VISUAL AND ALGORITHMIC EXPLANATIONS TO FORTIFY AI SAFETY

A Dissertation
Presented to
The Academic Faculty

By

Seongmin Lee

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in Computer Science

Georgia Institute of Technology

August 2026

© Seongmin Lee 2026

VISUAL AND ALGORITHMIC EXPLANATIONS TO FORTIFY AI SAFETY

Thesis committee:

Dr. Duen Horng (Polo) Chau
School of Computational Science & Engineering
Georgia Institute of Technology

Dr. Judy Hoffman
Donald Bren School of Information and Computer Sciences
University of California, Irvine

Dr. Alex Endert
School of Interactive Computing
Georgia Institute of Technology

Dr. Oliver Brdiczka
AI Safety & Quality, Adobe Firefly
Adobe

Dr. Chao Zhang
School of Computational Science & Engineering
Georgia Institute of Technology

Date approved: May 18, 2026

It is our choices that show what we truly are, far more than our abilities.
Albus Dumbledore

To my parents and sister, for everything.

ACKNOWLEDGMENTS

My PhD has been a journey filled with genuine joy, excitement, curiosity, and fulfillment. I have been incredibly fortunate to be surrounded by wonderful people who made the joyful moments even more memorable and turned every experience along the way into an invaluable opportunity for growth and learning. This dissertation would not have been possible without their support and presence in my life.

First and foremost, I would like to express my deepest gratitude to my advisor, Polo Chau. His unwavering support throughout my PhD has meant more to me than words can fully express. He has been not only an exceptional academic advisor but also a role model, a mentor, and an academic father who has profoundly shaped both my research and my growth as a person. Through his constant guidance, encouragement, and trust in me, he created an environment where I was able to explore ideas freely, take on challenges with confidence, and steadily grow along the way. Even during challenging moments, I often found myself thinking that I may have used up most of my luck to have Polo as my advisor—and that thought gave me strength and reassurance to keep moving forward, something I hold with the deepest appreciation.

I also greatly appreciate brilliant people at the Polo Club of Data Science at Georgia Tech. I would like to thank my academic siblings—Haekyu Park, Austin Wright, Jay Wang, Ben Hoover, Anthony Peng, Alec Helbling, Matthew Hull, Mansi Phute, Aeree Cho, Alex Yang, and Alex Greenhalgh—for their insights, encouragement, and the many joyful moments (trips and food) we shared. I also extend my sincere appreciation to the master's and undergraduate students who worked closely with me, especially Grace Kim, Kevin Li, Aishwarya Chakravarthy, and Omar Shaikh.

I would also like to thank the mentors I met during my internships, who played an important role in shaping my perspective beyond school. I am grateful to Ali Payani for his encouraging and invaluable mentorship during my first internship at Cisco. I also thank Richard Chen and Hsiang Hsu at JPMorgan Chase&Co. for their guidance, which greatly fostered my academic and professional growth. At Adobe, I had the pleasure of working with Oliver Brdiczka, Tianyuan Cao, Anelise Newman, Gahye Park, and Eric Slyman, all of whom made my internship experience inspiring and enriching. These experiences not only broadened my horizons but also made me genuinely excited about my future in industry.

My time in Atlanta would not have been joyful without my wonderful friends. To my DP friends, Garyoung Lee, Jeong Soo Lee, and Jooyeon Lee, thank you for all the *princess moments* we shared, which brought so much joy and light into my life. To my IMT friends, Sarah A Young Kim, Jinyoung Kim, Seung Jae Lieu, Chaeyeon Han, Seung Eun Choi, Garyoung Lee, and Jeong Soo Lee, thank you for your support and the fun times we had together. I am also grateful to all the GTKSA members I met during my five years there, especially those who worked with me during 2025-2026 while I had the honor of serving as

president—Jeongin Kim, Seungyeon Lee, Ikhwan Shin, Seung Yoon Lee, Haeyeon Choi, Yeongjun Kim, Taesoo Kim, Gyejin Lee, Ziwon Yoon, Yonghyun Kim, Hyunju Kim, and Eunho Yeo.

To my dear friends outside Atlanta, thank you for being my constant anchors. Jeongwon Lee, my best friend since high school, your unwavering support carried me through some of the most challenging times. I truly could not have endured them without you. I appreciate my *helper* Ho-Young Jang, my *sparring mates* Dongha Shin and Youngjae Kim, and my *PhD journey companion* Hyelim Do, for their steady friendship, and I wish all of us strength and good luck in the journeys ahead.

I would also like to express my heartfelt gratitude to Frederick Chung, for allowing me to happily place a period at the end of this long journey. I could not have finished my PhD on a happier note without your support, encouragement, and presence.

Last, but most importantly, I thank my family—my parents and my sister—for everything. Their unconditional love and belief in me have been the foundation of all that I have achieved. This dissertation is dedicated to my mother, Jin Han, my father, Sang Cheol Lee, and my sister, Sunghee Lee. I love you always and forever.

TABLE OF CONTENTS

Acknowledgments v

List of Tables xiii

List of Figures xv

Summary xx

Chapter 1: Introduction 1

1.1 Thesis Overview 2

1.1.1 **Part I: Attribute Risks** 2

1.1.2 **Part II: Explain Failure** 4

1.1.3 **Part III: Guide Mitigation** 6

1.2 Thesis Statement 7

1.3 Research Contributions 8

1.4 Impact 9

Chapter 2: Background and Related Work 11

2.1 AI Safety 11

2.1.1 Unreasonable Predictions of Classical AI Models 11

2.1.2 Unsafe Generations of Modern LLMs 11

2.1.3 Guiding AI Safety 12
2.2 Interpretation Algorithms for AI 13
2.2.1 Interpretation for Classical Models 13
2.2.2 Interpretation for Modern LLMs 13
2.3 Visual Explanations for AI 13
2.3.1 Interactive Visualizations for AI Models 13
2.3.2 Visual Toolkits for AI Risks 14

I Attribute Risks 15

Chapter 3: Interpretation Meets Safety: A Survey on Interpretation Methods and Tools for Improving LLM Safety 17

3.1 Introduction 17
3.2 Interpretation Methods for LLM Safety 18
3.2.1 Attribute Safety to Training Process 18
3.2.2 Identify Safety-Critical Input Tokens 20
3.2.3 Interpret Safety via Inference-Time Model Internals 21
3.2.4 Self-explain with Reason Generation 23
3.3 Enhance Safety using Interpretation 24
3.3.1 Attend to Relevant Input Tokens (Sec. 3.2.2) 24
3.3.2 Modify Model Internals for Safety (Sec. 3.2.3) 24
3.3.3 Verify Safety before Outputs (Sec. 3.2.3, Sec. 3.2.4) 25
3.3.4 Output with Self-Reasoning (Sec. 3.2.4). 25
3.4 Tools Operationalizing Safety-Focused Interpretation 25
3.4.1 Training Data Attribution (TDA) Visualization (Sec. 3.2.1) 25

3.4.2 Input Token Visualizations (Sec. 3.2.2) . . . 25
3.4.3 Latent vector visualizations (Sec. 3.2.3) . . . 25
3.4.4 Neuron visualizations (Sec. 3.2.3) . . . 26
3.5 Research Directions and Conclusion . . . 26
3.5.1 Defense against Interpretation-based Attack . . . 26
3.5.2 Reliable Evaluation of Interpretation . . . 26
3.5.3 Training Attribution for Safety Enhancement . . . 27
3.5.4 User-centered Presentation of Safety Interpretations . . . 27
3.5.5 Refining Safety Dimensions . . . 27
3.6 Conclusion . . . 27

Chapter 4: LLM ATTRIBUTOR: Interactive Visual Attribution for LLM Generation . . . 28

4.1 Introduction . . . 28
4.2 System Design and Implementation . . . 29
4.2.1 Data Attribution Score . . . 30
4.2.2 System Design . . . 30

II Explain Failure 31

Chapter 5: DIFFUSION EXPLAINER: Visual Explanation for Text-to-image Stable Diffusion . . . 33

5.1 Introduction . . . 33
5.2 Design Goals . . . 35
5.3 System Design and Implementation . . . 35
5.3.1 Overview . . . 35

5.3.2 Architecture View 36
5.3.3 Refinement Comparison View 37
5.4 Human Evaluation 38
5.4.1 Procedure 38
5.4.2 Results and Design Lessons 40
5.5 Conclusion 42

Chapter 6: VISCUIT: Visual Auditor for Bias in CNN Image Classifier 43
6.1 Introduction 43
6.2 System Design and Implementation 45
6.2.1 Overview 45
6.2.2 Subgroup Panel 45
6.2.3 Neuron Activation Panel 46
6.3 Usage Scenario 49
6.3.1 Bias Characterization 49
6.3.2 Model Performance Verification 50
6.4 Conclusion 50

III Guide Mitigation 52

Chapter 7: Effective Guidance for Model Attention with Simple Yes-no Annotations 54
7.1 Introduction 54
7.2 Methods 55
7.2.1 Overview 55

7.2.2 CRAYON-Attention: Guide Saliency Maps 55
7.2.3 CRAYON-Pruning: Prune Irrelevant Neurons 57
7.2.4 Time Complexity 57
7.3 Evaluation with Human Annotations 58
7.3.1 Benchmark Datasets and Model Training 58
7.3.2 Annotating Visualized Interpretations 59
7.3.3 CRAYON Configurations 60
7.3.4 Compared Methods 61
7.3.5 Evaluation Metrics 63
7.3.6 Results: CRAYON is Effective and SOTA 64
7.3.7 Scalability: Runtime Scales Linearly with Data Points 65
7.3.8 Varying Number of Annotations 66
7.3.9 Ablation Study 67
7.3.10 Transferability of Attention Annotations 67
7.3.11 Hyperparameter Sensitivity 68
7.4 Conclusion 69

Chapter 8: Probing LLM Hallucination from Within: Perturbation-Driven Approach via Internal Knowledge 70

8.1 Introduction 70
8.2 Discovery: Entity Perturbation Impact for Hallucination Probing 72
8.3 SHINE: Hallucination Probing Method 74
8.3.1 Model Knowledge Test 75
8.3.2 Alignment Test 76

8.4 Experiments 77
8.4.1 Hallucination Probing 77
8.4.2 Hallucination Detection 79
8.4.3 Efficiency of SHINE 81
8.4.4 Effectiveness of Model Knowledge Test 83
8.4.5 Hyperparameter Sensitivity 84
8.5 Conclusion 84

IV Conclusions **85**

Chapter 9: Conclusions and Future Directions 86
9.1 Research Contributions 86
9.2 Impact 87
9.3 Future Directions 88
9.3.1 Unified Systems From Risk Diagnosis to Automatic Mitigation . . . 88
9.3.2 Adaptive Explanations for Diverse Users and Contexts 89
9.3.3 Cross-model Interfaces for Rapidly Evolving AI Ecosystems 89
9.3.4 Refining AI Safety Taxonomies for Emerging Risks 89
9.3.5 Evaluating the Effectiveness of Interpretability Methods 89
9.4 Conclusion 90

References 91

LIST OF TABLES

3.1 Overview of representative works at the intersections of safety-focused interpretation (Sec. 3.2), safety enhancements they inform (Sec. 3.3), and tools operationalizing them (Sec. 3.4). Each row is one work; each column corresponds to a technique or tool. Safety issues, techniques, and tools addressed by a work are indicated by a colored cell. 19

7.1 CRAYON achieves the **best**, state-of-the-art performance in correcting model attention, outperforming all existing methods across 3 benchmark datasets, even surpassing methods that require more complex annotations. . 62

7.2 Ablation study demonstrates that $\mathcal{L}_{rel}$ guides model attention significantly stronger than $\mathcal{L}_{irrel}$ which directs attention away from irrelevant image regions. 66

7.3 The attention annotations for the *Original* ResNet50 model effectively refine ConvNeXt trained on 3 datasets. 67

8.1 Confusion matrix showing SHINE's hallucination probing performance (%) on LLaMA2, LLaMA3, and Mistral across two datasets: NEC / biography. . 79

8.2 SHINE (in **bold**) outperforms all existing approaches in hallucination detection (%), especially for fabricated text, across the NEC / Biography datasets. For a fair comparison, we adapt SHINE's predictions to binary by combining its misaligned and fabricated classes into the conventional *hallucinated* class, and mapping its aligned class to the *faithful* class. 80

8.3 SHINE (**bolded**) outperforms all existing methods in hallucination detection (%) on HaluEval/TriviaQA. 82

8.4 We vary k_{eff} for the LLaMA2 model on the 322 NEC data points evaluated with the Alignment Test. We report the number of data points examined by SelfCheckGPT (# SelfCheckGPT) and the corresponding accuracy. Larger k_{eff} values offer higher efficiency while preserving state-of-the-art accuracy, whereas smaller k_{eff} values yield additional accuracy gains. 83

LIST OF FIGURES

1.1 This thesis develops methods that **attributes** AI model risks to their underlying causes, **explains** these risks to a variety of audiences, and **guides** their mitigation. 1

1.2 My thesis includes three complementary parts. Each part addresses one research question with research answers and example works mapped to six chapters of this thesis. 2

1.3 Framework of our survey bridging LLM interpretation and safety. It connects safety-focused interpretation methods, safety enhancements they inform, and the tools that operationalize them, organized by a taxonomy based on LLM workflow: training, input tokens, inference, and generation. 3

1.4 LLM Attributor visualizes training data attribution in computational notebooks. Users can interactively select (A) tokens for attribution, viewing (B) top-ranked training data points as (B1) a list with (B2) expandable details, (C) explore summarized keywords, and (D) analyze attribution scores through a histogram. 4

