional neural network models," *arXiv preprint arXiv:1908.01224*, 2019.

[58] C. Olah, A. Mordvintsev, and L. Schubert, "Feature visualization," *Distill*, vol. 2, no. 11, p. e7, 2017.

[59] A. Mordvintsev, C. Olah, and M. Tyka, "DeepDream-a code example for visualizing neural networks," *Google Research*, vol. 2, no. 5, 2015.

[60] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, "Striving for simplicity: The all convolutional net," *arXiv preprint arXiv:1412.6806*, 2014.

[61] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," *arXiv preprint arXiv:1312.6034*, 2013.

[62] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *European conference on computer vision*. Springer, 2014, pp. 818–833.

[63] 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*, 2016, pp. 1135–1144.

[64] V. Petsiuk, A. Das, and K. Saenko, "Rise: Randomized input sampling for explanation of black-box models," *arXiv preprint arXiv:1806.07421*, 2018.

[65] D. Smilkov, N. Thorat, B. Kim, F. Viégas, and M. Wattenberg, "Smoothgrad: removing noise by adding noise," *arXiv preprint arXiv:1706.03825*, 2017.

[66] H. Wang, M. Du, F. Yang, and Z. Zhang, "Score-CAM: Improved visual explanations via score-weighted class activation mapping," *arXiv preprint arXiv:1910.01279*, 2019.

[67] Q. Zhang, L. Rao, and Y. Yang, "Group-cam: group score-weighted visual explanations for deep convolutional networks," *arXiv preprint arXiv:2103.13859*, 2021.

[68] P. E. Rauber, S. G. Fadel, A. X. Falcao, and A. C. Telea, "Visualizing the hidden activity of artificial neural networks," *IEEE transactions on visualization and computer graphics*, vol. 23, no. 1, pp. 101–110, 2016.

[69] M. Liu, J. Shi, Z. Li, C. Li, J. Zhu, and S. Liu, "Towards better analysis of deep convolutional neural networks," *IEEE transactions on visualization and computer graphics*, vol. 23, no. 1, pp. 91–100, 2016.

[70] K. Wongsuphasawat, D. Smilkov, J. Wexler, J. Wilson, D. Mane, D. Fritz, D. Krishnan, F. B. Viégas, and M. Wattenberg, "Visualizing dataflow graphs of deep learning models in tensorflow," *IEEE transactions on visualization and computer graphics*, vol. 24, no. 1, pp. 1–12, 2017.

[71] J. Krause, A. Perer, and K. Ng, "Interacting with predictions: Visual inspection of black-box machine learning models," in *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, 2016, pp. 5686–5697.

[72] H. Zeng, H. Haleem, X. Plantaz, N. Cao, and H. Qu, "CNNComparator: Comparative analytics of convolutional neural networks," *arXiv preprint arXiv:1710.05285*, 2017.

[73] J. Zhang, Y. Wang, P. Molino, L. Li, and D. S. Ebert, "Manifold: A model-agnostic framework for interpretation and diagnosis of machine learning models," *IEEE transactions on visualization and computer graphics*, vol. 25, no. 1, pp. 364–373, 2018.

[74] S. Das, D. Cashman, R. Chang, and A. Endert, "Beames: Interactive multimodel steering, selection, and inspection for regression tasks," *IEEE computer graphics and applications*, vol. 39, no. 5, pp. 20–32, 2019.

[75] F. Hohman, H. Park, C. Robinson, and D. H. P. Chau, "Summit: Scaling deep learning interpretability by visualizing activation and attribution summarizations," *IEEE transactions on visualization and computer graphics*, vol. 26, no. 1, pp. 1096–1106, 2019.

[76] L. Schubert, M. Petrov, S. Carter, N. Cammarata, G. Goh, and C. Olah, "Openai microscope," Apr 2020, https://microscope.openai.com/models.

[77] Y. Li, T. Fujiwara, Y. K. Choi, K. K. Kim, and K.-L. Ma, "A visual analytics system for multi-model comparison on clinical data predictions," *Visual Informatics*, vol. 4, no. 2, pp. 122–131, 2020.

[78] J. Wexler, M. Pushkarna, T. Bolukbasi, M. Wattenberg, F. Viégas, and J. Wilson, "The what-if tool: Interactive probing of machine learning models," *IEEE transactions on visualization and computer graphics*, vol. 26, no. 1, pp. 56–65, 2019.

[79] X. Zhang, J. P. Ono, H. Song, L. Gou, K.-L. Ma, and L. Ren, "Sliceteller: A data slice-driven approach for machine learning model validation," *IEEE Transactions on Visualization and Computer Graphics*, vol. 29, no. 1, pp. 842–852, 2022.

[80] A. Rawal, J. McCoy, D. B. Rawat, B. M. Sadler, and R. S. Amant, "Recent advances in trustworthy explainable artificial intelligence: Status, challenges, and perspectives," *IEEE Transactions on Artificial Intelligence*, vol. 3, no. 6, pp. 852–866, 2021.

[81] X. Yan, X. Xuan, J. P. Ono, J. Guo, V. Mohanty, S. A. Kumar, L. Gou, B. Wang, and L. Ren, "Vislix: An xai framework for validating vision models with slice discovery and analysis," in *Computer Graphics Forum*. Wiley Online Library, 2025, p. e70125.

[82] E. Pastor, L. De Alfaro, and E. Baralis, "Looking for trouble: Analyzing classifier behavior via pattern divergence," in *Proceedings of the 2021 International Conference on Management of Data*, 2021, pp. 1400–1412.

[83] Y. Chung, T. Kraska, N. Polyzotis, K. H. Tae, and S. E. Whang, "Slice finder: Automated data slicing for model validation," in *2019 IEEE 35th International Conference on Data Engineering (ICDE)*. IEEE, 2019, pp. 1550–1553.

[84] S. Sagadeeva and M. Boehm, "Sliceline: Fast, linear-algebra-based slice finding for ml model debugging," in *Proceedings of the 2021 International Conference on Management of Data*, 2021, pp. 2290–2299.

[85] E. Pastor, A. Gavgavian, E. Baralis, and L. de Alfaro, "How divergent is your data?" *Proceedings of the VLDB Endowment*, vol. 14, no. 12, pp. 2835–2838, 2021.

[86] S. Eyuboglu, M. Varma, K. K. Saab, J.-B. Delbrouck, C. Lee-Messer, J. Dunnmon, J. Zou, and C. Re, "Domino: Discovering systematic errors with cross-modal embeddings," in *International Conference on Learning Representations*, 2022.

[87] D. Zha, Z. P. Bhat, K.-H. Lai, F. Yang, Z. Jiang, S. Zhong, and X. Hu, "Data-centric artificial intelligence: A survey," *ACM Computing Surveys*, vol. 57, no. 5, pp. 1–42, 2025.

[88] M. Kejriwal, E. Kildebeck, R. Steininger, and A. Shrivastava, "Challenges, evaluation and opportunities for open-world learning," *Nature Machine Intelligence*, vol. 6, no. 6, pp. 580–588, 2024.

[89] J. Parmar, S. Chouhan, V. Raychoudhury, and S. Rathore, "Open-world machine learning: applications, challenges, and opportunities," *ACM Computing Surveys*, vol. 55, no. 10, pp. 1–37, 2023.

[90] Z. Wang, S. Cai, A. Liu, Y. Jin, J. Hou, B. Zhang, H. Lin, Z. He, Z. Zheng, Y. Yang *et al.*, "Jarvis-1: Open-world multi-task agents with memory-augmented multimodal language models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 47, no. 3, pp. 1894–1907, 2024.