1.5 Diffusion Explainer integrates an overview of Stable Diffusion's components with explanations of their operations. Users can see how (A) a text prompt is encoded into (B) a representation that guides (C) iterative image representation refinement, with (D) each step reviewable, before (E) the representation is upscaled into higher resolution. 5

1.6 VisCUIT reveals *how* and *why* a CNN image classifier is biased. For a smile classifier that spuriously links hair color with smiling, it displays (A) underperforming subgroups with (A1, A2) attentions and (A3) performance summary, and (B) neuron activations with (B1, B2) associated concepts. . . 6

3.1 Visual overview our survey's unified framework bridging LLM interpretation and safety, summarizing the connections between safety-focused interpretation methods (Sec. 3.2), the safety enhancements they inform (Sec. 3.3), and the tools that operationalize them (Sec. 3.4). We organize surveyed research using a novel taxonomy based on LLM workflow: training, input tokens, inference, and generation. 18

4.1 LLM ATTRIBUTOR visualizes the training data attribution in computational notebooks. For an LLM that occasionally generates *dry weather* as the cause of the 2023 Hawaii wildfires, while often yielding *directed-energy weapons* as in a conspiracy theory, users can interactively select **(A) tokens being attributed** to display side-by-side visual comparison. **(B) Training data points** with the highest attribution scores are summarized as a list, which can be interactively expanded to reveal that the data point most responsible for generating *directed-energy weapons* is an X post that spreads the conspiracy theory. **(C) Keyword Summary** shows important words in the displayed training data. **(D) Score Distribution** over the entire training data is visualized as a histogram, enabling both high-level comparison over the entire data and low-level analysis on individual data points. Below, the training data points with the lowest attribution scores are visualized in the same way. 28

5.1 With DIFFUSION EXPLAINER, users can visually examine how **(A)** *text prompt*, e.g., *"a cute and adorable bunny... pixar character"*, is encoded by **(B)** the *Text Representation Generator* into vectors to guide **(C)** the *Image Representation Refiner* to iteratively refine the vector representation of the image being generated. **(D)** The *Timestep Controller* enables users to review the incremental improvements in image quality and adherence to the prompt over timesteps. **(E)** The final image representation is upscaled to a high-resolution image. Diffusion Explainer tightly integrates a visual overview of Stable Diffusion's complex components with detailed explanations of their underlying operations, enabling users to fluidly transition between multiple levels of abstraction through animations and interactive elements (Fig. 5.2, Fig. 5.3). 33

5.2 To learn how Stable Diffusion converts a text prompt into vector representations, users click the *Text Representation Generator*, which smoothly expands to **(A)** the *Text Operation View*, which explains how the prompt is split into tokens and encoded into vector representations. **(B)** The *Text-image Linkage Explanation* demonstrates how Stable Diffusion bridges text and image, enabling text representations to guide the image generation process. 37

5.3 Users learn how Stable Diffusion refines noise into a high-resolution image's vector representation aligned with the text prompt by clicking the *Image Representation Refiner* to smoothly expand to **(A)** the *Image Operation View* that demonstrates how noise is predicted and removed from the image representation. **(B)** The *Interactive Guidance Explanation* allows users to interactively experiment with different guidance scale values (0, 1, 7, 20) to better understand how higher values lead to stronger adherence. 38

5.4 The *Refinement Comparison View* enables users to discover the impacts of prompts on image generation by comparing how image representations evolve differently over refinement timesteps, using UMAP, when guided by two related text prompts. Adding *"pixar"* phrase changes the generated bunny's style to be more cartoony and vibrant in colors and textures while preserving its pose. 39

5.5 DIFFUSION EXPLAINER more usable than blog post. 40

5.6 All features were rated highly. 40

5.7 DIFFUSION EXPLAINER more effective than blog post for learning most Stable Diffusion concepts. 41

6.1 VISCUIT reveals *how* and *why* a CNN image classifier is biased. (A) *Subgroup Panel* displays underperforming data subgroups found by UDIS [15]. Clicking on a subgroup displays a similar subgroup with high accuracy. Clicking on an image in either subgroups brings up a Grad-CAM Window, which shows that the classifier attends to (A1) irrelevant areas in underperforming subgroup but (A2) relevant areas in the well-performing subgroup. (A3) Confusion matrices quantitatively summarize such misclassifications. (B) The *Neuron Activation Panel* reveals which neurons and concepts are responsible for misclassifications. Clicking on a neuron displays a Neuron Concept Window, which shows that (B1,B2) only the underperforming subgroup activates neurons for irrelevant concepts. 44

6.2 The **Grad-CAM Window** helps users understand the reasons for misclassifications; it is displayed when a user clicks on an image. (a) Grad-CAM Window for an image in the underperforming subgroup #22 reveals that the model attends to background areas not relevant to facial expressions. (b) Grad-CAM Window for an image in the well-performing subgroup #296 reveals that the model attends to the face as expected. 46

6.3 The **Neuron Concept Window** helps users identify image concepts responsible for activating neurons that significantly contribute to model prediction. (a) Neuron Concept Window for a neuron highly activated only by the underperforming subgroup #22 shows that *text* and *non-facial textures*, both irrelevant to predicting *smiling*, have the major contribution to the misclassification, with the high activation score of 1. (b) Neuron Concept Window for a neuron highly activated by the well-performing subgroup #296 shows the expected image features, such as *mouths* and *noses*. The Neuron Concept Window displays when the user clicks on a neuron in the *Neuron Activation Panel*. 48

6.4 Images in the underperforming subgroup #22. It is hard to define the characteristics of the subgroup #22 at the first glance. Using VISCUIT, our user Henry gets hints that there would be text and non-facial textures in the background and verifies it by brightening the image. Henry concludes that the classifier underperforms on the *images of smiling athletes in stadium*. . 49

7.1 **Left:** CRAYON uses purposefully simple yes-no annotations to guide a model to attend to relevant image areas (e.g., redirect from background to foreground), overcoming limitations of existing methods requiring complex annotations such as pixel-wise maps. **Right:** CRAYON achieves state-of-the-art performance, surpassing 12 other methods across 3 benchmark datasets. 55

7.2 CRAYON-PRUNING prunes the neurons activated by irrelevant concepts in the penultimate layer and fine-tunes the last layer. For example, in a smile classifier, **Left:** a neuron is activated by *smiling mouth*, which is relevant to smiling, while **Right:** another neuron is activated by irrelevant *blond hair* and is pruned. 57

7.3 Example visualization shown to participants for attention annotations. . . . 59

7.4 For the Waterbirds dataset, even with annotations for just 10% of the training data points, both CRAYON-ATTENTION and CRAYON-PRUNING+ATTENTION nearly reach their peak performance. CRAYON-PRUNING realizes its full effectiveness when annotations are provided for most neurons in the penultimate layer. The performance of CRAYON-PRUNING+ATTENTION with no attention annotations differs from CRAYON-PRUNING's peak performance, as the latter retrains the entire model, while the former fine-tunes only the last layer. For each annotation number n, we run each method five times with different random seeds and report the average MGA and WGA values. 65

7.5 For the Waterbirds dataset, both CRAYON-ATTENTION (left) and CRAYON-PRUNING+ATTENTION (right) effectively correct model attention for wide range of α and β. 68

8.1 We introduce a novel **hallucination probing** task that distinguishes hallucination into *misaligned* and *fabricated* text. SHINE, our novel hallucination probing method, achieves **best** performance in hallucination detection, outperforming 7 methods across 4 datasets and 4 LLMs (Table 8.2, 8.3). . . . 71

8.2 Entity Perturbation Impact, measured by KL divergence and ΔP_i, varies across aligned, misaligned, and fabricated text. **(A)** Tokens in aligned and misaligned text, strongly linked to the LLM's knowledge of the entity, have high KL divergence, while most tokens in fabricated text have near-zero values, indicating weak association with the LLM's knowledge. The mean KL divergence of fabricated text is significantly lower than aligned and misaligned text. **(B)** While tokens in aligned and fabricated text generally show negative or near-zero ΔP_i, misaligned text has tokens with positive ΔP_i, particularly those in the contradictory phrase, leading the mean ΔP of misaligned text to surpass that of aligned and fabricated text. 73

8.3 SHINE classifies LLM-generated text into aligned, misaligned, and fabricated without external knowledge or supervised training through two steps: Model Knowledge Test (Sec. 8.3.1) and Alignment Test (Sec. 8.3.2). The Model Knowledge Test identifies whether the LLM has sufficient knowledge to generate a faithful response by perturbing entities in the prompt and evaluating the impact; fabricated text is differentiated at this stage. For the data that the LLM has enough knowledge, we conduct the Alignment Test to examine whether the text aligns with the LLM's knowledge. 74

8.4 ROC curves of the Model Knowledge Score and existing methods for differentiating fabricated and non-fabricated (i.e., aligned and misaligned) text on the NEC dataset generated by the LLaMA2, LLaMA3, and Mistral models. The Model Knowledge Score achieves the highest ROC AUC values, demonstrating its superiority in identifying whether a model has sufficient knowledge about the prompt and text. 83

8.5 SHINE's mean accuracy on the NEC dataset for the LLaMA2, LLaMA3, and Mistral models while varying each of $\boldsymbol{\sigma}$, k_{att}, p^{MKS}, and p^{AS}. 84

SUMMARY

As modern AI systems, such as diffusion-based generative models or large language models (LLMs), continue to grow in scale, complexity, and societal impact, understanding and mitigating their risks has become increasingly urgent yet challenging due to their black-box nature.

My thesis addresses this critical challenge by developing novel visualizations and algorithms that help people understand the reasons and mechanisms behind AI behaviors, and take actionable steps to mitigate risks. Our work is organized into three complementary thrusts:

(1) **Attribute risks.** We begin with investigating how to uncover the underlying causes of AI risks. We present the first survey bridging LLM interpretation and safety. Building on insights from our survey that training data can offer intuitive explanations for LLM generations, we develop LLM Attributor, which visually reveals the training data sources behind LLM-generated text, offering a novel way to diagnose unsafe outputs.

(2) **Explain failure.** While interpretation algorithms reveal causes of AI risks, their impact depends on how effectively they are communicated. To fill this gap, we introduce interactive visualizations that explain complex model mechanisms to broad audiences. Diffusion Explainer helps non-experts understand modern generative AI, outperforming traditional tools in 56-participant user studies. Extending visualization to non-generative models, VisCUIT empowers experts to explore the mechanisms behind failures of classifiers.

(3) **Guide mitigation.** To reduce risks, we introduce CRAYON, simple yet powerful algorithms that help classifiers overcome reliance on irrelevant features using yes-no annotations; experiments with large-scale human evaluations with 5,893 participants show its superiority over 12 methods across three datasets — even those requiring complex annotations. Extending to modern LLMs, we develop SHINE algorithm to determine whether hallucinations stem from limited model knowledge or flawed generation strategies. SHINE effectively differentiates faithful text and two types of hallucinations across three LLMs, and outperforms seven hallucination detection methods across four datasets and four LLMs.

My PhD research develops practical, innovative, human-centered solutions for research problems grounded in real-world needs, from advancing AI education to improving LLM safety, leveraging close partnership with leading companies like Google, Adobe, Cisco, JPMorgan Chase, ADP, and Avast. My work has made significant impacts across academia, industry, and society: Diffusion Explainer and its followup work Transformer Explainer have reached over 638k users in 210+ countries and have been integrated into university AI courses (e.g., MIT, Columbia). My research has been recognized with honors including the Korean Honor Scholarship, NCWIT AiC Collegiate Award Finalist, and IEEE VIS Best Poster Award.

CHAPTER 1
INTRODUCTION

With the rapid advancement of artificial intelligence (AI), its integration into real-world applications is expanding. However, AI systems also introduce significant risks that can lead to serious societal consequences. For example, AI-driven legal [1, 2] and hiring [3] decisions have been shown to produce varying impacts to people in different sectors, while information generated by large language models (LLMs) has caused widespread issues in critical domains such as healthcare, politics, and finance [4, 5, 6, 7].

A major challenge in addressing these risks is the black-box nature of AI models, which makes it difficult to detect and mitigate such harmful behaviors. For instance, ensuring the accuracy of LLM-generated text requires exhaustive human review, which can be extremely tedious and even impossible. Similarly, to inspect whether an AI model underperforms for specific data groups, practitioners have to obtain additional labels for training or validation data and analyze their correlation with model predictions. Even after identifying the underperforming data groups, practitioners would struggle to mitigate the issues due to the lack of understanding of the reasons and mechanisms for the underperformance.

To fortify AI safety, there is an urgent need to deeply understand and correct AI models' undesirable behaviors. This requires not only uncovering the reasons behind unsafe AI outputs but also providing explanations that make these insights accessible to broad audiences.

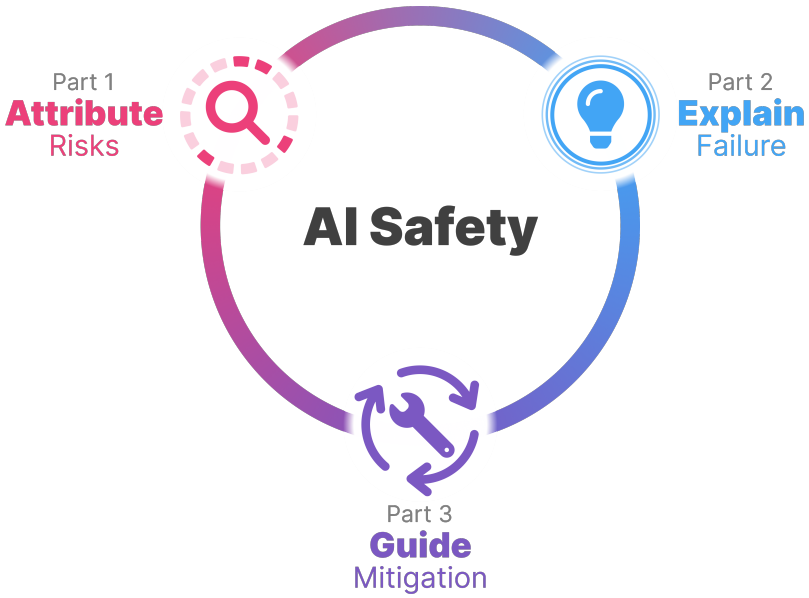


Figure 1.1: This thesis develops methods that **attributes** AI model risks to their underlying causes, **explains** these risks to a variety of audiences, and **guides** their mitigation.

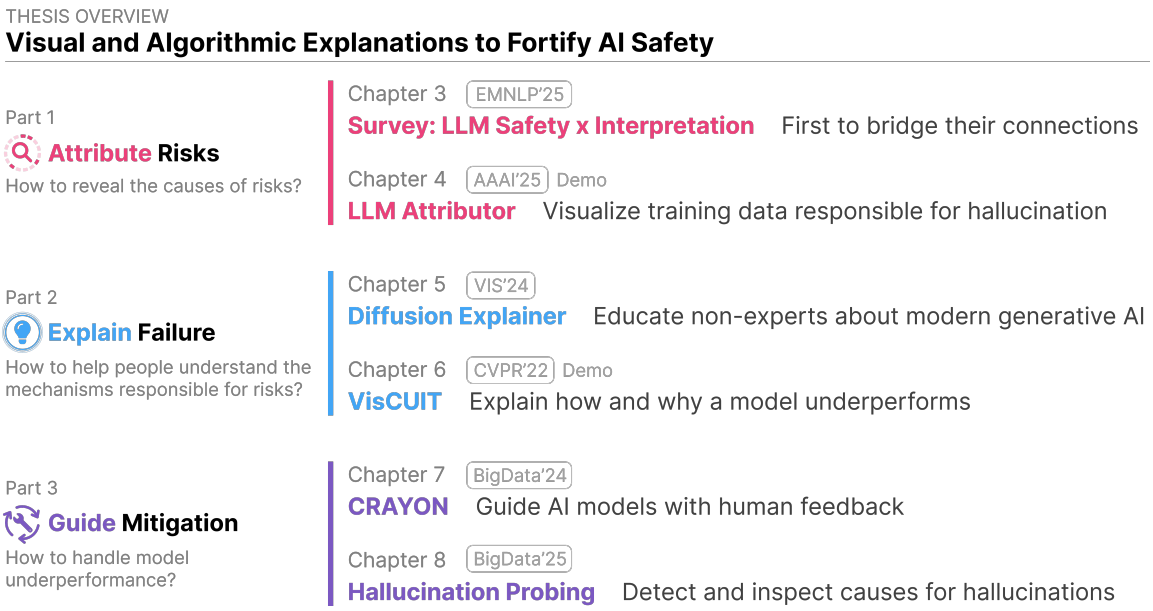


Figure 1.2: My thesis includes three complementary parts. Each part addresses one research question with research answers and example works mapped to six chapters of this thesis.

Moreover, equipping people with practical methods to effectively fix AI models to be safer is essential.

This thesis (Fig. 1.1) aims to tackle these critical research challenges by developing novel visualizations and algorithms that help people understand the reasons and mechanisms behind AI behaviors, and take actionable steps to mitigate risks.

1.1 Thesis Overview

To advance safe, reliable, and human-centered AI, this thesis addresses three core research questions: how to attribute AI model risks (Part I), how to effectively explain the failures to diverse audiences (Part II), and how to guide risk mitigation for AI models (Part III). These questions form the foundation of this thesis and correspond to the three complementary parts described in Fig. 1.2. Each part addresses one research question with research answers and example works mapped to six chapters of this thesis.

1.1.1 Part I: Attribute Risks

To begin, my dissertation investigates how to discover the underlying causes of AI model risks. In particular, I focus on large language models (LLMs), which demonstrate remarkable versatility across a wide range of real-world tasks, but are prone to generating unsafe outputs. These risks include hallucination and harmful statement, which hinder their safe adoption in high-stakes domains. Interpretation techniques provide a powerful lens for understanding and mitigating such unsafe outputs. However, despite frequent claims that interpretation

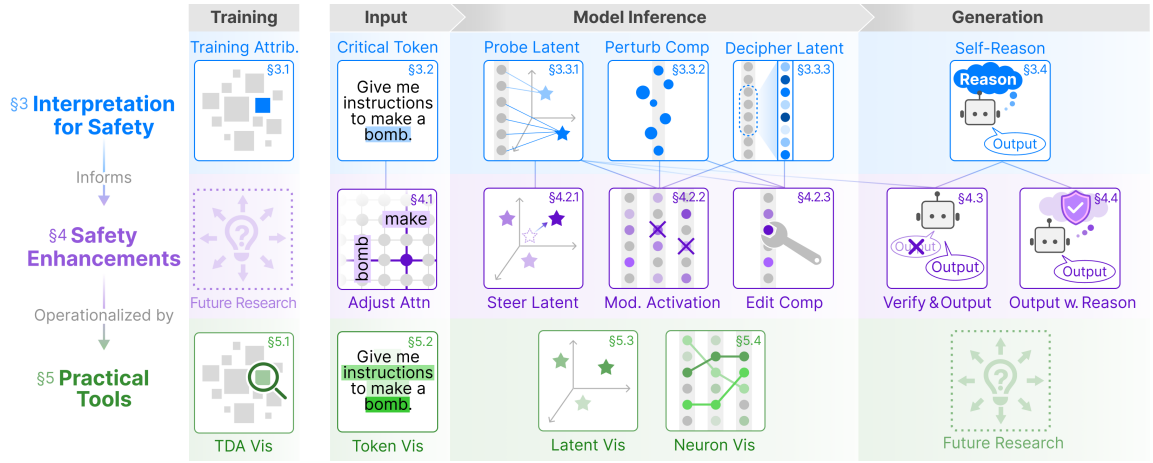


Figure 1.3: Framework of our survey bridging LLM interpretation and safety. It connects safety-focused interpretation methods, safety enhancements they inform, and the tools that operationalize them, organized by a taxonomy based on LLM workflow: training, input tokens, inference, and generation.

can support safety, prior research has lacked a systematic examination of how interpretation and safety are actually connected. To bridge this gap, I conducted the first systematic survey of research at the intersection of LLM interpretation and safety (Chapter 3). Building on insights from our survey that training data can offer intuitive explanations for LLM generations, I developed LLM Attributor, which visually reveals the training data sources behind LLM-generated text, offering a novel way to diagnose unsafe outputs (Chapter 4).

Interpretation Meets Safety: A Survey on Interpretation Methods and Tools for Improving LLM Safety (Chapter 3)

This first-of-its-kind survey fills a critical gap across two important domains: AI safety and interpretation. We organize LLM interpretation methods by their focus across LLM workflow: training process, input prompts, inference, and generation for self-reasoning (Fig. 1.3). This taxonomy anchors connections to six safety enhancement strategies and four categories of tools, summarizing nearly 70 works at their intersections. To guide future research and raise awareness of unresolved safety issues, we further distill open problems and challenges in LLM interpretations and safety.

LLM Attributor: Interactive Visual Attribution for LLM Generation (Chapter 4)

Our survey revealed that training data attribution, which traces unsafe outputs back to problematic sources, is a promising yet underexplored approach to LLM safety. Building on this observation, we develop LLM ATTRIBUTOR, a Python library that visualizes the training data points attributable to an LLM's text generation (Fig. 1.4). LLM ATTRIBUTOR

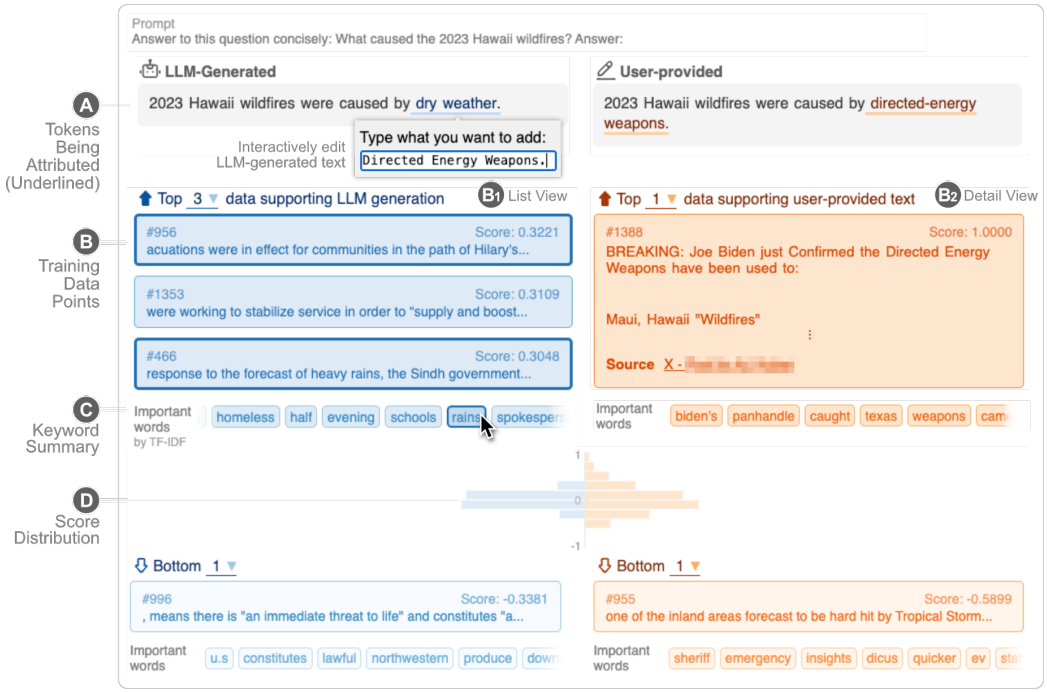


Figure 1.4: LLM Attributor visualizes training data attribution in computational notebooks. Users can interactively select (A) tokens for attribution, viewing (B) top-ranked training data points as (B1) a list with (B2) expandable details, (C) explore summarized keywords, and (D) analyze attribution scores through a histogram.

enables users to quickly trace generated text back to training data, offering insights into model behavior and enhancing trustworthiness. Furthermore, our tool supports comparisons between model-generated and user-provided text, enabling users to understand why certain model outputs have predominance. Designed for seamless use in computational notebooks, our library provides interactive visualizations that integrate easily into existing workflows, making attribution accessible to researchers and practitioners.

1.1.2 Part II: Explain Failure

While Part I examined how to uncover the causes of risky AI behavior, identifying these causes alone is insufficient. Interpretation algorithms may surface valuable insights, but their practical impact depends on how effectively those insights are communicated. Part II of the thesis fills this gap by introducing interactive visualizations that explain complex model mechanisms to a variety of audiences. We begin with Diffusion Explainer (Chapter 5), which visualizes how text-to-image diffusion models transform prompts into images. Yet, the need for clear explanations goes beyond generative models, particularly to CNN classifiers, which often make unreliable predictions based on spurious correlations. To address this, we develop VisCUIT (Chapter 6), an interactive tool that reveals how and why a CNN classifier underperforms on specific data groups, enabling researchers and practitioners to

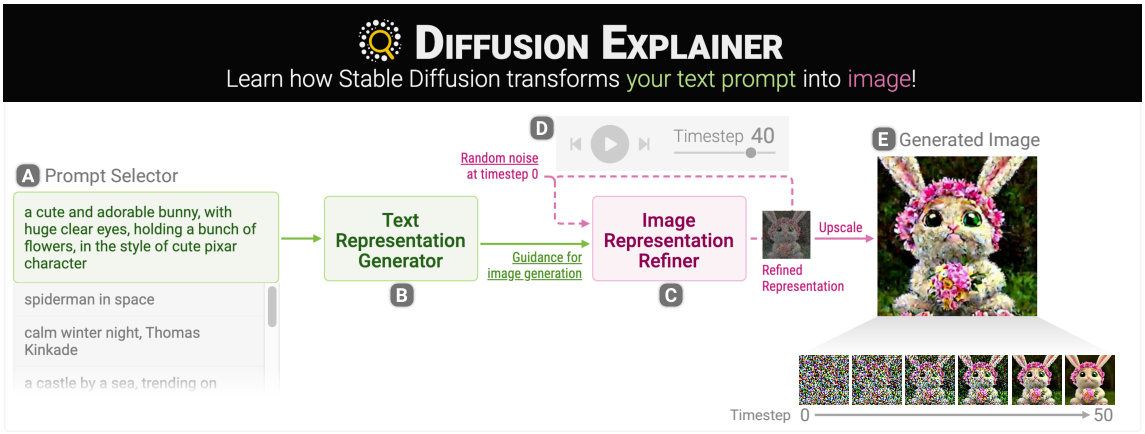


Figure 1.5: Diffusion Explainer integrates an overview of Stable Diffusion's components with explanations of their operations. Users can see how (A) a text prompt is encoded into (B) a representation that guides (C) iterative image representation refinement, with (D) each step reviewable, before (E) the representation is upscaled into higher resolution.

easily pinpoint the causes of model failures.