[91] X. Wang, W. He, X. Xuan, C. Sebastian, J. P. Ono, X. Li, S. Behpour, T. Doan, L. Gou, H. W. Shen, and L. Ren, "USE: Universal segment embeddings for open-vocabulary image segmentation," in *Conference on Computer Vision and Pattern Recognition 2024*, 2024.

[92] J. Chen, Z. Xu, X. Pan, Y. Hu, C. Qin, T. Goldstein, L. Huang, T. Zhou, S. Xie, S. Savarese *et al.*, "Blip3-o: A family of fully open unified multimodal models-architecture, training and dataset," *arXiv preprint arXiv:2505.09568*, 2025.

[93] N. Ravi, V. Gabeur, Y.-T. Hu, R. Hu, C. Ryali, T. Ma, H. Khedr, R. Rädle, C. Rolland, L. Gustafson, E. Mintun, J. Pan, K. V. Alwala, N. Carion, C.-Y. Wu, R. Girshick, P. Dollár, and C. Feichtenhofer, "Sam 2: Segment anything in images and videos," *arXiv preprint arXiv:2408.00714*, 2024. [Online]. Available: https://arxiv.org/abs/2408.00714

[94] A. Hurst, A. Lerer, A. P. Goucher, A. Perelman, A. Ramesh, A. Clark, A. Ostrow, A. Welihinda, A. Hayes, A. Radford *et al.*, "Gpt-4o system card," *arXiv preprint arXiv:2410.21276*, 2024.

[95] S. Shao, Y. Bai, Y. Wang, B. Liu, and B. Liu, "Collaborative consortium of foundation models for open-world few-shot learning," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 38, no. 5, 2024, pp. 4740–4747.

[96] D. Zhu, J. Chen, K. Haydarov, X. Shen, W. Zhang, and M. Elhoseiny, "Chatgpt asks, blip-2 answers: Automatic questioning towards enriched visual descriptions," *arXiv preprint arXiv:2303.06594*, 2023.

[97] M. Xu, Z. Zhang, F. Wei, H. Hu, and X. Bai, "Side adapter network for open-vocabulary semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 2945–2954.

[98] J. Ding, N. Xue, G.-S. Xia, and D. Dai, "Decoupling zero-shot semantic segmentation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 11 583–11 592.

[99] G. Ghiasi, X. Gu, Y. Cui, and T.-Y. Lin, "Scaling open-vocabulary image segmentation with image-level labels," in *European Conference on Computer Vision*. Springer, 2022, pp. 540–557.

[100] F. Liang, B. Wu, X. Dai, K. Li, Y. Zhao, H. Zhang, P. Zhang, P. Vajda, and D. Marculescu, "Open-vocabulary semantic segmentation with mask-adapted clip," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 7061–7070.

[101] Y. Q. and et al., "Convolutions die hard: Ovs with single frozen convolutional clip," in *NeurIPS*, 2023.

[102] M. Oquab, T. Darcet, T. Moutakanni, H. V. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, R. Howes, P.-Y. Huang, H. Xu, V. Sharma, S.-W. Li, W. Galuba, M. Rabbat, M. Assran, N. Ballas, G. Synnaeve, I. Misra, H. Jegou, J. Mairal, P. Labatut, A. Joulin, and P. Bojanowski, "DINOv2: Learning robust visual features without supervision," *arXiv preprint arXiv:2304.07193*, 2023.

[103] H. Liu, C. Li, Q. Wu, and Y. J. Lee, "Visual instruction tuning," in *NeurIPS*, 2023.

[104] S. Bai, K. Chen, X. Liu, J. Wang, W. Ge, S. Song, K. Dang, P. Wang, S. Wang, J. Tang *et al.*, "Qwen2.5-vl technical report," *arXiv preprint arXiv:2502.13923*, 2025.

[105] J.-B. Alayrac, J. Donahue, P. Luc, A. Miech, I. Barr, Y. Hasson, K. Lenc, A. Mensch, K. Millican, M. Reynolds *et al.*, "Flamingo: a visual language model for few-shot learning," *Advances in neural information processing systems*, vol. 35, pp. 23 716–23 736, 2022.

[106] X. Xuan, Z. Deng, and K.-L. Ma, "ReME: A data-centric framework for training-free open-vocabulary segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025, pp. 20 954–20 965.

[107] D. Bhatt, C. Patel, H. Talsania, J. Patel, R. Vaghela, S. Pandya, K. Modi, and H. Ghayvat, "CNN variants for computer vision: History, architecture, application, challenges and future scope," *Electronics*, vol. 10, no. 20, p. 2470, 2021.

[108] Y. LeCun, L. Jackel, L. Bottou, C. Cortes, J. S. Denker, H. Drucker, I. Guyon, U. A. Muller, E. Sackinger, P. Simard *et al.*, "Learning algorithms for classification: A comparison on handwritten digit recognition," *Neural networks: the statistical mechanics perspective*, vol. 261, p. 276, 1995.

[109] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[110] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "SqueezeNet: Alexnet-level accuracy with 50x fewer parameters and <0.5 mb model size," *arXiv preprint arXiv:1602.07360*, 2016.

[111] 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*, 2017, pp. 4700–4708.

[112] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[113] X. Zhang, X. Zhou, M. Lin, and J. Sun, "Shufflenet: An extremely efficient convolutional neural network for mobile devices," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 6848–6856.

[114] Y. Hu, G. Wen, M. Luo, D. Dai, J. Ma, and Z. Yu, "Competitive inner-imaging squeeze and excitation for residual network," *arXiv preprint arXiv:1807.08920*, 2018.

[115] M. F. Aydogdu, V. Celik, and M. F. Demirci, "Comparison of three different CNN architectures for age classification," in *2017 IEEE 11th International Conference on Semantic Computing (ICSC)*. IEEE, 2017, pp. 372–377.

[116] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, and F. E. Alsaadi, "A survey of deep neural network architectures and their applications," *Neurocomputing*, vol. 234, pp. 11–26, 2017.

[117] A. Khan, A. Sohail, U. Zahoora, and A. S. Qureshi, "A survey of the recent architectures of deep convolutional neural networks," *arXiv preprint arXiv:1901.06032*, 2019.

[118] A. Canziani, A. Paszke, and E. Culurciello, "An analysis of deep neural network models for practical applications," *arXiv preprint arXiv:1605.07678*, 2016.

[119] A. Mukhopadhyay, P. Biswas, A. Agarwal, and I. Mukherjee, "Performance comparison of different cnn models for indian road dataset," in *Proceedings of the 2019 3rd International Conference on Graphics and Signal Processing*, 2019, pp. 29–33.

[120] R. C. Fong and A. Vedaldi, "Interpretable explanations of black boxes by meaningful perturbation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 3429–3437.

[121] A. Ghorbani, A. Abid, and J. Zou, "Interpretation of neural networks is fragile," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, no. 01, 2019, pp. 3681–3688.

[122] S. Murugesan, S. Malik, F. Du, E. Koh, and T. M. Lai, "Deepcompare: Visual and interactive comparison of deep learning model performance," *IEEE computer graphics and applications*, vol. 39, no. 5, pp. 47–59, 2019.

[123] D. L. Arendt, N. Nur, Z. Huang, G. Fair, and W. Dou, "Parallel embeddings: a visualization technique for contrasting learned representations," in *Proceedings of the 25th International Conference on Intelligent User Interfaces*, 2020, pp. 259–274.