Diffusion Explainer: Visual Explanation for Text-to-image Stable Diffusion (Chapter 5)

Our first work of this part, Diffusion Explainer (Fig. 1.5), responds to the growing popularity of diffusion-based text-to-image generative models. While these models are widely used by diverse people, ranging from artists to marketing professionals, their complexity makes them difficult to understand, and their outputs raise critical safety concerns such as inappropriate content or intellectual property violations. To build awareness of these risks and help users better understand model behavior, Diffusion Explainer offers an interactive visual overview of Stable Diffusion's architecture with clear explanations of its underlying operations. Our tool also enables users to experiment with prompt variations, revealing the impact of keywords changes on image generation. A 56-participant user study demonstrates that DIFFUSION EXPLAINER offers substantial learning benefits to non-experts. Our tool has been used by over 28,000 users from over 140 countries at https://poloclub.github.io/diffusion-explainer/.

VisCUIT: Visual Auditor for Bias in CNN Image Classifier (Chapter 6)

We extend our approach of developing interactive visualization to classical CNN image classifiers, which often make unreliable predictions due to spurious correlations. Their complex and opaque internal operations make it difficult to pinpoint why predictions fail. To address this, we develop VISCUIT, an interactive visualization system that reveals how and why a CNN classifier is biased (Fig. 1.6). VISCUIT visually summarizes the subgroups where the classifier underperforms and helps users discover and characterize the

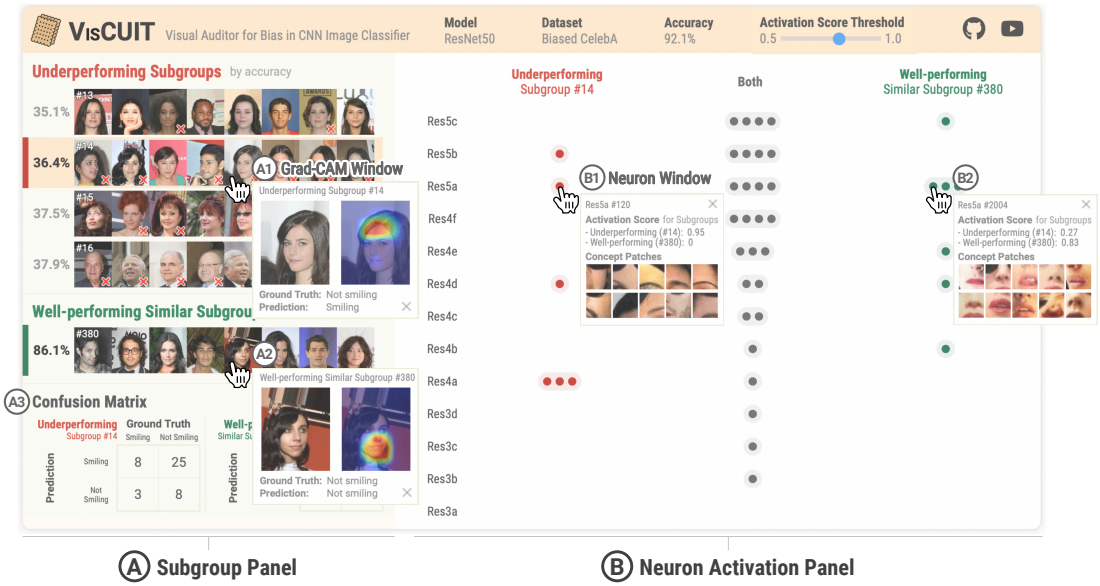


Figure 1.6: VisCUIT reveals *how* and *why* a CNN image classifier is biased. For a smile classifier that spuriously links hair color with smiling, it displays (A) underperforming subgroups with (A1, A2) attentions and (A3) performance summary, and (B) neuron activations with (B1, B2) associated concepts.

causes of these failures by linking misclassifications to the image concepts that activate specific neurons. VisCUIT runs directly in modern browsers and is open source, making it broadly accessible and easy to extend to other model architectures and datasets. VisCUIT is publicly available at https://poloclub.github.io/VisCUIT, with a demo video at https://youtu.be/eNDbSyM4R_4.

1.1.3 Part III: Guide Mitigation

Building on Part I and Part II, which uncovered and communicated the root causes of risky AI behaviors, Part III of my dissertation turns to the natural next step: mitigating these risks. This part starts with CRAYON (Chapter 7), an algorithm that resolves a common failure in AI classifiers — relying on spurious knowledge learned from training data and attending to irrelevant features when making predictions. Extending this idea to modern LLMs, I introduce the *hallucination probing task* (Chapter 8), which examines whether an LLM possesses sufficient knowledge to generate a faithful response and whether the response aligns with the knowledge, and develop the SHINE algorithm to address these questions.

Effective Guidance for Model Attention with Simple Yes-no Annotations (Chapter 7)

We first introduce CRAYON (Correcting Reasoning with Annotations of Yes Or No), which provides effective, scalable, and practical solutions to guide model attention using only simple yes-no annotations. AI models often rely on spurious correlations learned from training

data, resulting in unreliable predictions. To address this challenge, CRAYON integrates two complementary strategies: CRAYON-Attention, which directs classic saliency-map-based interpretations toward relevant image regions, and CRAYON-Pruning, which removes irrelevant neurons identified by modern concept-based methods. Together, these approaches align model attention with human expectations, mitigating bias without costly annotation requirements. Through extensive experiments with both quantitative and human evaluation, we showcase CRAYON's effectiveness, scalability, and practicality. CRAYON achieves state-of-the-art performance, outperforming 12 methods across 3 benchmark datasets, surpassing approaches that require more complex annotations.

Probing LLM Hallucination from Within: Perturbation-Driven Approach via Internal Knowledge (Chapter 8)

We extend this overarching principle — ensuring models rely on appropriate knowledge — to modern LLMs, which present a new challenge of hallucinations, where outputs may appear convincing but lack factual grounding. To better understand the causes of hallucinations, we introduce *hallucination probing*, a new task that classifies LLM-generated text into three categories: *aligned* (LLM has sufficient knowledge and responds faithfully), *misaligned* (LLM has knowledge but generates contradictions), and *fabricated* (LLM lacks relevant knowledge). Driven by our novel discovery that perturbing key entities in prompts affects LLM's generation of these three types of text differently, we develop SHINE (**S**ystematic **H**allucination **I**nspection with **N**oisy **E**ntity), a novel hallucination probing method that does not require external knowledge, supervised training, or LLM fine-tuning. SHINE is effective in hallucination probing across three modern LLMs, and achieves state-of-the-art performance in hallucination detection, outperforming seven competing methods across four datasets and four LLMs, underscoring the importance of probing for accurate detection.

1.2 Thesis Statement

Solutions to enhance AI safety by empowering diverse audiences—from novices to expert practitioners—identify risks, interpret model behaviors, and intervene in harmful outcomes through the development of methods that:

1. **Attribute** AI risks by unifying interpretation techniques and presenting their outcomes as clear, easy-to-understand visualizations,
2. **Explain** failures of both modern generative models and classical classifiers through engaging, interactive visualizations, and
3. **Guide** risk mitigation by designing practical algorithms that systematically align model decision processes with human reasoning.

1.3 Research Contributions

My thesis contributes to the fields of human-computer interaction, machine learning, visualization, and their intersections by developing approaches to **attribute** risks (Part I), **explain** model failures (Part II), and **guide** risk mitigation (Part III).

Bridging Interpretability and Safety: A New Paradigm for Responsible and Safe AI

- We present the **first comprehensive survey** that connects LLM interpretation and safety (Chapter 3). We introduce a unified framework that summarizes safety-focused interpretation methods, interpretation-informed safety enhancement strategies, and practical tools that operationalize them. This timely survey provides researchers and practitioners with a clear roadmap of key advancements, establishing interpretation as a cornerstone for building safer LLMs.

- LLM Attributor (Chapter 4) elevates training data attribution, a powerful yet underexplored interpretation technique, into a **practical safety tool**. By enabling practitioners to directly visualize and inspect the training data most responsible for an output, it delivers interactive, evidence-based reliability checks that can be leveraged to diagnose unsafe generations.

- VisCUIT (Chapter 6) transforms interpretation into **fast, intuitive diagnostics of spurious model behaviors**, exposing how and why models produce unreliable predictions. VisCUIT empowers practitioners to identify and preempt risks before deployment in high-stakes applications.

- CRAYON (Chapter 7) pioneers an algorithmic bridge between interpretation and intervention, offering a **practical and scalable solution to rectify improper model attention**. CRAYON enables both classic and modern interpretation methods to directly enhance model robustness and reliability.

Revolutionizing AI Understanding: Global Reach through Interactive Visualization

- Diffusion Explainer (Chapter 5) demonstrates the **substantial learning benefits of interactive visualization** for understanding generative AI compared to traditional resources, validated by a 56-participant user study. Together with the follow-up Transformer Explainer, these tools have reached over 638,000 users across 200+ countries and been integrated into top university courses (e.g., MIT, Columbia), establishing interactive visualization as a proven approach for democratizing AI education.

- LLM Attributor (Chapter 4) offers **visualizations for computational notebooks**, making attribution fast, legible, and seamlessly integrated into workflows. Features such

as influential example visualization and side-by-side comparison allow practitioners to interactively explore how outputs arise, simplifying adoption of attribution.

- VisCUIT (Chapter 6) advances **interactive exploration of model internals**, enabling researchers and practitioners to easily probe and interpret complex attention patterns. Its visualization design provides an accessible yet powerful interface for understanding model behaviors, complementing its role as a diagnostic tool for safety.

Breakthrough Algorithms Driving Safer Models

- CRAYON (Chapter 7) introduces the **first practical and scalable method to correct model attention** with simple yes-no feedback. Validated by a large-scale evaluation with nearly 6,000 participants, CRAYON demonstrates effectiveness, scalability, and practicality in refining model attention.

- SHINE (Chapter 8) establishes a **new paradigm for probing LLM hallucinations through model knowledge**. Without external knowledge, supervised training, or fine-tuning, SHINE predicts whether an LLM possesses sufficient knowledge to generate a faithful response, and whether the response aligns with the model's knowledge, providing a powerful diagnostic for hallucination detection and analysis.

Open-source and Extensible Systems Accelerating Global AI Innovation

- Across this thesis, I have released **many open-source, paradigm-shifting tools**— from web-based visual interfaces (Chapter 4, 5, 6) to scalable algorithms to improve AI safety (Chapter 7, 8)— that are freely available for adoption and extension. This work has empowered researchers, practitioners, and educators worldwide to build upon my designs, accelerating innovation and advancing the responsible use of AI.

1.4 Impact

My thesis research is already making significant impact on industry and society.

- Diffusion Explainer (Chapter 5) and its followup work Transformer Explainer have pioneered generative AI education and literacy, being used by more than **638,000 novices across 200+ countries** and integrated into top university courses (e.g., MIT, Columbia). These contributions have been recognized by prestigious honors, including being named an NCWIT AiC Collegiate Award Finalist and winning the IEEE VIS Best Poster Award.

- My visualization designs for safety and interpretability, exemplified in LLM Attributor (Chapter 4) and VisCUIT (Chapter 6), led to **Cisco's OpenRAI library for responsible AI**, and informed the creation of a visualization interface with AVAST to strengthen web security.

- My research was recognized with the **Korean Honor Scholarship**, a competitive fellowship awarded by the Korean government to only 36 Korean students studying in the US.
- My works have been recognized with multiple awards: VisCUIT (Chapter 6) received the **Best Project Award Honorable Mention** (2nd place) in CS7001, a course taken by over 60 first-year CS PhD students at Georgia Tech; SHINE (Chapter 8) received the NeurIPS SoLaR Workshop Travel Award; CRAYON (Chapter 7) received the IEEE BigData Student Travel Award. Beyond academia, these ideas were translated into industry impact, with my work winning the **Best Poster Award** in the Adobe Firefly intern poster session.
- My thesis ideas contributed to securing two highly competitive funded grant awards: the concept of democratizing AI (Chapter 5 and 6) formed the foundation of a funded **$1.2M NSF Medium Grant**, while educational innovation in the era of generative AI (Chapter 5) received a **Google Research Award**.

CHAPTER 2
BACKGROUND AND RELATED WORK

This chapter reviews related work, focusing on the areas to which my thesis contributes: (1) AI safety (Sec. 2.1), (2) AI interpretation algorithms (Sec. 2.2), and (3) AI explanation visualizations (Sec. 2.3).

2.1 AI Safety

2.1.1 Unreasonable Predictions of Classical AI Models

There have been many efforts to understand failures in the state-of-the-art AI systems. Many of the face recognition algorithms have been proven to present performance disparity across racial and gender groups [8, 9, 10, 11]. Lambrecht et al. [12] and Angwin et al. [13] revealed that the advertisement recommendation systems and legal decision making software also struggle with similar issues.

To deeply analyze these issues in AI image classifiers, Singh et al. [14] investigated the co-occurrence between objects and their contexts for each category and attempted to decorrelate them to reduce classifiers' dependency on the contexts. UDIS [15], which is developed to generate the image subgroups without human guides or additional attributes, clustered images based on the last-layer feature vectors from classifiers and extracted the subgroups with low accuracies. However, it is hard to pinpoint how a classifier underperforms using these methods since numerous subgroups are generated and the characteristics of each subgroup is often unclear. Furthermore, most existing approaches [9, 10, 16] argue that the skewness in training datasets is the major source of such performance disparity.

2.1.2 Unsafe Generations of Modern LLMs

There are four major LLM safety concerns commonly studied in research [17]: *hallucination*, *jailbreaks and harmfulness*, *bias*, and *privacy leakage*. These safety risks often share similar underlying causes, such as risky training data and objectives that favor memorization over generalization. Such design choices can embed unintended correlations, reinforce biases, and leave models vulnerable to adversarial prompts. As a result, hallucinations arise from confident fabrication; jailbreaks exploit alignment flaws; bias reflects systemic data skew; and privacy leakage stems from memorized content. Despite differing in expression, these risks all stem from failures to generalize appropriately.

Among these safety issues, hallucinations have been actively explored to improve LLMs' accuracy and trustworthiness. Some approaches have verified the reliability of

LLM-generated text by comparing it to external sources or human evaluation [18, 19, 20]. For example, FactScore [19] examines text using Wikipedia or human evaluation, which can be unavailable or costly. As an alternative, researchers have trained classifiers to differentiate faithful and hallucinated text [21, 22, 23], or fine-tuned LLMs to avoid potentially hallucinated responses [24, 25, 26]. However, as these methods require supervised training on large labeled datasets, prompting [27, 28, 29] or consistency checks [30, 31, 32, 33] have been proposed but may fail against confident fabrications due to LLM's auto-regressive nature [34, 35, 36, 37]. Uncertainty estimation methods [38, 39, 36, 40, 41, 42, 43, 44] often overlook hallucinations from randomness or token dependencies [45, 46, 47].

2.1.3 Guiding AI Safety

To resolve AI models' failures, a lot of efforts have been dedicated to rectifying model attention. Some researchers have attributed irrelevant attention to spurious correlations in training data [48] and alleviate such issues by reweighting or subsampling training data [49, 50, 51, 52, 53]. However, challenges arise when a training dataset lacks spurious-free data. In response, some researchers opt to create balanced training datasets by collecting or generating additional instances [54, 55, 56, 57, 58]. Yet, these approaches can be costly or impractical in real-world scenarios [59]. Various loss terms [60, 61, 62] and pruning approaches [63] have been introduced to counter the impact of spurious correlations. However, all these methods require annotations for the attributes that are spuriously correlated with the class labels, while identifying the correlated attributes is challenging in practice [64]. Several methods addressing such limitations have shown some efficacy [51, 65, 66, 67].

To achieve higher performance while overcoming all the aforementioned limitations, researchers have incorporated humans in refining vision models [68, 69, 70]. Various approaches have extended model interpretation techniques beyond mere identification, aiming to align model attributions with human intuition [71, 72]. Many researchers have focused mainly on saliency maps and collected human annotations for model attention. The RRR loss [73] was proposed to redirect MLP models away from regions annotated by humans as irrelevant, later extending its applicability to deeper models [74, 75]. CDEP [76] and SPIRE [77] aim to reduce the impact of irrelevant pixels by leveraging contextual decomposition and masking specific objects in images, respectively. Stammer et al. [78] refine models at both pixel and concept levels by disentangling concepts within an image. However, all these methods require humans to supply ground truth attention maps for each image, which can be prohibitively costly to obtain. To address this challenge, some progress has been made by using eye-gaze tracking apparatus [79] or introducing simpler alternatives such as scribble maps [80] and bounding boxes [81], which yet do not resolve the inherent limitations of human-provided attention maps [75].

2.2 Interpretation Algorithms for AI

2.2.1 Interpretation for Classical Models

A growing body of research has proposed techniques to help people interpret the behaviors of convolutional neural networks (CNNs). Earlier CNN interpretation approaches have made input-level explanations, which aim to reveal the features in inputs with major contribution to model behaviors [82, 83]. However, these approaches do not demonstrate which neurons in CNN models are responsible for the model behaviors. Recently, several methods propose neuron-level explanations [84, 85, 86]. In parallel, some research attempts to interpret adversarial attack in CNNs [87, 88, 89], hyperparameter tuning [90, 91, 92], and model selection [93].

2.2.2 Interpretation for Modern LLMs

Defining LLM interpretation as *recognizing and explaining LLM behaviors in human-understandable terms* [94, 95, 96, 97], researchers have developed many interpretation methods for modern generative LLMs [98, 99]. For example, training data attribution (TDA), which identifies the training data points most responsible for model behaviors, has been actively explored thanks to its wide-ranging applications, including model interpretations [100] and debugging [101, 102, 103]. While some researchers have estimated the impact of individual training data points on model performance [104, 105, 106] and training loss [102, 107], others have attempted to scale influence functions [108], a classical gradient-based method, to non-convex deep models [101]. Recent efforts have been dedicated to adapt these methods to large generative models, primarily focusing on improving their efficiency [109, 103, 110].

2.3 Visual Explanations for AI

2.3.1 Interactive Visualizations for AI Models

Early tools like ConvNetJS MNIST demo [111] and Tensorflow Playground [112] enabled users to experiment with simple models and datasets directly in their browsers. To explain more advanced techniques, researchers have developed interactive articles [113, 114, 115, 116, 117, 118], but they often assume prior machine learning knowledge. To address the needs of non-experts, interactive visualization tools such as CNN Explainer [119], GAN Lab [120], and Adversarial-Playground [121] were developed.

Recent efforts have been also made to develop visual attributions tailored for LLMs [122, 123, 124, 125]. Transformers-Interpret [122], InSeq [123], and LIT [124, 125] visually highlight important segments of the input prompt, while Ecco [126] visualizes neuron activations and token evolution across model layers to probe model internals. However,

these methods that attribute model behaviors solely relying on the input prompt are not sufficient to explain the text generations of LLMs, whose behaviors are intricately linked to the training data [127].

2.3.2 Visual Toolkits for AI Risks

Since AI risks can hugely affect various people's lives, the toolkits to help people understand specific risks have been actively developed. FairML [128] quantifies the relative significance of the inputs to a predictive model to evaluate the model's fairness. While Aequitas [129] enables users to easily measure fairness of algorithms using various metrics, AI Fairness 360 [130] integrates a number of state-of-the-art techniques for algorithmic biases, including bias assessment metrics, bias mitigation algorithms, and bias explanations. FairVis [131] allows users to generate and explore data subgroups based on their domain knowledge and suggests relevant subgroups. However, these approaches are applicable only for the datasets with abundant well-defined attributes (e.g. tabular data), and therefore cannot handle the models associated with image data unless additional attributes for the data are provided.

Part I

Attribute Risks

Overview As AI systems are increasingly adopted in high-stakes applications, ensuring their safety has become critical. Among them, large language models (LLMs) pose unique risks due to their widespread deployment and open-ended generation, making safety-focused interpretations particularly important. While many interpretation techniques have been proposed to explain LLM behavior, it remains unclear which methods are most effective for assessing safety and how their results should be presented and applied to improve model safety.

Part I of this thesis begins by introducing a unified framework that connects safety-focused interpretation methods with the safety enhancements they enable, as well as the tools that operationalize them (Chapter 3). This chapter builds on work published at EMNLP 2025 [132].

Chapter 3

Interpretation Meets Safety: A Survey on Interpretation Methods and Tools for Improving LLM Safety. Seongmin Lee, Aeree Cho, Grace C. Kim, ShengYun Peng, Mansi Phute, and Duen Horng Chau. *Conference of Empirical Methods in Natural Language Processing (EMNLP)*, 2025. PDF

Our survey suggests that training data attribution is a particularly valuable direction, as it links model outputs back to the training data. This approach offers practitioners intuitive insights into why a model produces certain outputs and highlights potentially unsafe training data. To make such analysis more accessible in practice, we develop **LLM Attributor (Chapter 4)**, an open-source Python library that provides interactive visualizations for training data attribution of LLM text generation [133]. This chapter is adapted from work published at AAAI 2025 Demo.

Chapter 4

LLM Attributor: Interactive Visual Attribution for LLM Generation. Seongmin Lee, Zijie J. Wang, Aishwarya Chakravarthy, Alec Helbling, ShengYun Peng, Mansi Phute, Duen Horng Chau, Minsuk Kahng. *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2025. PDF

CHAPTER 3
INTERPRETATION MEETS SAFETY: A SURVEY ON INTERPRETATION METHODS AND TOOLS FOR IMPROVING LLM SAFETY

3.1 Introduction

Large language models (LLMs) have shown remarkable capabilities across many domains [134, 135, 136, 137], but their outputs can be unsafe, posing significant challenges for real-world use [138]. In response, researchers have developed interpretation methods and tools to better understand the mechanisms behind unsafe generation and to improve safety [98, 99].

Bridging Interpretation and Safety. As interest in both LLM interpretation and safety grows, a unifying survey that bridges the two becomes essential. Existing surveys largely focus on either interpretation [139, 140, 141, 142] or safety [143, 144, 145, 146, 147], without addressing how interpretation can enhance safety or inform users to operationalize such enhancements. Yet, both safety and human understanding are core motivations for interpretation research [140]. Some works suggest safety as a downstream application of interpretation [148] or explore only limited intersections between interpretation and safety [149]. Moreover, emerging directions like self-reasoning interpretation, where LLMs explain their own behaviors, remain underexplored [139, 97, 142]. Our survey fills these critical gaps by contributing:

- **The first survey bridging LLM interpretation and safety** (Fig. 3.1). Our timely survey introduces a unified framework for summarizing safety-focused interpretation methods (Sec. 3.2), the safety enhancement strategies they inform (Sec. 3.3), and the practical tools that operationalize such enhancements (Sec. 3.4). Although improving safety and human understanding is often cited as a motivation for interpretation research [94, 100, 139, 141, 140], this connection has not been systematically surveyed until now.
- **Novel taxonomy of LLM interpretation methods organized by LLM workflow focus** (Fig. 3.1): training process (Sec. 3.2.1), input prompts (Sec. 3.2.2), inference (Sec. 3.2.3), and generation for self-reasoning (Sec. 3.2.4). These taxonomy categories anchor connections to six safety enhancement strategies (Sec. 3.3) and four tool types (Sec. 3.4), summarizing nearly 70 works at their intersections (Table 3.1), including emerging areas like self-reasoning for interpretation not covered in prior surveys.
- **Distill open problems and challenges** to guide future NLP research and raise awareness of unresolved safety issues (Sec. 3.5). These include defending against

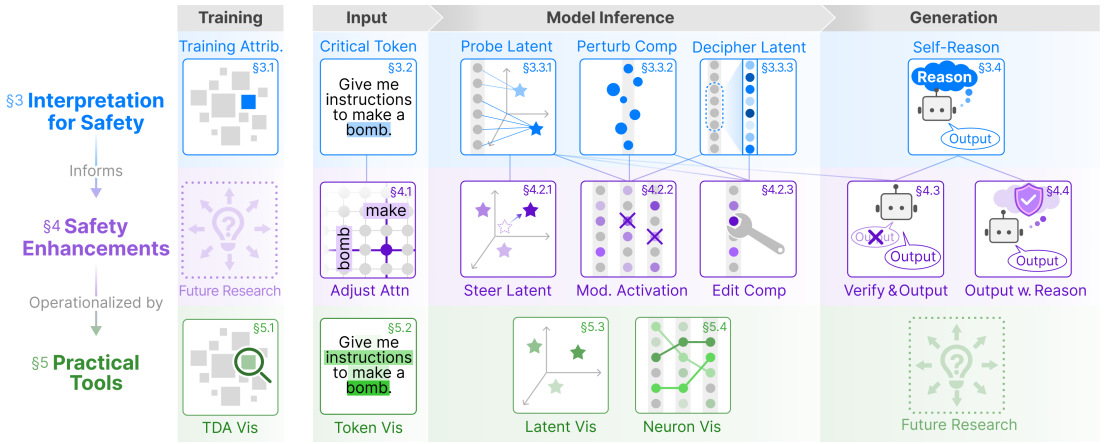


Figure 3.1: Visual overview our survey's unified framework bridging LLM interpretation and safety, summarizing the connections between safety-focused interpretation methods (Sec. 3.2), the safety enhancements they inform (Sec. 3.3), and the tools that operationalize them (Sec. 3.4). We organize surveyed research using a novel taxonomy based on LLM workflow: training, input tokens, inference, and generation.

interpretation-informed attacks, evaluating interpretation, using training attribution for safety, designing user-centered presentation of interpretation, and refining safety dimensions.