[124] P. Lipton, "Contrastive explanation," *Royal Institute of Philosophy Supplements*, vol. 27, pp. 247–266, 1990.

[125] J. Woodward, "Sensitive and insensitive causation," *The Philosophical Review*, vol. 115, no. 1, pp. 1–50, 2006.

[126] G. Grinfeld, D. Lagnado, T. Gerstenberg, J. F. Woodward, and M. Usher, "Causal responsibility and robust causation," *Frontiers in Psychology*, vol. 11, p. 1069, 2020.

[127] D. S. Watson, L. Gultchin, A. Taly, and L. Floridi, "Local explanations via necessity and sufficiency: Unifying theory and practice," in *Uncertainty in Artificial Intelligence*. PMLR, 2021, pp. 1382–1392.

[128] M. Glymour, J. Pearl, and N. P. Jewell, *Causal inference in statistics: A primer*. John Wiley & Sons, 2016.

[129] H. Chockler and J. Y. Halpern, "Responsibility and blame: A structural-model approach," *Journal of Artificial Intelligence Research*, vol. 22, pp. 93–115, 2004.

[130] H. Debbi, "Causal explanation of convolutional neural networks," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2021, pp. 633–649.

[131] M. Harradon, J. Druce, and B. Ruttenberg, "Causal learning and explanation of deep neural networks via autoencoded activations," *arXiv preprint arXiv:1802.00541*, 2018.

[132] A. Bilal, A. Jourabloo, M. Ye, X. Liu, and L. Ren, "Do convolutional neural networks learn class hierarchy?" *IEEE transactions on visualization and computer graphics*, vol. 24, no. 1, pp. 152–162, 2017.

[133] M. Kahng, P. Y. Andrews, A. Kalro, and D. H. P. Chau, "A cti v is: Visual exploration of industry-scale deep neural network models," *IEEE transactions on visualization and computer graphics*, vol. 24, no. 1, pp. 88–97, 2017.

[134] Y. Ma, A. Fan, J. He, A. R. Nelakurthi, and R. Maciejewski, "A visual analytics framework for explaining and diagnosing transfer learning processes," *IEEE Transactions on Visualization and Computer Graphics*, vol. 27, no. 2, pp. 1385–1395, 2020.

[135] J. Pearl, *Causality*. Cambridge university press, 2009.

[136] J. Y. Halpern and J. Pearl, "Causes and explanations: A structural-model approach. part i: Causes," *The British journal for the philosophy of science*, 2020.

[137] L. Yao, Z. Chu, S. Li, Y. Li, J. Gao, and A. Zhang, "A survey on causal inference," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 15, no. 5, pp. 1–46, 2021.

[138] D. A. Lagnado, T. Gerstenberg, and R. Zultan, "Causal responsibility and counterfactuals," *Cognitive science*, vol. 37, no. 6, pp. 1036–1073, 2013.

[139] C. Baier, C. Dubslaff, F. Funke, S. Jantsch, R. Majumdar, J. Piribauer, and R. Ziemek, "From verification to causality-based explications," *arXiv preprint arXiv:2105.09533*, 2021.

[140] R. Kommiya Mothilal, D. Mahajan, C. Tan, and A. Sharma, "Towards unifying feature attribution and counterfactual explanations: Different means to the same end," in *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*, 2021, pp. 652–663.

[141] A. Chattopadhyay, P. Manupriya, A. Sarkar, and V. N. Balasubramanian, "Neural network attributions: A causal perspective," in *International Conference on Machine Learning*. PMLR, 2019, pp. 981–990.

[142] E. Štrumbelj and I. Kononenko, "Explaining prediction models and individual predictions with feature contributions," *Knowledge and information systems*, vol. 41, no. 3, pp. 647–665, 2014.

[143] A. Datta, S. Sen, and Y. Zick, "Algorithmic transparency via quantitative input influence: Theory and experiments with learning systems," in *2016 IEEE symposium on security and privacy (SP)*. IEEE, 2016, pp. 598–617.

[144] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," *Advances in neural information processing systems*, vol. 30, 2017.

[145] M. Sundararajan and A. Najmi, "The many shapley values for model explanation," in *International conference on machine learning*. PMLR, 2020, pp. 9269–9278.

[146] K. Aas, M. Jullum, and A. Løland, "Explaining individual predictions when features are dependent: More accurate approximations to shapley values," *Artificial Intelligence*, vol. 298, p. 103502, 2021.

[147] T. Narendra, A. Sankaran, D. Vijaykeerthy, and S. Mani, "Explaining deep learning models using causal inference," *arXiv preprint arXiv:1811.04376*, 2018.

[148] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "ImageNet large scale visual recognition challenge," *International journal of computer vision*, vol. 115, no. 3, pp. 211–252, 2015.

[149] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of machine learning research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

[150] J. D. Hunter, "Matplotlib: A 2d graphics environment," *Computing in science & engineering*, vol. 9, no. 03, pp. 90–95, 2007.

[151] M. L. Waskom, "Seaborn: statistical data visualization," *Journal of Open Source Software*, vol. 6, no. 60, p. 3021, 2021.

[152] N. Baker, H. Lu, G. Erlikhman, and P. J. Kellman, "Deep convolutional networks do not classify based on global object shape," *PLoS computational biology*, vol. 14, no. 12, p. e1006613, 2018.

[153] R. Geirhos, P. Rubisch, C. Michaelis, M. Bethge, F. A. Wichmann, and W. Brendel, "ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness," *arXiv preprint arXiv:1811.12231*, 2018.

[154] L. A. Gatys, A. S. Ecker, and M. Bethge, "Image style transfer using convolutional neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2414–2423.

[155] W. Cai and Z. Wei, "Diversity-generated image inpainting with style extraction," *arXiv preprint arXiv:1912.01834*, 2019.

[156] K. Roy and J. Mukherjee, "Image similarity measure using color histogram, color coherence vector, and sobel method," *International Journal of Science and Research (IJSR)*, vol. 2, no. 1, pp. 538–543, 2013.

[157] A. Malakar and J. Mukherjee, "Image clustering using color moments, histogram, edge and k-means clustering," *International Journal of Science and Research*, vol. 2, no. 1, pp. 532–537, 2013.

[158] S. Phetnuam and T. Yingthawornsuk, "Classfication of categorized kmutt-bkt's landscape images using rgb color feature," in *2018 14th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)*. IEEE, 2018, pp. 327–331.

[159] T. G. Falleti and J. F. Lynch, "Context and causal mechanisms in political analysis," *Comparative political studies*, vol. 42, no. 9, pp. 1143–1166, 2009.

[160] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[161] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona, "Caltech-ucsd birds 200," 2010.

[162] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[163] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.

[164] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," 2017.

[165] Wikipedia contributors, "Taxonomic rank — Wikipedia, the free encyclopedia," 2022, [Online; accessed 19-October-2022]. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Taxonomic_rank&oldid=1116635228

[166] X. Dong, J. Han, D. Chen, J. Liu, H. Bian, Z. Ma, H. Li, X. Wang, W. Zhang, and N. Yu, "Robust superpixel-guided attentional adversarial attack," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 12 895–12 904.

[167] J. Wang, Z. Yin, J. Jiang, and Y. Du, "Attention-guided black-box adversarial attacks with large-scale multiobjective evolutionary optimization," *International Journal of Intelligent Systems*, 2022.

[168] T. Xiang, H. Liu, S. Guo, T. Zhang, and X. Liao, "Local black-box adversarial attacks: A query efficient approach," *arXiv preprint arXiv:2101.01032*, 2021.

[169] Z. Dai, S. Liu, K. Tang, and Q. Li, "Saliency attack: Towards imperceptible black-box adversarial attack," *arXiv preprint arXiv:2206.01898*, 2022.

[170] J. Adebayo, J. Gilmer, M. Muelly, I. Goodfellow, M. Hardt, and B. Kim, "Sanity checks for saliency maps," *Advances in neural information processing systems*, vol. 31, 2018.

[171] S. Singla, M. Moayeri, and S. Feizi, "Core risk minimization using salient imagenet." arXiv, 2022.

[172] M. A. Madaio, L. Stark, J. Wortman Vaughan, and H. Wallach, "Co-designing checklists to understand organizational challenges and opportunities around fairness in ai," in *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, 2020, pp. 1–14.

[173] N. C. F. Codella, D. Gutman, M. E. Celebi, B. Helba, M. A. Marchetti, S. W. Dusza, A. Kalloo, K. Liopyris, N. Mishra, H. Kittler, and A. Halpern, "Skin lesion analysis toward melanoma detection: A challenge at the 2017 international symposium on biomedical imaging (isbi), hosted by the international skin imaging collaboration (isic)," in *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*, 2018, pp. 168–172.

[174] S. Sagawa*, P. W. Koh*, T. B. Hashimoto, and P. Liang, "Distributionally robust neural networks," in *International Conference on Learning Representations*, 2020. [Online]. Available: https://openreview.net/forum?id=ryxGuJrFvS

[175] S. Lee, A. Payani, and D. H. P. Chau, "Towards mitigating spurious correlations in image classifiers with simple yes-no feedback."

[176] A. S. Ross, M. C. Hughes, and F. Doshi-Velez, "Right for the right reasons: Training differentiable models by constraining their explanations," in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, 2017, pp. 2662–2670. [Online]. Available: https://doi.org/10.24963/ijcai.2017/371

[177] Y. Gao, T. S. Sun, L. Zhao, and S. R. Hong, "Aligning eyes between humans and deep neural network through interactive attention alignment," *Proc. ACM Hum.-Comput. Interact.*, vol. 6, no. CSCW2, nov 2022. [Online]. Available: https://doi.org/10.1145/3555590

[178] S. Rao, M. Böhle, A. Parchami-Araghi, and B. Schiele, "Using explanations to guide models," *arXiv preprint arXiv:2303.11932*, 2023.

[179] G. Barash, E. Farchi, I. Jayaraman, O. Raz, R. Tzoref-Brill, and M. Zalmanovici, "Bridging the gap between ML solutions and their business requirements using feature interactions," in *Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, 2019, pp. 1048–1058.

[180] N. Sohoni, J. Dunnmon, G. Angus, A. Gu, and C. Ré, "No subclass left behind: Fine-grained robustness in coarse-grained classification problems," *Advances in Neural Information Processing Systems*, vol. 33, pp. 19 339–19 352, 2020.

[181] G. d'Eon, J. d'Eon, J. R. Wright, and K. Leyton-Brown, "The spotlight: A general method for discovering systematic errors in deep learning models," in *2022 ACM Conference on Fairness, Accountability, and Transparency*, 2022, pp. 1962–1981.

[182] A. Krishnakumar, V. Prabhu, S. Sudhakar, and J. Hoffman, "Udis: Unsupervised discovery of bias in deep visual recognition models," in *British Machine Vision Conference (BMVC)*, vol. 1, no. 2, 2021, p. 3.

[183] S. Lee, Z. J. Wang, J. Hoffman, and D. H. P. Chau, "Viscuit: Visual auditor for bias in cnn image classifier," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 21 475–21 483.

[184] S. Lai, W. Luan, and J. Tao, "Explore your network in minutes: A rapid prototyping toolkit for understanding neural networks with visual analytics," *IEEE Transactions on Visualization and Computer Graphics*, 2023.

[185] J. Wang, S. Liu, and W. Zhang, "Visual analytics for machine learning: A data perspective survey," *IEEE Transactions on Visualization and Computer Graphics*, 2024.

[186] V. Prasad, R. J. van Sloun, S. van den Elzen, A. Vilanova, and N. Pezzotti, "The transform-and-perform framework: Explainable deep learning beyond classification," *IEEE Transactions on Visualization and Computer Graphics*, vol. 30, no. 2, pp. 1502–1515, 2022.

[187] J. Huang, A. Mishra, B.-C. Kwon, and C. Bryan, "Conceptexplainer: Understanding the mental model of deep learning algorithms via interactive concept-based explanations," 2022.

[188] C. Chen, J. Yuan, Y. Lu, Y. Liu, H. Su, S. Yuan, and S. Liu, "Oodanalyzer: Interactive analysis of out-of-distribution samples," *IEEE transactions on visualization and computer graphics*, vol. 27, no. 7, pp. 3335–3349, 2020.

[189] Z. Li, X. Wang, W. Yang, J. Wu, Z. Zhang, Z. Liu, M. Sun, H. Zhang, and S. Liu, "A unified understanding of deep nlp models for text classification," *IEEE Transactions on Visualization and Computer Graphics*, vol. 28, no. 12, pp. 4980–4994, 2022.

[190] S. Robertson, Z. J. Wang, D. Moritz, M. B. Kery, and F. Hohman, "Angler: Helping machine translation practitioners prioritize model improvements," in *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, 2023, pp. 1–20.

[191] C. Chen, Y. Guo, F. Tian, S. Liu, W. Yang, Z. Wang, J. Wu, H. Su, H. Pfister, and S. Liu, "A unified interactive model evaluation for classification, object detection, and instance segmentation in computer vision," *IEEE Transactions on Visualization and Computer Graphics*, 2023.

[192] X. Zhang, J. H. Piazentin Ono, W. He, L. Gou, M. Sachan, K.-L. Ma, and L. Ren, "Slicing, chatting, and refining: A concept-based approach for machine learning model validation with conceptslicer," in *Proceedings of the 29th International Conference on Intelligent User Interfaces*, 2024, pp. 274–287.

[193] W. Yang, M. Liu, Z. Wang, and S. Liu, "Foundation models meet visualizations: Challenges and opportunities," *Computational Visual Media*, pp. 1–26, 2024.

[194] Z. Jin, X. Wang, F. Cheng, C. Sun, Q. Liu, and H. Qu, "ShortcutLens: A visual analytics approach for exploring shortcuts in natural language understanding dataset," *IEEE Transactions on Visualization and Computer Graphics*, 2023.

[195] X. Zhang, X. Xuan, A. Dima, T. Sexton, and K.-L. Ma, "Labelvizier: Interactive validation and relabeling for technical text annotations," in *2023 IEEE 16th Pacific Visualization Symposium (PacificVis)*. IEEE, 2023, pp. 167–176.

[196] M. Kahng, D. Fang, and D. H. Chau, "Visual exploration of machine learning results using data cube analysis," in *Proceedings of the Workshop on Human-In-the-Loop Data Analytics*, 2016, pp. 1–6.

[197] Á. A. Cabrera, W. Epperson, F. Hohman, M. Kahng, J. Morgenstern, and D. H. Chau, "FairVis: Visual analytics for discovering intersectional bias in machine learning," in *Proceedings of 2019 IEEE Conference on Visual Analytics Science and Technology (VAST)*. IEEE, 2019, pp. 46–56.