3.2 Interpretation Methods for LLM Safety

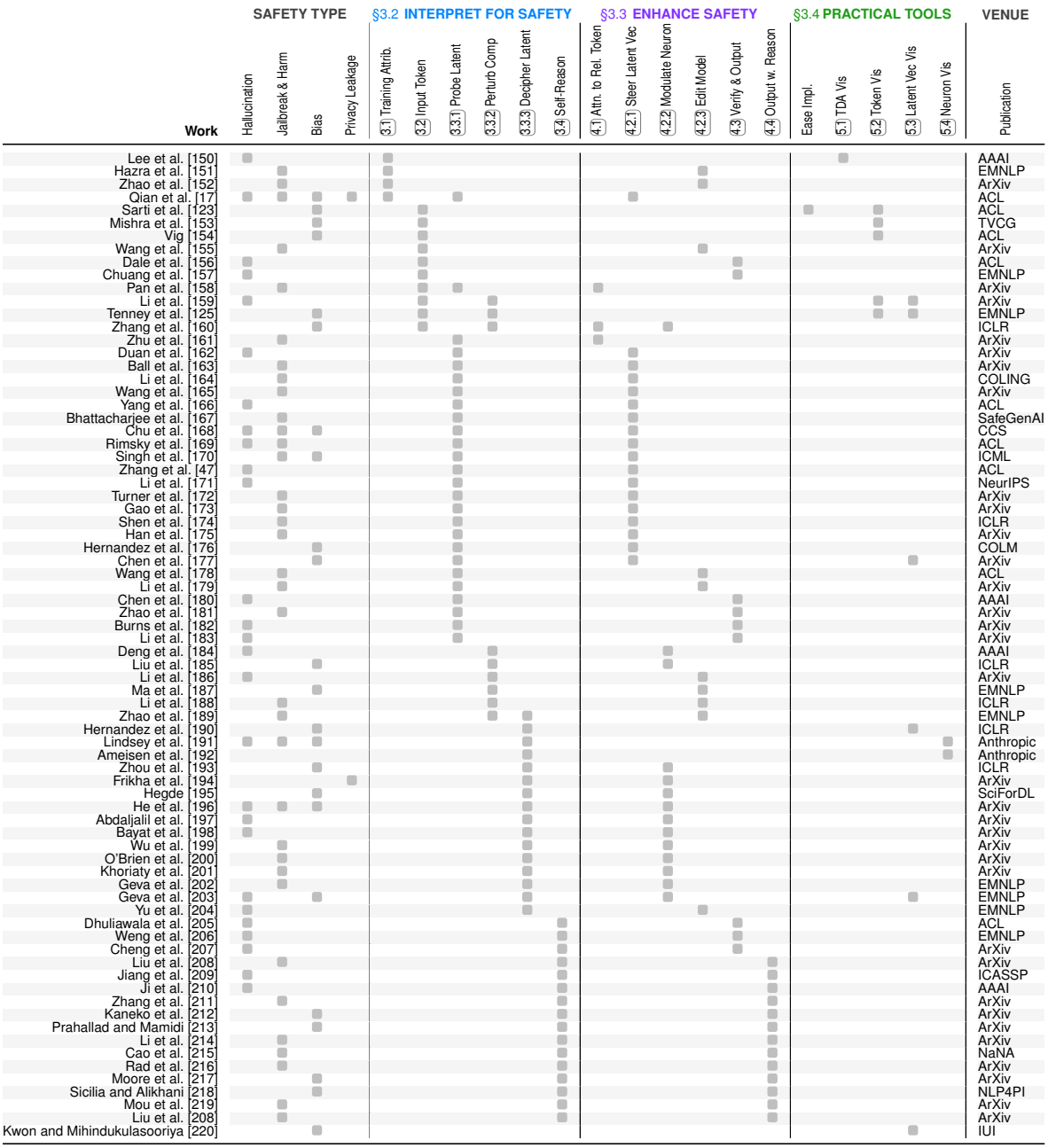
We categorize interpretation methods by where they operate in the LLM workflow (Fig. 3.1): training process (Sec. 3.2.1), input tokens (Sec. 3.2.2), model internals (Sec. 3.2.3), and underlying knowledge revealed by LLMs' self-explanatory capabilities (Sec. 3.2.4).

3.2.1 Attribute Safety to Training Process

Since LLMs are shaped by their training data [103], training data attribution (TDA) assumes unsafe behaviors stem from problematic training data and evaluates the contribution of each training data point to model behavior. **Representation-based attribution** does this by comparing the similarity between the latent vectors of each training example and the output [221, 222, 223, 224]. While effective for identifying related data, it does not establish causality [225].

To assess causal influence, **gradient-based methods** estimate how sensitive a model's parameters are to individual training examples. Many build on TracIn [102], which traces influence by measuring the alignment between gradients of losses computed for a model output and each training data. Variants have improved its accuracy [226, 227, 228, 106, 229] and adapted it for LLMs [230, 231]. However, these methods fall short in estimating the effect of removing a training point [232, 225]. More theoretically grounded work builds

Table 3.1: Overview of representative works at the intersections of safety-focused interpretation (Sec. 3.2), safety enhancements they inform (Sec. 3.3), and tools operationalizing them (Sec. 3.4). Each row is one work; each column corresponds to a technique or tool. Safety issues, techniques, and tools addressed by a work are indicated by a colored cell.



on **influence function** [233, 108, 101], which estimates how downweighting a training example affects model parameters and predictions [101]. Despite scalability improvements [234, 235, 236, 237, 238, 109] and extension to LLMs [103, 239, 240, 241, 242], their effectiveness is debated due to strong assumptions like model convexity [243, 244, 245],

which rarely hold for LLMs [225].

Another direction explores **Data Shapley**, which estimates the contribution of individual or groups of data points by approximating the effect of their removal or addition [104, 246, 247]. While promising, these methods are computationally expensive and have so far been limited to smaller models [248, 249]. Furthermore, the inaccessibility of LLMs' proprietary training data poses challenges for application of TDA overall [250, 251].

Beyond training data, researchers attempted to **understand LLM training dynamics** of learning new concepts and capabilities. Some studies use synthetic tasks with well-defined concepts, examining how models acquire them over training [252, 253]. Others compare models pre- and post-fine-tuning [152, 254, 151], simulate training [105, 107, 255], or analyze model internals (Sec. 3.2.3) across training checkpoints [256, 257, 258, 253, 259, 260]. These studies have revealed how safety capabilities like toxic content refusal emerge during training [17, 261].

3.2.2 Identify Safety-Critical Input Tokens

A major branch of AI interpretation research attributes model outputs to specific input features, hypothesizing that unsafe outputs stem from over-attending or misinterpreting specific input tokens [262, 263, 264, 265, 266, 267, 268]. For Transformer-based models, early methods examined **attention weights**, based on the intuition that higher weights singal greater importance [269, 270, 271]. While attention weights offer insights into model behavior [272, 273], monitoring all heads in LLMs can be overwhelming. Aggregation strategies, from simple heuristics (e.g., mean, max) [274, 275] to more principled attention rollout [270], help reduce complexity.

To improve the faithfulness by accounting for other model components like residuals and layer norms [276], **vector-based methods** decompose latent vectors into vectors attributable to input tokens [276, 277, 278, 279, 280, 281, 282, 283, 284]. These are applied to modern LLMs [285] and used to analyze hallucinations [286, 156, 157]. However, these methods require model-specific designs, limiting adaptability [287].

Perturbation-based methods offer a model-agnostic approach by modifying input tokens and observing output changes [264]. Perturbations include altering latent vectors [288, 289], masking or zeroing token embeddings [290, 291, 292, 293], replacing tokens [294, 295, 292, 296], or prompting counterfactuals [297, 298, 299]. Some extend Shapley value, which has been largely explored for classical models [268, 300], to estimate the influence of specific input tokens [301, 302, 303]. These methods have identified tokens triggering prompt poisoning [293] and biases [302]. However, they can be costly and create unnatural, out-of-distribution inputs, causing unfaithful interpretations [287, 282].

To address these pitfalls, **gradient-based methods** compute the gradient of the model output with respect to input embeddings, quantifying how changes to each token affect

output [262, 263, 265, 266, 267]. Initially designed for smaller models [304, 282, 305, 283], they have since been refined with contrastive explanations [290, 291, 306, 275] and extended to LLMs [307, 308, 309, 158].

Other approaches include **similarity-based** methods that compare the final output representation to input token embeddings, assuming higher similarity indicates greater token importance [278, 287]. In parallel, researchers have proposed **prompt-based approaches** [310, 311], which instruct LLMs to identify influential tokens for behaviors like jailbreaks [155], and **optimization-based techniques**, which search for token attributions that maximize certain interpretability metrics [312, 313, 314].

3.2.3 Interpret Safety via Inference-Time Model Internals

Multiple works have shown that unsafe behaviors are associated with certain regions in latent space [315] or specific model components [316] or neurons [317], which can be interpreted in three ways: (1) identifying latent regions tied to safety (Sec. 3.2.3); (2) perturbing components to assess impact (Sec. 3.2.3); and (3) deciphering latent vectors with human-understandable terms (Sec. 3.2.3).

Probe Safety Regions in Latent Space

Recent work investigates whether and how safety-related concepts are encoded in LLMs' latent vectors [315]. This work builds on the *linear representation hypothesis*, which posits that high-level concepts like factuality or harmfulness are embedded as linear directions in the model's latent space [318, 319, 320]. Under this view, researchers analyze latent vectors from individual layers—or concatenated across multiple layers—to identify which latent vectors encode safety behaviors.

A simple yet powerful approaches compute **mean latent vectors** for the data points with and without a particular concept (e.g., hallucinated vs. factual). These reveal directions associated with hallucinations [321, 180] and jailbreaks [322, 181, 161, 323]. Dimensionality reduction using PCA or SVD further uncover axes responsible for unsafe behaviors [162, 163, 158].

Another widely used technique is **probing classifiers**, where a model is trained to predict whether a latent vector encodes a safety-related property [324, 325, 326, 327, 328, 321, 329]. Probing successfully detects hallucinations [182, 330, 331, 332], jailbreaks [333, 334, 335, 332], and bias [331]. However, these properties are not always linearly separable [336], introducing non-linear classifiers [22, 337, 338, 23, 339, 340, 341, 342] or contrastive learning [343, 344] to better capture complex boundaries of unsafe model behaviors. A known challenge of the probing methods is poor generalization across tasks and datasets [345, 346, 347], which has been partially resolved by incorporating distributional differences into the loss function [339] or training probing models on diverse datasets [321].

Perturb to Assess Safety Impact

A common way to understand how specific components affect model behavior is to perturb them and observe changes. One approach uses **gradient-based analysis**, computing output gradients with respect to model parameters to evaluate each parameter's influence. While useful for explaining mechanisms behind knowledge conflicts in RAG [348] and biased generations [185], such methods may not sufficiently capture causal relations [349].

A more direct approach is **component knockout**, which ablates layers, attention heads, or parameters to identify their influence [350, 316]. This has localized components responsible for hallucinations [348, 186], jailbreaks [189, 351], and biases [352, 187]. Instead of full ablation, **parameter scaling** adjusts component influence [353], pinpointing safety-critical layers [188] and heads [354], while **parameter perturbation** modifies model weights and evaluates how safety properties respond to the structural changes [355, 356, 357], offering a broader perspective on the stability and robustness of safety alignment across the model's parameter landscape.

Activation patching, inspired by causal mediation analysis [358, 359], replaces intermediate activations (e.g., latent vectors, attention weights) from one input with those from another input, measuring how such intervention affects the model output. It localizes model components associated with hallucinations [360, 184] and biases [359], as well as general model capabilities [361, 362, 256, 363] and factual knowledge [364, 365, 366].

To uncover how components interact, researchers extract *computational circuits*, graphs with important components as nodes and information flow as edges [361, 367]. **Path patching**, an extension of activation patching, modifies outputs along specific computational paths while freezing the rest of the network[1] [368, 369, 253, 370]. Due to the high reliance on human inspection, several efforts automate circuit discovery [371, 372, 373], while attribution patching approximates causal effect for scalability [374, 375, 376, 377]. However, as LLM circuit analysis is still in its early stages, most focus on simple grammatical or arithmetic tasks, with very few addressing real-world safety problems [377].

Decipher Latent Vectors with Language

One approach to understand latent vectors through language is to analyze how their individual neurons respond to input data. By identifying inputs that highly activate a neuron and their shared patterns, researchers have inferred concepts encoded by each neuron [317, 378]. However, many neurons in LLMs are *polysemantic*, encoding multiple unrelated concepts, making interpretation challenging [379, 380, 381].

To address this, researchers have developed techniques to disentangle concepts superposed in the latent vectors [319]. A prominent method is using Sparse dictionary

[1] Path patching differs from activation patching in that it selectively modifies only the information flowing along specific paths, whereas activation patching replaces entire activations at specific components.

learning [382, 383, 367] to train **sparse autoencoders (SAEs)** [384, 380, 385, 386]. An SAE consist of an encoder and a decoder; the encoder maps a latent vector into a sparse, high-dimensional concept vector, where each dimension — SAE neuron — represents a distinct, interpretable concept, characterized by the inputs that strongly activate it [387]; the decoder reconstructs the original latent vector from the concept vector.

Since training separate SAEs for each (sub)layer in LLMs can be computationally intensive and redundant, later research enhances scalability and expressiveness through new architectures [388, 381, 389, 390], activation functions [391], and training strategies [392, 393, 394, 395, 396]. These advances have discovered more diverse concepts [397, 381, 398], offering insights into LLMs' hallucinations [399, 400], jailbreaks [401, 402, 403], biases [195, 193], and privacy leakage [194].

Beyond individual neurons, **SAE circuits** are extracted to reveal how interpretable concepts interact to produce specific outputs [404, 405, 406, 407]. SAE variants, such as Crosscoder [408] and Transcoder [389, 409, 192], enhance circuit interpretability and reduce redundancy, making it easier to isolate mechanisms behind unsafe behaviors [191]. In parallel, **logit lens** projects intermediate latent vectors onto the model's vocabulary space using the final projection matrix, viewing latent vectors on the vocabulary level [410, 367, 202, 411]. Further research enhances its robustness [412, 413] and extends it to analyze training dynamics [414]. The logit lens has been leveraged to investigate LLMs' knowledge store and recall mechanisms [415, 416] and safety issues, such as hallucinations [204, 417, 272, 348] and jailbreaks and harmfulness [189, 418, 261].

3.2.4 Self-explain with Reason Generation

The rapid advances in LLMs' self-explanatory and reasoning capabilities has prompted a surge of recent work aiming to explore how LLMs can interpret their own outputs by expressing the reasoning behind them in natural language [419, 139, 420]. A common approach is **in-generation reasoning**, where LLMs are prompted or trained to generate responses along with rationales [421, 422, 423] or uncertainty estimates [327, 424, 425, 426]. Chain-of-thought (CoT) prompting is a notable example, where LLMs generate intermediate reasoning steps to reach ananswer [427, 428, 429, 430]. Many CoT variants support more complex reasoning [431, 432] and improve explanation faithfulness [433, 434, 435, 436, 437]. However, such explanations can be unreliable [438, 439, 440], necessitating further verification [439, 441, 206, 442].