[198] T. Wu, M. T. Ribeiro, J. Heer, and D. Weld, "Errudite: Scalable, reproducible, and testable error analysis," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019.

[199] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Transactions on knowledge and data engineering*, vol. 21, no. 9, pp. 1263–1284, 2009.

[200] S. A. Taghanaki, K. Choi, A. H. Khasahmadi, and A. Goyal, "Robust representation learning via perceptual similarity metrics," in *International Conference on Machine Learning*. PMLR, 2021, pp. 10 043–10 053.

[201] J. Nam, H. Cha, S. Ahn, J. Lee, and J. Shin, "Learning from failure: De-biasing classifier from biased classifier," *Advances in Neural Information Processing Systems*, vol. 33, pp. 20 673–20 684, 2020.

[202] E. Creager, J.-H. Jacobsen, and R. Zemel, "Environment inference for invariant learning," in *International Conference on Machine Learning*. PMLR, 2021, pp. 2189–2200.

[203] F. Ahmed, Y. Bengio, H. Van Seijen, and A. Courville, "Systematic generalisation with group invariant predictions," in *International Conference on Learning Representations*, 2020.

[204] Y. Gao, T. S. Sun, L. Zhao, and S. R. Hong, "Aligning eyes between humans and deep neural network through interactive attention alignment," *Proceedings of the ACM on Human-Computer Interaction*, vol. 6, no. CSCW2, pp. 1–28, 2022.

[205] M. Srivastava, T. Hashimoto, and P. Liang, "Robustness to spurious correlations via human annotations," in *International Conference on Machine Learning*. PMLR, 2020, pp. 9109–9119.

[206] A. Bontempelli, S. Teso, K. Tentori, F. Giunchiglia, and A. Passerini, "Concept-level debugging of part-prototype networks," in *The Eleventh International Conference on Learning Representations*, 2023. [Online]. Available: https://openreview.net/forum?id=oiwXWPDTyNk

[207] Y. Yang, B. Nushi, H. Palangi, and B. Mirzasoleiman, "Mitigating spurious correlations in multi-modal models during fine-tuning," in *Proceedings of the 40th International Conference on Machine Learning*, ser. ICML'23. JMLR.org, 2023.

[208] A. Mahendran and A. Vedaldi, "Understanding deep image representations by inverting them," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 5188–5196.

[209] S. Sikdar, P. Bhattacharya, and K. Heese, "Integrated directional gradients: Feature interaction attribution for neural nlp models," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2021, pp. 865–878.

[210] J. D. Janizek, P. Sturmfels, and S.-I. Lee, "Explaining explanations: Axiomatic feature interactions for deep networks," *Journal of Machine Learning Research*, vol. 22, no. 104, pp. 1–54, 2021.

[211] L. Rieger, C. Singh, W. Murdoch, and B. Yu, "Interpretations are useful: penalizing explanations to align neural networks with prior knowledge," in *International conference on machine learning*. PMLR, 2020, pp. 8116–8126.

[212] C. Mao, K. Xia, J. Wang, H. Wang, J. Yang, E. Bareinboim, and C. Vondrick, "Causal transportability for visual recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 7521–7531.

[213] J. Zhuang, J. Cai, R. Wang, J. Zhang, and W. Zheng, "Care: Class attention to regions of lesion for classification on imbalanced data," in *International Conference on Medical Imaging with Deep Learning*. PMLR, 2019, pp. 588–597.

[214] Y. Yang, B. Nushi, H. Palangi, and B. Mirzasoleiman, "Mitigating spurious correlations in multi-modal models during fine-tuning," *arXiv preprint arXiv:2304.03916*, 2023.

[215] L. McInnes, J. Healy, N. Saul, and L. Großberger, "Umap: Uniform manifold approximation and projection," *Journal of Open Source Software*, vol. 3, no. 29, 2018.

[216] K. Pal and M. Sharma, "Performance evaluation of non-linear techniques umap and t-sne for data in higher dimensional topological space," in *2020 fourth international conference on I-SMAC (IoT in social, mobile, analytics and cloud)(I-SMAC)*. IEEE, 2020, pp. 1106–1110.

[217] C. B. Barber, D. P. Dobkin, and H. Huhdanpaa, "The quickhull algorithm for convex hulls," *ACM Transactions on Mathematical Software (TOMS)*, vol. 22, no. 4, pp. 469–483, 1996.

[218] S. Carter, Z. Armstrong, L. Schubert, I. Johnson, and C. Olah, "Activation atlas," *Distill*, 2019, https://distill.pub/2019/activation-atlas.

[219] A. Nguyen, J. Yosinski, and J. Clune, "Multifaceted feature visualization: Uncovering the different types of features learned by each neuron in deep neural networks," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2016.

[220] T. Fel, T. Boissin, V. Boutin, A. Picard, P. Novello, J. Colin, D. Linsley, T. Rousseau, R. Cadène, and T. S. Laurent Gardes, "Unlocking feature visualization for deeper networks with magnitude constrained optimization," *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

[221] D. Zhou, O. Bousquet, T. Lal, J. Weston, and B. Schölkopf, "Learning with local and global consistency," *Advances in neural information processing systems*, vol. 16, 2003.

[222] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

[223] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The caltech-ucsd birds-200-2011 dataset," 2011.

[224] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, "Places: A 10 million image database for scene recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 6, pp. 1452–1464, 2017.

[225] S. Joshi, Y. Yang, Y. Xue, W. Yang, and B. Mirzasoleiman, "Towards mitigating spurious correlations in the wild: A benchmark & a more realistic dataset," *arXiv preprint arXiv:2306.11957*, 2023.

[226] 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.

[227] 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*, 2021, pp. 10 012–10 022.

[228] H. Wang, Z. Wang, M. Du, F. Yang, Z. Zhang, S. Ding, P. Mardziel, and X. Hu, "Score-cam: Score-weighted visual explanations for convolutional neural networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, 2020, pp. 24–25.

[229] Z. Chen, Y. Deng, Y. Wu, Q. Gu, and Y. Li, "Towards understanding the mixture-of-experts layer in deep learning," *Advances in neural information processing systems*, vol. 35, pp. 23 049–23 062, 2022.

[230] S. Jelassi and Y. Li, "Towards understanding how momentum improves generalization in deep learning," in *International Conference on Machine Learning*. PMLR, 2022, pp. 9965–10 040.

[231] Y. Deng, Y. Yang, B. Mirzasoleiman, and Q. Gu, "Robust learning with progressive data expansion against spurious correlation," in *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. [Online]. Available: https://openreview.net/forum?id=9QEVJ9qm46

[232] Y. Xue, A. Payani, Y. Yang, and B. Mirzasoleiman, "Eliminating spurious correlations from pre-trained models via data mixing," *arXiv preprint arXiv:2305.14521*, 2023.

[233] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 3730–3738.

[234] X. Zhang, Y. He, R. Xu, H. Yu, Z. Shen, and P. Cui, "Nico++: Towards better benchmarking for domain generalization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 16 036–16 047.

[235] K. Xiao, L. Engstrom, A. Ilyas, and A. Madry, "Noise or signal: The role of image backgrounds in object recognition," *ArXiv preprint arXiv:2006.09994*, 2020.

[236] S. Wu, M. Yuksekgonul, L. Zhang, and J. Zou, "Discover and cure: Concept-aware mitigation of spurious correlation," in *Proceedings of the 40th International Conference on Machine Learning*, ser. ICML'23. JMLR.org, 2023.

[237] Z. Peng, W. Wang, L. Dong, Y. Hao, S. Huang, S. Ma, and F. Wei, "Kosmos-2: Grounding multimodal large language models to the world," *arXiv preprint arXiv:2306.14824*, 2023.

[238] H. Liu, C. Li, Q. Wu, and Y. J. Lee, "Visual instruction tuning," *Advances in neural information processing systems*, vol. 36, 2024.

[239] J. Bernard, M. Hutter, M. Zeppelzauer, D. Fellner, and M. Sedlmair, "Comparing visual-interactive labeling with active learning: An experimental study," *IEEE transactions on visualization and computer graphics*, vol. 24, no. 1, pp. 298–308, 2017.

[240] M. Chegini, J. Bernard, P. Berger, A. Sourin, K. Andrews, and T. Schreck, "Interactive labelling of a multivariate dataset for supervised machine learning using linked visualisations, clustering, and active learning," *Visual Informatics*, vol. 3, no. 1, pp. 9–17, 2019.

[241] M. Khayat, M. Karimzadeh, J. Zhao, and D. S. Ebert, "Vassl: A visual analytics toolkit for social spambot labeling," *IEEE transactions on visualization and computer graphics*, vol. 26, no. 1, pp. 874–883, 2019.

[242] Y. Zhang, B. Coecke, and M. Chen, "On the cost of interactions in interactive visual machine learning," in *2019 IEEE workshop on EValuation of Interactive VisuAl Machine Learning systems (Eviva-ML). IEEE*, 2019, pp. 1–5.

[243] Y. Wang, J. Zhao, J. Hong, R. G. Askin, and R. Maciejewski, "A simulation-based approach for quantifying the impact of interactive label correction for machine learning," *IEEE Transactions on Visualization and Computer Graphics*, pp. 1–17, 2024.

[244] M. Choi, C. Park, S. Yang, Y. Kim, J. Choo, and S. R. Hong, "Aila: Attentive interactive labeling assistant for document classification through attention-based deep neural networks," in *Proceedings of the 2019 CHI conference on human factors in computing systems*, 2019, pp. 1–12.

[245] Y. Zhang, Y. Wang, H. Zhang, B. Zhu, S. Chen, and D. Zhang, "Onelabeler: A flexible system for building data labeling tools," in *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, 2022, pp. 1–22.

[246] J. Bernard, M. Zeppelzauer, M. Sedlmair, and W. Aigner, "A unified process for visual-interactive labeling," in *Proceedings of the EuroVis Workshop on Visual Analytics*, 2017, pp. 73–77.

[247] J. Moehrmann, S. Bernstein, T. Schlegel, G. Werner, and G. Heidemann, "Improving the usability of hierarchical representations for interactively labeling large image data sets," in *Human-Computer Interaction. Design and Development Approaches: 14th International Conference, HCI International 2011, Orlando, FL, USA, July 9-14, 2011, Proceedings, Part I 14*. Springer, 2011, pp. 618–627.

[248] M. Desmond, M. Muller, Z. Ashktorab, C. Dugan, E. Duesterwald, K. Brimijoin, C. Finegan-Dollak, M. Brachman, A. Sharma, N. N. Joshi *et al.*, "Increasing the speed and accuracy of data labeling through an ai assisted interface," in *26th International Conference on Intelligent User Interfaces*, 2021, pp. 392–401.

[249] M. Chegini, J. Bernard, J. Cui, F. Chegini, A. Sourin, K. Andrews, and T. Schreck, "Interactive visual labelling versus active learning: an experimental comparison," *Frontiers of Information Technology & Electronic Engineering*, vol. 21, pp. 524–535, 2020.

[250] B. Höferlin, R. Netzel, M. Höferlin, D. Weiskopf, and G. Heidemann, "Inter-active learning of ad-hoc classifiers for video visual analytics," in *2012 IEEE conference on visual analytics science and technology (VAST)*. IEEE, 2012, pp. 23–32.

[251] C. Chen, J. Wu, X. Wang, S. Xiang, S.-H. Zhang, Q. Tang, and S. Liu, "Towards better caption supervision for object detection," *IEEE Transactions on Visualization and Computer Graphics*, vol. 28, no. 4, pp. 1941–1954, 2021.

[252] W. Yang, X. Ye, X. Zhang, L. Xiao, J. Xia, Z. Wang, J. Zhu, H. Pfister, and S. Liu, "Diagnosing ensemble few-shot classifiers," *IEEE Transactions on Visualization and Computer Graphics*, vol. 28, no. 9, pp. 3292–3306, 2022.

[253] C. Vajiac, D. H. Chau, A. Olligschlaeger, R. Mackenzie, P. Nair, M.-C. Lee, Y. Li, N. Park, R. Rabbany, and C. Faloutsos, "Trafficvis: visualizing organized activity and spatio-temporal patterns for detecting and labeling human trafficking," *IEEE Transactions on Visualization and Computer Graphics*, vol. 29, no. 1, pp. 53–62, 2022.

[254] C. Chen, Z. Wang, J. Wu, X. Wang, L.-Z. Guo, Y.-F. Li, and S. Liu, "Interactive graph construction for graph-based semi-supervised learning," *IEEE Transactions on Visualization and Computer Graphics*, vol. 27, no. 9, pp. 3701–3716, 2021.

[255] W. Yang, Y. Guo, J. Wu, Z. Wang, L.-Z. Guo, Y.-F. Li, and S. Liu, "Interactive reweighting for mitigating label quality issues," *IEEE Transactions on Visualization and Computer Graphics*, 2023.

[256] C. Chen, J. Chen, W. Yang, H. Wang, J. Knittel, X. Zhao, S. Koch, T. Ertl, and S. Liu, "Enhancing single-frame supervision for better temporal action localization," *IEEE Transactions on Visualization and Computer Graphics*, 2024.

[257] S. Sharifi Noorian, S. Qiu, B. Sayin, A. Balayn, U. Gadiraju, J. Yang, and A. Bozzon, "Perspective: leveraging human understanding for identifying and characterizing image atypicality," in *Proceedings of the 28th International Conference on Intelligent User Interfaces*, 2023, pp. 650–663.

[258] A. Bäuerle, H. Neumann, and T. Ropinski, "Classifier-guided visual correction of noisy labels for image classification tasks," in *Computer graphics forum*, vol. 39, no. 3. Wiley Online Library, 2020, pp. 195–205.

[259] S. Liu, C. Chen, Y. Lu, F. Ouyang, and B. Wang, "An interactive method to improve crowdsourced annotations," *IEEE transactions on visualization and computer graphics*, vol. 25, no. 1, pp. 235–245, 2018.

[260] S. Xiang, X. Ye, J. Xia, J. Wu, Y. Chen, and S. Liu, "Interactive correction of mislabeled training data," in *2019 IEEE Conference on Visual Analytics Science and Technology (VAST)*. IEEE, 2019, pp. 57–68.

[261] M. N. Hoque, W. He, A. K. Shekar, L. Gou, and L. Ren, "Visual concept programming: A visual analytics approach to injecting human intelligence at scale," *IEEE Transactions on Visualization and Computer Graphics*, vol. 29, no. 1, pp. 74–83, 2022.

[262] D. Huynh, J. Kuen, Z. Lin, J. Gu, and E. Elhamifar, "Open-vocabulary instance segmentation via robust cross-modal pseudo-labeling," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 7020–7031.