Alternatively, **post-hoc explanation** methods assess and explain a response after generation [443, 444]. These methods prompt LLMs to evaluate the correctness or safety of their outputs and provide rationales [445, 208, 446]. To explain hallucinations, a response may be split into factual claims or questions, which the model is then asked to verify against its knowledge [205, 447, 448].

3.3 Enhance Safety using Interpretation

Recent advances in LLM interpretation (Sec. 3.2) have inspired techniques to enhance model safety. This section reviews methods that leverage interpretation to mitigate unsafe behaviors, following the stages of the LLM workflow discussed in Sec. 3.2.

3.3.1 Attend to Relevant Input Tokens (Sec. 3.2.2)

Some methods prompt LLMs to attend to relevant input tokens to reduce hallucinations [449] and improve factuality [450]. Others remove jailbreak-triggering tokens [158] or manipulate attention to user-specified relevant tokens [160].

3.3.2 Modify Model Internals for Safety (Sec. 3.2.3)

Steer Latent Vectors To Safe Directions

Representation engineering guides LLMs' latent vectors toward safe regions by adding safety vectors identified by probing [315] or training transformations that map unsafe vectors into safe regions [176] (Sec. 3.2.3). These methods mitigate a range of safety concerns [17, 169, 170, 168], such as hallucinations [171, 166, 47, 162], jailbreaks and harmfulness [172, 167, 161, 163, 173, 174, 164, 175], and bias [176].

Modulate (Un)Safe Neurons' Activations

Suppressing risky neurons or amplifying safer ones guides LLMs away from unsafe behaviors. SAEs help locate and control (un)safe SAE neuron activations [451] (Sec. 3.2.3), addressing risks [196] like hallucinations [197, 198], jailbreaks and harmfulness [200, 452, 201, 199], biases [185, 195, 406, 193], and privacy leaks [194]. Alternatives remove dependency on SAEs by using logit lens (Sec. 3.2.3) to find and upscale safe MLP sublayers [202, 165] or amplify safety-critical attention weights (Sec. 3.2.3, Sec. 3.2.3) on user-specified reliable tokens [160, 184].

Edit Harmful Model Components

Safety can be improved by pruning or downscaling components (e.g., attention heads or (sub)layers) linked to hallucinations [186, 204], jailbreaks [189, 178, 179, 188, 155], and biases [187]. Other techniques identify safety directions in parameter space by comparing aligned and unaligned model weights, then steer critical parameters accordingly [151, 165, 152].

3.3.3 Verify Safety before Outputs (Sec. 3.2.3, Sec. 3.2.4)

Some approaches generate multiple candidate responses, evaluate their safety (Sec. 3.2.3, Sec. 3.2.4), and select only safe ones to construct final output [182, 453, 206, 156, 442, 157, 205, 180]. Others intervene during generation, resampling tokens when an unsafe sequence is detected [183, 207] or producing refusal messages [181, 219].

3.3.4 Output with Self-Reasoning (Sec. 3.2.4).

Building on CoT reasoning's success for performance and interpretation [210], several approaches fine-tune or prompt LLMs to generate intermediate reasoning steps to reinforce safety constraints during generation [212, 213, 214, 215, 218, 217, 216, 211, 219]. For instance, GuardReasoner [208] prompts models to explain why a response may be harmful, enabling safer behavior through self-reflection.

3.4 Tools Operationalizing Safety-Focused Interpretation

To apply interpretation methods (Sec. 3.2) and safety enhancement strategies (Sec. 3.3), practitioners need tools that support actionability [454, 455, 456]. Some efforts focus on developing libraries to **ease implementation** of interpretation and safety techniques [240, 457, 123, 458], while others introduce interactive visual tools, inspired by their effectiveness in enhancing human understanding of classical AI models [459, 460, 461, 462, 463].

3.4.1 Training Data Attribution (TDA) Visualization (Sec. 3.2.1)

TDA visualization shows how training examples shape model behavior. A prominent tool is LLM Attributor [150], which traces outputs to training data, identifying hallucination sources.

3.4.2 Input Token Visualizations (Sec. 3.2.2)

Token visualizations reveal individual tokens' importance and attention patterns, showing how tokens influence one another across heads. These are incorporated into LLM analysis tools [464, 125, 465, 159, 466, 467], and reveal spurious token associations indicative of bias [154]. Many tools also support interactive perturbation, allowing users to edit tokens or attention weights and observe the effects [468, 125, 466, 153].

3.4.3 Latent vector visualizations (Sec. 3.2.3)

Several methods have been developed to visualize how concepts are encoded and propagated during model inference. Some tools project latent vectors into 2D space [125, 159, 220].

Others visualize semantics of latent vectors revealed by logit lens [469, 470, 190] (Sec. 3.2.3), while some enable users to steer latent vectors for safer outputs [203, 177] (Sec. 3.3).

3.4.4 Neuron visualizations (Sec. 3.2.3)

Neuron visualizations display data points that highly activate each neuron during inference, revealing interpretable concepts [471, 472, 473]. (Sec. 3.2.3). Similar approaches are applied to SAE neurons [474], helping concept identification and SAE circuit discovery [475] for multiple unsafe behaviors [191, 192].

3.5 Research Directions and Conclusion

Our survey identifies five key open problems at the intersection of interpretation and safety research. As interpretation tools enter high-stake use, a pressing concern is their potential misuse, particularly when adversaries exploit interpretation methods to attack LLMs (Sec. 3.5.1). Mitigating this risk requires interpretations that are both faithful to model behavior and understandable to users (Sec. 3.5.2). With such reliability, interpretations can be leveraged proactively during training to promote safer model behaviors (Sec. 3.5.3). Their impact also depends on how effectively they are presented, underscoring the need for more user-aligned communication strategies (Sec. 3.5.4). Finally, we advocate for expanding interpretation research to address a broader range of safety risks beyond those currently emphasized (Sec. 3.5.5).

3.5.1 Defense against Interpretation-based Attack

Interpretation-driven attacks pose a unique threat, distinct from traditional adversarial attacks, as they exploit insights gained from interpretation methods [323, 476, 322, 477, 478]. For example, interpretation techniques can reveal why certain input tokens trigger risky outputs, enabling malicious users to craft diverse prompt suffixes that elicit unsafe behavior. Similarly, adversaries can use interpretation to identify and manipulate model components associated with harmful outputs. Because these threats arise unexpectedly from tools intended to enhance transparency and safety, defending against them is particularly challenging. This underscores the need to design interpretation methods with potential misuse in mind, along with implementing stronger defenses—such as proactively removing risky content [199].

3.5.2 Reliable Evaluation of Interpretation

The field currently lacks a clear consensus on the definitions of faithfulness and interpretability, as well as standardized methods for assessing interpretability [479]. Interdisciplinary collaboration—spanning machine learning, human-computer interaction, psychology, and

cognitive science—is essential for developing both formal definitions and practical evaluation frameworks. Moreover, even in human-centered evaluations, interpretability scores are often influenced by how interpretation results are presented to users [462]. This highlights the need for standardized evaluation protocols that consider both the presentation format and user characteristics, such as domain expertise and task relevance [480].

3.5.3 Training Attribution for Safety Enhancement

TDA shows promise for tracing unsafe behavior to training examples (Sec. 3.2.1), but its use in safety enhancement is limited. Prior work on retraining after removing problematic data [481, 482] focuses on non-safety issues on small non-generative models and cannot be scaled to LLMs. Developing safety enhancement methods based on training attribution could open new paths for risk mitigation.

3.5.4 User-centered Presentation of Safety Interpretations

How to present interpretation results to assist safety-critical decisions remains underexplored (Sec. 3.4). In particular, presentation of long, complex textual explanations from LLMs should be further investigated (Sec. 3.2.4); conversational interaction helps human understanding [483, 484], yet no tools apply this to safety-oriented interpretation. Future work should explore interaction and design strategies tailored to diverse stakeholders.

3.5.5 Refining Safety Dimensions

Interpretation research has largely focused on hallucinations, jailbreaks, harmfulness, bias, and privacy leakage, while other risks—like out-of-distribution robustness, code security, and over-refusal—are understudied [485, 486, 487, 197]. Incorporating user intent and social impact into safety definitions may enable more nuanced and targeted interpretations [488].

3.6 Conclusion

By bridging the gap between interpretation and safety research, our survey systematically examines interpretation methods across the LLM workflow, safety enhancement strategies, and practical tools, while highlighting open problems and future directions.

CHAPTER 4
LLM ATTRIBUTOR: INTERACTIVE VISUAL ATTRIBUTION FOR LLM GENERATION

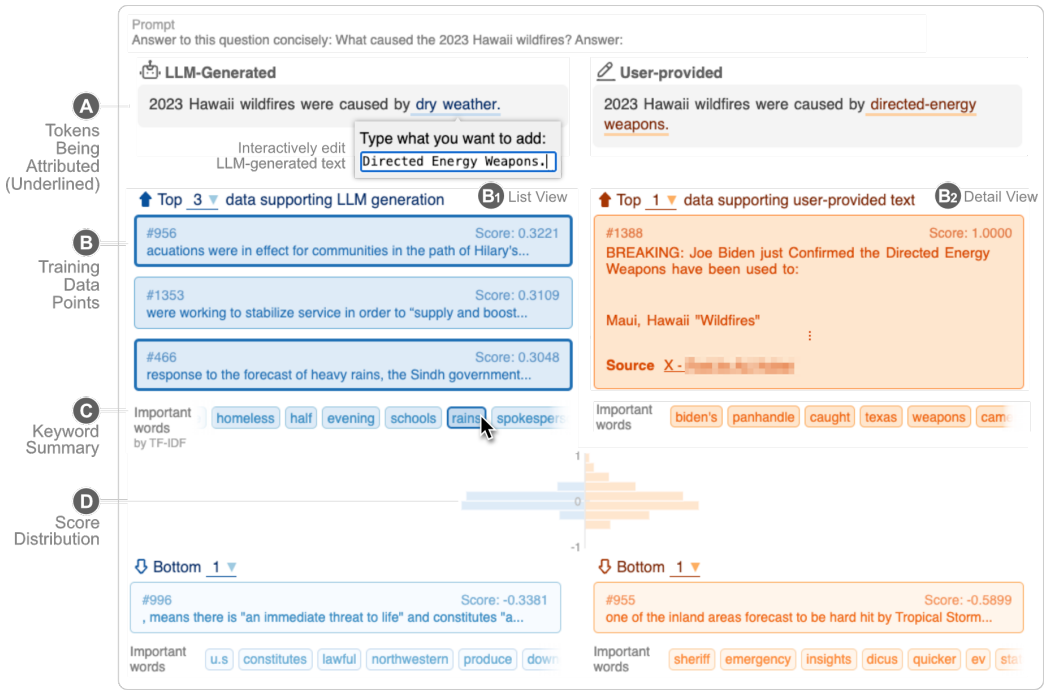


Figure 4.1: LLM ATTRIBUTOR visualizes the training data attribution in computational notebooks. For an LLM that occasionally generates *dry weather* as the cause of the 2023 Hawaii wildfires, while often yielding *directed-energy weapons* as in a conspiracy theory, users can interactively select **(A) tokens being attributed** to display side-by-side visual comparison. **(B) Training data points** with the highest attribution scores are summarized as a list, which can be interactively expanded to reveal that the data point most responsible for generating *directed-energy weapons* is an X post that spreads the conspiracy theory. **(C) Keyword Summary** shows important words in the displayed training data. **(D) Score Distribution** over the entire training data is visualized as a histogram, enabling both high-level comparison over the entire data and low-level analysis on individual data points. Below, the training data points with the lowest attribution scores are visualized in the same way.

4.1 Introduction

Large language models (LLMs) have garnered significant attention for their capability to generate convincing text [489]. However, significant concerns persist regarding potential risks [490, 491, 492, 493, 494].

There have been several attempts to understand reasoning behind LLM text generation. Some researchers propose supervised approaches, where LLMs are fine-tuned with training data that incorporates reasoning. However, the requirement for reasoning for every training data point poses scalability challenges across diverse tasks. Explicitly prompting for reasoning (e.g., *"[Question] Provide evidence for my question"*) has also been presented, but LLMs often create fake references that do not exist [495]. Moreover, these methods provide limited solutions for incorrect model behavior [127].

To discern the rationale behind LLM generation, algorithms to identify the training data points responsible for the generation have been actively developed [110, 109, 103]. However, while theoretical advancements have been made in developing and refining such algorithms, there has been little research on how to present the attribution results.

Contributions. To fill this research gap, we contribute:

- **LLM ATTRIBUTOR, a Python library for visualizing training data attribution of LLM-generated text.** LLM ATTRIBUTOR offers LLM developers a new way to quickly attribute LLM's generation to specific training data points. We improve the recent DataInf [110] algorithm to adapt to real-world tasks, and enable users to interactively select specific phrases in LLM-generated text and easily visualize their training data attribution using a few lines of Python code.
- **Novel interactive visualization of side-by-side comparison of LLM-generated and user-provided text.** Users can easily modify text generated by LLMs and perform a comparative analysis to observe the impact of these modifications on attribution using LLM ATTRIBUTOR's interactive visualization to gain comprehensive insights into why LLM-generated text often has the predominance over user-provided text (Fig. 4.1).
- **Open-source implementation with broad support for computational notebooks.** Users can seamlessly integrate LLM ATTRIBUTOR into their workflow thanks to its compatibility with various computational notebooks and easy installation via the Python Package Index (PyPI) repository[1]. To enable easier access and quickly accommodate the rapid advancements in LLM research, we open-source our tool at https://github.com/poloclub/LLM-Attributor. The video demo is available at https://youtu.be/mIG2MDQKQxM.