[263] M. Xu, Z. Zhang, F. Wei, Y. Lin, Y. Cao, H. Hu, and X. Bai, "A simple baseline for open-vocabulary semantic segmentation with pre-trained vision-language model," in *European Conference on Computer Vision*. Springer, 2022, pp. 736–753.

[264] J. Xu, S. Liu, A. Vahdat, W. Byeon, X. Wang, and S. De Mello, "Open-vocabulary panoptic segmentation with text-to-image diffusion models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 2955–2966.

[265] X. Zou, Z.-Y. Dou, J. Yang, Z. Gan, L. Li, C. Li, X. Dai, H. Behl, J. Wang, L. Yuan *et al.*, "Generalized decoding for pixel, image, and language," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 15 116–15 127.

[266] C. Zhou, C. C. Loy, and B. Dai, "Extract free dense labels from clip," in *European Conference on Computer Vision*. Springer, 2022, pp. 696–712.

[267] M. Lan, C. Chen, Y. Ke, X. Wang, L. Feng, and W. Zhang, "Proxyclip: Proxy attention improves clip for open-vocabulary segmentation," in *ECCV*, 2024.

[268] T. Shao, Z. Tian, H. Zhao, and J. Su, "Explore the potential of clip for training-free open vocabulary semantic segmentation," *arXiv preprint arXiv:2407.08268*, 2024.

[269] G. Shin, W. Xie, and S. Albanie, "Reco: Retrieve and co-segment for zero-shot transfer," *Advances in Neural Information Processing Systems*, vol. 35, pp. 33 754–33 767, 2022.

[270] L. Karazija, I. Laina, A. Vedaldi, and C. Rupprecht, "Diffusion models for open-vocabulary segmentation," in *European Conference on Computer Vision*. Springer, 2025, pp. 299–317.

[271] L. Barsellotti, R. Amoroso, L. Baraldi, and R. Cucchiara, "Fossil: Free open-vocabulary semantic segmentation through synthetic references retrieval," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2024, pp. 1464–1473.

[272] Z. Gui, S. Sun, R. Li, J. Yuan, Z. An, K. Roth, A. Prabhu, and P. Torr, "kNN-CLIP: Retrieval enables training-free segmentation on continually expanding large vocabularies," *Transactions on Machine Learning Research*, 2024.

[273] Y. Wang, R. Sun, N. Luo, Y. Pan, and T. Zhang, "Image-to-image matching via foundation models: A new perspective for open-vocabulary semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 3952–3963.

[274] C. Wu, Z. Lin, S. Cohen, T. Bui, and S. Maji, "Phrasecut: Language-based image segmentation in the wild," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 10 216–10 225.

[275] Explosion AI, "spacy: Industrial-strength natural language processing in python," 2020, software available from https://spacy.io/.

[276] L. McInnes, J. Healy, S. Astels *et al.*, "hdbscan: Hierarchical density based clustering." *J. Open Source Softw.*, vol. 2, no. 11, pp. 205–206, 2017.

[277] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne." *Journal of machine learning research*, vol. 9, no. 11, 2008.

[278] M. Ester, H.-P. Kriegel, J. Sander, X. Xu *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *kdd*, vol. 96, 1996, pp. 226–231.

[279] Y. Li, H. Wang, Y. Duan, and X. Li, "Clip surgery for better explainability with enhancement in open-vocabulary tasks," *arXiv preprint arXiv:2304.05653*, 2023.

[280] C. Wei, C. Liu, S. Qiao, Z. Zhang, A. Yuille, and J. Yu, "De-diffusion makes text a strong cross-modal interface," *arXiv preprint arXiv:2311.00618*, 2023.

[281] S. Shetiya, I. P. Swift, A. Asudeh, and G. Das, "Fairness-aware range queries for selecting unbiased data," in *2022 IEEE 38th International Conference on Data Engineering (ICDE)*. IEEE, 2022, pp. 1423–1436.

[282] R. Zhang, W. Zhang, R. Fang, P. Gao, K. Li, J. Dai, Y. Qiao, and H. Li, "Tip-Adapter: Training-free adaption of clip for few-shot classification." Berlin, Heidelberg: Springer-Verlag, 2022, p. 493–510. [Online]. Available: https://doi.org/10.1007/978-3-031-19833-5_29

[283] A. Lex, N. Gehlenborg, H. Strobelt, R. Vuillemot, and H. Pfister, "UpSet: visualization of intersecting sets," *IEEE transactions on visualization and computer graphics*, vol. 20, no. 12, pp. 1983–1992, 2014.

[284] H. Caesar, J. Uijlings, and V. Ferrari, "Coco-stuff: Thing and stuff classes in context," in *Computer vision and pattern recognition (CVPR), 2018 IEEE conference on*. IEEE, 2018.

[285] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 3213–3223.

[286] F. Wang, J. Mei, and A. Yuille, "Sclip: Rethinking self-attention for dense vision-language inference," in *European Conference on Computer Vision*. Springer, 2024, pp. 315–332.

[287] B. Zhou, H. Zhao, X. Puig, T. Xiao, S. Fidler, A. Barriuso, and A. Torralba, "Semantic understanding of scenes through the ade20k dataset," *International Journal of Computer Vision*, vol. 127, pp. 302–321, 2019.

[288] H. Caesar, J. Uijlings, and V. Ferrari, "Coco-stuff: Thing and stuff classes in context," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 1209–1218.

[289] L. Barsellotti, R. Amoroso, M. Cornia, L. Baraldi, and R. Cucchiara, "Training-free open-vocabulary segmentation with offline diffusion-augmented prototype generation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 3689–3698.

[290] H. Luo, J. Bao, Y. Wu, X. He, and T. Li, "Segclip: Patch aggregation with learnable centers for open-vocabulary semantic segmentation," in *International Conference on Machine Learning*. PMLR, 2023, pp. 23 033–23 044.

[291] J. Mukhoti, T.-Y. Lin, O. Poursaeed, R. Wang, A. Shah, P. H. Torr, and S.-N. Lim, "Open vocabulary semantic segmentation with patch aligned contrastive learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 19 413–19 423.

[292] M. Yi, Q. Cui, H. Wu, C. Yang, O. Yoshie, and H. Lu, "A simple framework for text-supervised semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 7071–7080.

[293] J. Cha, J. Mun, and B. Roh, "Learning to generate text-grounded mask for open-world semantic segmentation from only image-text pairs," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 11 165–11 174.

[294] J. Chen, D. Zhu, G. Qian, B. Ghanem, Z. Yan, C. Zhu, F. Xiao, S. C. Culatana, and M. Elhoseiny, "Exploring open-vocabulary semantic segmentation from clip vision encoder distillation only," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 699–710.

[295] J. Xu, J. Hou, Y. Zhang, R. Feng, Y. Wang, Y. Qiao, and W. Xie, "Learning open-vocabulary semantic segmentation models from natural language supervision," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 2935–2944.

[296] J.-J. Wu, A. C.-H. Chang, C.-Y. Chuang, C.-P. Chen, Y.-L. Liu, M.-H. Chen, H.-N. Hu, Y.-Y. Chuang, and Y.-Y. Lin, "Image-text co-decomposition for text-supervised semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 26 794–26 803.

[297] S. Sun, R. Li, P. Torr, X. Gu, and S. Li, "Clip as rnn: Segment countless visual concepts without training endeavor," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 13 171–13 182.

[298] J. Wang, X. Li, J. Zhang, Q. Xu, Q. Zhou, Q. Yu, L. Sheng, and D. Xu, "Diffusion model is secretly a training-free open vocabulary semantic segmenter," *arXiv preprint arXiv:2309.02773*, 2023.

[299] J. Li, P. Chen, S. Qian, and J. Jia, "Tagclip: Improving discrimination ability of open-vocabulary semantic segmentation," *arXiv preprint arXiv:2304.07547*, 2023.

[300] Y. Guo, D. Stutz, and B. Schiele, "Robustifying token attention for vision transformers," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 17 557–17 568.

[301] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *International journal of computer vision*, vol. 59, pp. 167–181, 2004.

[302] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 10 684–10 695.

[303] L. Fan, K. Chen, D. Krishnan, D. Katabi, P. Isola, and Y. Tian, "Scaling laws of synthetic images for model training... for now," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 7382–7392.

[304] K. Singh, T. Navaratnam, J. Holmer, S. Schaub-Meyer, and S. Roth, "Is synthetic data all we need? benchmarking the robustness of models trained with synthetic images," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 2505–2515.

[305] C. Schuhmann, R. Beaumont, R. Vencu, C. W. Gordon, R. Wightman, M. Cherti, T. Coombes, A. Katta, C. Mullis, M. Wortsman, P. Schramowski, S. R. Kundurthy, K. Crowson, L. Schmidt, R. Kaczmarczyk, and J. Jitsev, "LAION-5b: An open large-scale dataset for training next generation image-text models," in *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022. [Online]. Available: https://openreview.net/forum?id=M3Y74vmsMcY

[306] F. Radenovic, A. Dubey, A. Kadian, T. Mihaylov, S. Vandenhende, Y. Patel, Y. Wen, V. Ramanathan, and D. Mahajan, "Filtering, distillation, and hard negatives for vision-language pre-training," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 6967–6977.

[307] T. Nguyen, S. Y. Gadre, G. Ilharco, S. Oh, and L. Schmidt, "Improving multimodal datasets with image captioning," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[308] 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*, vol. 30, 2017.

[309] P. Rewatbowornwong, N. Chatthee, E. Chuangsuwanich, and S. Suwajanakorn, "Zero-guidance segmentation using zero segment labels," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 1162–1172.

[310] W. Bousselham, F. Petersen, V. Ferrari, and H. Kuehne, "Grounding everything: Emerging localization properties in vision-language transformers," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 3828–3837.

[311] M. Wysoczańska, M. Ramamonjisoa, T. Trzciński, and O. Siméoni, "Clip-diy: Clip dense inference yields open-vocabulary semantic segmentation for-free," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2024, pp. 1403–1413.

[312] J. Luo, S. Khandelwal, L. Sigal, and B. Li, "Emergent open-vocabulary semantic segmentation from off-the-shelf vision-language models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 4029–4040.

[313] Q. Nguyen, T. Vu, A. Tran, and K. Nguyen, "Dataset diffusion: Diffusion-based synthetic data generation for pixel-level semantic segmentation," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[314] A. Fang, A. M. Jose, A. Jain, L. Schmidt, A. T. Toshev, and V. Shankar, "Data filtering networks," in *The Twelfth International Conference on Learning Representations*, 2024. [Online]. Available: https://openreview.net/forum?id=KAk6ngZ09F

[315] A. Mahmoud, M. Elhoushi, A. Abbas, Y. Yang, N. Ardalani, H. Leather, and A. S. Morcos, "Sieve: Multimodal dataset pruning using image captioning models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 22 423–22 432.

[316] B. Xiao, H. Wu, W. Xu, X. Dai, H. Hu, Y. Lu, M. Zeng, C. Liu, and L. Yuan, "Florence-2: Advancing a unified representation for a variety of vision tasks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 4818–4829.

[317] L. Cao, B. Zhang, C. Chen, Y. Yang, X. Du, W. Zhang, Z. Lu, and Y. Zheng, "Less is more: Removing text-regions improves clip training efficiency and robustness," *arXiv preprint arXiv:2305.05095*, 2023.

[318] S. Y. Gadre, G. Ilharco, A. Fang, J. Hayase, G. Smyrnis, T. Nguyen, R. Marten, M. Wortsman, D. Ghosh, J. Zhang *et al.*, "Datacomp: In search of the next generation of multimodal datasets," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[319] Z. Lai, H. Zhang, B. Zhang, W. Wu, H. Bai, A. Timofeev, X. Du, Z. Gan, J. Shan, C.-N. Chuah *et al.*, "Veclip: Improving clip training via visual-enriched captions," in *European Conference on Computer Vision*. Springer, 2025, pp. 111–127.

[320] D. Zhang, F. Liu, and Q. Tang, "Corrclip: Reconstructing correlations in clip with off-the-shelf foundation models for open-vocabulary semantic segmentation," *arXiv preprint arXiv:2411.10086*, 2024.

[321] C. Jia, Y. Yang, Y. Xia, Y.-T. Chen, Z. Parekh, H. Pham, Q. Le, Y.-H. Sung, Z. Li, and T. Duerig, "Scaling up visual and vision-language representation learning with noisy text supervision," in *International conference on machine learning*. PMLR, 2021, pp. 4904–4916.

[322] J. Li, D. Li, S. Savarese, and S. Hoi, "Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models," in *International conference on machine learning*. PMLR, 2023, pp. 19 730–19 742.

[323] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat *et al.*, "Gpt-4 technical report," *arXiv preprint arXiv:2303.08774*, 2023.

[324] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin, "Emerging properties in self-supervised vision transformers," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 9650–9660.

[325] M. Oquab, T. Darcet, T. Moutakanni, H. V. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. HAZIZA, F. Massa, A. El-Nouby, M. Assran, N. Ballas, W. Galuba, R. Howes, P.-Y. Huang, S.-W. Li, I. Misra, M. Rabbat, V. Sharma, G. Synnaeve, H. Xu, H. Jegou, J. Mairal, P. Labatut, A. Joulin, and P. Bojanowski, "DINOv2: Learning robust visual features without supervision," *Transactions on Machine Learning Research*, 2024. [Online]. Available: https://openreview.net/forum?id=a68SUt6zFt

[326] X. Lai, Z. Tian, Y. Chen, Y. Li, Y. Yuan, S. Liu, and J. Jia, "LISA: Reasoning segmentation via large language model," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 9579–9589.

[327] R. Zhang, W. Zhang, R. Fang, P. Gao, K. Li, J. Dai, Y. Qiao, and H. Li, "Tip-adapter: Training-free adaption of clip for few-shot classification," in *European conference on computer vision*. Springer, 2022, pp. 493–510.

[328] S. Hajimiri, I. B. Ayed, and J. Dolz, "Pay attention to your neighbours: Training-free open-vocabulary semantic segmentation," *arXiv preprint arXiv:2404.08181*, 2024.

[329] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International journal of computer vision*, vol. 88, pp. 303–338, 2010.

[330] R. Mottaghi, X. Chen, X. Liu, N.-G. Cho, S.-W. Lee, S. Fidler, R. Urtasun, and A. Yuille, "The role of context for object detection and semantic segmentation in the wild," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 891–898.

[331] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*. Springer, 2014, pp. 740–755.

[332] P. Krähenbühl and V. Koltun, "Efficient inference in fully connected crfs with gaussian edge potentials," *Advances in neural information processing systems*, vol. 24, 2011.

[333] A. S. Wannenwetsch and S. Roth, "Probabilistic pixel-adaptive refinement networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 642–11 651.