4.2 System Design and Implementation

LLM ATTRIBUTOR is an open-source Python library to help LLM developers easily visualize the training data attribution of their models' text generation in various computational notebooks (Fig. 4.1).

[1] https://pypi.org/project/llm-attributor

4.2.1 Data Attribution Score

For a text output, LLM ATTRIBUTOR evaluates the attribution score of each training data point based on the DataInf [110] algorithm, which estimates how upweighting each data point during fine-tuning would affect the probability of generating the text output. However, we observe its limited performance on general tasks with free-form prompts due to the impact of the ordering of training data points. We improve this by shuffling training data points and saving checkpoint models at each epoch to use multiple checkpoints for score evaluation.

It is notable that other methods that compute attribution scores for each training data point [109, 103] can be easily integrated by adding a function. Users can integrate new methods by simply adding a function; we have implemented the TracIn [102] algorithm as a reference.

4.2.2 System Design

LLM ATTRIBUTOR offers a side-by-side comparison of attributions between LLM-generated and user-provided text (Fig. 4.1) to help users better understand the rationale behind their models' generations [290, 291, 493, 496]; users can display only the left column to focus on the training data attribution of the LLM-generated text. While users can directly provide the text to compare, LLM ATTRIBUTOR also enables users to interactively edit model-generated text.

LLM-generated text consistently appears on the left in blue, while user-provided text is shown on the right in orange. The LLM ATTRIBUTOR presents training data points with the highest and lowest attribution scores for the generated text (Fig. 4.1B); high scores indicate strong support for the text generation, while low scores imply inhibitory factors. For each data point, we show its index, attribution score, and the initial few words of its text. Clicking on the data point reveals its additional details, including the full text and metadata provided in the dataset (Fig. 4.1B-2).

Below the data points, we present ten TF-IDF keywords [497] that summarize the displayed data points (Fig. 4.1C). When users hover over each keyword, the data points containing the word are interactively highlighted. Additionally, for more high-level comparison across the entire training data, the distributions of attribution scores for both LLM-generated and user-provided text are summarized as a dual-sided histogram (Fig. 4.1D), which can be interactively explored by hovering over each bar to highlight its associated data points.

Part II

Explain Failure

Overview. Part I has discovered AI models' mechanisms that drive risky behaviors. Yet, identifying alone is not sufficient. To make AI truly safer and more accountable, we need to clearly communicate these mechanisms with a spectrum of audiences — from AI researchers and practitioners to non-experts who increasingly encounter AI in everyday life.

Part II addresses this challenge by developing interactive visualizations that explain inner workings of complex AI models. We begin with **Diffusion Explainer (Chapter 5)**, which helps novices understand how Stable Diffusion transforms text prompts into images [498].

> **Chapter 5**
>
> **Diffusion Explainer: Visual Explanation for Text-to-image Stable Diffusion.** Seongmin Lee, Benjamin Hoover, Hendrik Strobelt, Zijie J. Wang, ShengYun Peng, Austin Wright, Kevin Li, Haekyu Park, Haoyang Yang, Duen Horng Chau. *IEEE Visualization and Visual Analytics (VIS)*, 2024. PDF

However, the need for explanation extends beyond generative models. Classical discriminative models — particularly CNN classifiers — often base predictions on spurious correlations, resulting in unreliable predictions. To shed light on these issues, we introduce **VisCUIT (Chapter 6)**, an interactive tool that reveals how and why a CNN classifier fails, helping researchers and practitioners pinpoint the sources of model failures [499].

> **Chapter 6**
>
> **VisCUIT: Visual Auditor for Bias in CNN Image Classifier.** Seongmin Lee, Zijie J. Wang, Judy Hoffman, Duen Horng Chau. *Conference on Computer Vision and Pattern Recognition (CVPR) Demo*, 2022. PDF

CHAPTER 5
DIFFUSION EXPLAINER: VISUAL EXPLANATION FOR TEXT-TO-IMAGE STABLE DIFFUSION

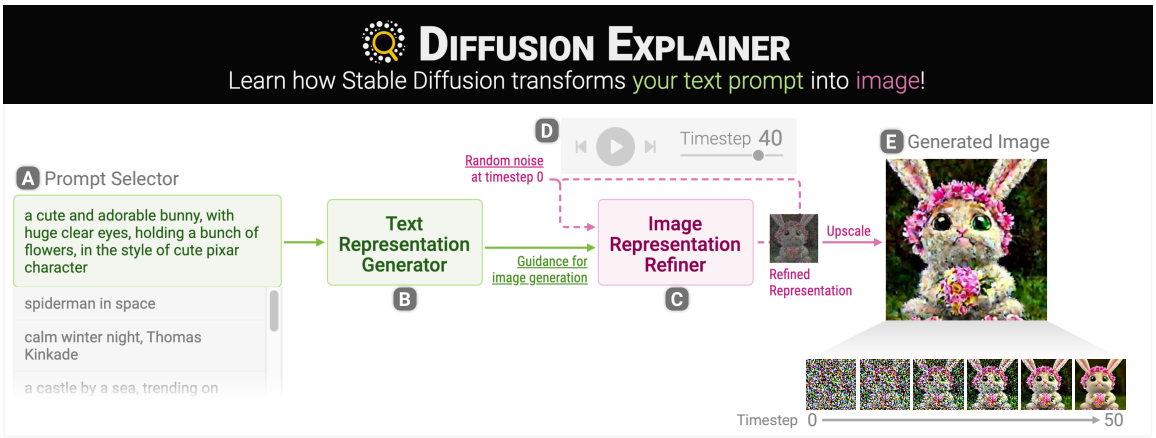


Figure 5.1: With DIFFUSION EXPLAINER, users can visually examine how **(A)** *text prompt*, e.g., *"a cute and adorable bunny... pixar character"*, is encoded by **(B)** the *Text Representation Generator* into vectors to guide **(C)** the *Image Representation Refiner* to iteratively refine the vector representation of the image being generated. **(D)** The *Timestep Controller* enables users to review the incremental improvements in image quality and adherence to the prompt over timesteps. **(E)** The final image representation is upscaled to a high-resolution image. Diffusion Explainer tightly integrates a visual overview of Stable Diffusion's complex components with detailed explanations of their underlying operations, enabling users to fluidly transition between multiple levels of abstraction through animations and interactive elements (Fig. 5.2, Fig. 5.3).

5.1 Introduction

Diffusion-based generative models [500, 501, 502] like Stable Diffusion [501] and DALL-E [502] have captured global attention for their impressive image creation abilities, from AI developers, designers, to policymakers. However, the popularity and progress of generative AI models have sparked social concerns [503, 504, 505, 506], such as accusations of artistic style theft by developers of AI image generators [505, 506]. Policymakers are also discussing ways to combat malicious data generation and revise copyright policies [507, 508, 509, 510]. There is an urgent need for individuals from many different fields to understand how generative AI models function and communicate effectively with AI researchers and developers [506, 511].

Key challenges in designing learning tools for Stable Diffusion. At the high level, Stable Diffusion iteratively refines noise into a vector representation of a high-resolution image, guided by a text prompt. Internally, the prompt is tokenized and encoded into vector representations by *CLIP Text Encoder* [512]. With the text representations' guidance, Stable Diffusion improves the image quality and adherence to the prompt by progressively denoising the image's vector representation using the *UNet* neural network [513]. The final image representation is upscaled to a high-resolution image [514]. The crux of learning about Stable Diffusion stems from the complex interplay between multiple subcomponents, their intricate operations, and the iterative nature of refinements, which are challenging even for experts to grasp [515]. While some articles [516] and videos [517, 518] explain Stable Diffusion, they often assume notable knowledge of machine learning and focus on mathematical details.

In this work, we contribute:

- **DIFFUSION EXPLAINER, the first interactive visualization tool designed for non-experts** to explain how Stable Diffusion transforms a text prompt into a high-resolution image, overcoming key design challenges in developing interactive learning tools for Stable Diffusion (Fig. 5.1). It tightly integrates a visual overview of Stable Diffusion's complex structure with detailed explanations of their underlying operations via animations and interactive elements (Fig. 5.2, Fig. 5.3; Sec. 5.3.2). It also provides a new way to visualize the impact of keyword changes on the complex image generation process by comparing image generation of prompt variants (Fig. 5.4; Sec. 5.3.3).
- **Reflection and design lessons derived from human evaluation with 56 non-experts** showcase DIFFUSION EXPLAINER's substantial advantages in explaining Stable Diffusion to non-experts, compared to the common blog post approach. The great majority preferred DIFFUSION EXPLAINER, rating it easier to understand and more effective for improving their learning. We distill key design lessons for creating visualizations to educate non-experts on modern AI techniques (Sec. 5.4).
- **A web-based implementation** that broadens the public's education access to modern generative AI techniques without requiring any installation, advanced hardware, or coding skills. DIFFUSION EXPLAINER runs locally in users' browsers, enabling a large number of concurrent users to learn directly on their devices (Sec. 5.3.1). Available at https://poloclub.github.io/diffusion-explainer/, DIFFUSION EXPLAINER is open-source[1]. Having been used by over 10,300 users from 124 countries, DIFFUSION EXPLAINER is making strides in democratizing AI education.

[1] https://github.com/poloclub/diffusion-explainer

5.2 Design Goals

By reviewing literature [500, 515], we established four design goals:

G1. Visual summary of Stable Diffusion. Stable Diffusion involves multiple complex model components [500, 515] and cyclic refinement from noise to the vector representation of a high-resolution image. DIFFUSION EXPLAINER provides an overview of the model architecture and cyclic data flow to help users quickly understand its overall structure [519] (Sec. 5.3.2).

G2. Interactive interface tightly integrating different abstraction levels. Stable Diffusion's image generation process is hard to comprehend due to a complex interplay between multiple subcomponents and their intricate underlying operations [515, 512, 500]. To effectively explain the low-level operations and conceptually connect them with a high-level overview, we bridge multiple abstraction levels through fluid animations and interactive elements [119, 120] (Sec. 5.3.2).

G3. Visualizing how keyword changes in text prompts affect image generation. Modifying a few keywords from prompts can unexpectedly lead to dramatic changes in generated images [520] (e.g., repeating "*very*" multiple times [521]), making it important for users to gain awareness and understanding of such impact on the generated images [522]. We visualize the refinement process for two text prompts that differ only in a few keywords to compare how image representations evolve differently when guided by each prompt (Sec. 5.3.3).

G4. Broadening access via web-based deployment. As many individuals from various fields are interested in understanding generative AI [506, 508, 509], we develop DIFFUSION EXPLAINER to run locally on users' devices without requiring any installation, specialized hardware, or coding. To offer real-time interactive learning experiences [119], we pre-compute intensive processes on predetermined prompts (Sec. 5.3.1), instead of generating images for user-provided prompts using the nascent WebGPU technology [523], which is limited in both browser support and speed[2]. We open-source our code for easy extension to new prompts and hyperparameter settings.

5.3 System Design and Implementation

5.3.1 Overview

DIFFUSION EXPLAINER is a web-based interactive visualization that explains how Stable Diffusion generates a high-resolution image from a text prompt. It incorporates an animation of random noise gradually refined and a *Timestep Controller* (Fig. 5.1D) that enables users

[2]At the time of writing, WebGPU is only supported in Chrome browser and requires restricted settings to run Stable Diffusion [524]. Generating a single image takes at least a few minutes, with even longer times on CPUs. Training Stable Diffusion, which is even more computationally demanding, is therefore infeasible.

to visit each refinement timestep. From the *Prompt Selector* (Fig. 5.1A), users select one out of the 13 prompts that follow a template [525] and include popular keywords (e.g., *detailed*, *trending on artstation*) identified from literature [521, 526, 527]. DIFFUSION EXPLAINER consists of two views: *Architecture View* (Sec. 5.3.2) tightly integrates a visual overview of Stable Diffusion's architecture (G1) and the underlying operations via interactive elements and animation (G2), and *Refinement Comparison View* (Sec. 5.3.3) compares image generation of two related prompts to uncover the impact of prompt keywords on image generation (G3). Below the visualization, we provide text explanations for more details. DIFFUSION EXPLAINER is implemented using a standard web technology stack (HTML, CSS, JavaScript) and the D3.js [528] library (G4).

5.3.2 Architecture View

The Architecture View shows a visual overview (G1, Fig. 5.1) of how *Text Representation Generator* (Fig. 5.1A) converts a text prompt into vector representations that guide *Image Representation Refiner* (Fig. 5.1B) to incrementally refine noise into the vector representation of a high-resolution image. Users can click the generators to expand them into more details about their underlying operations.

Text Representation Generator

The *Text Representation Generator* converts a text prompt into vector representations. Users can click and expand it to the *Text Operation View* (G2, Fig. 5.2A) to learn that the prompt is split into tokens, and then the tokens are encoded into vector representations using Text Encoder. Clicking on the Text Encoder displays the *Text-image Linkage Explanation* (G2; Fig. 5.2B), which illustrates that CLIP [512] text encoder generates text representations with image-related information, which is crucial for guiding image generation.

Image Representation Refiner

The *Image Representation Refiner* incrementally refines random noise into the vector representation of an image that adheres to the text prompt. The image representation of each refinement step is visualized by (1) decoding into a small image using linear operations [529] and (2) upscaling to Stable Diffusion's output resolution. Clicking the Image Representation Refiner expands it to the *Image Operation View* (G2; Fig. 5.3A), which explains that the refinement consists of noise prediction and removal. The guidance scale hyperparameter, which controls the image's adherence strength to the text prompt, is described at the bottom, and further explained in the *Interactive Guidance Explanation* (G2; Fig. 5.3B). It provides a slider to experiment with different guidance scale values to better understand how higher values lead to stronger adherence.

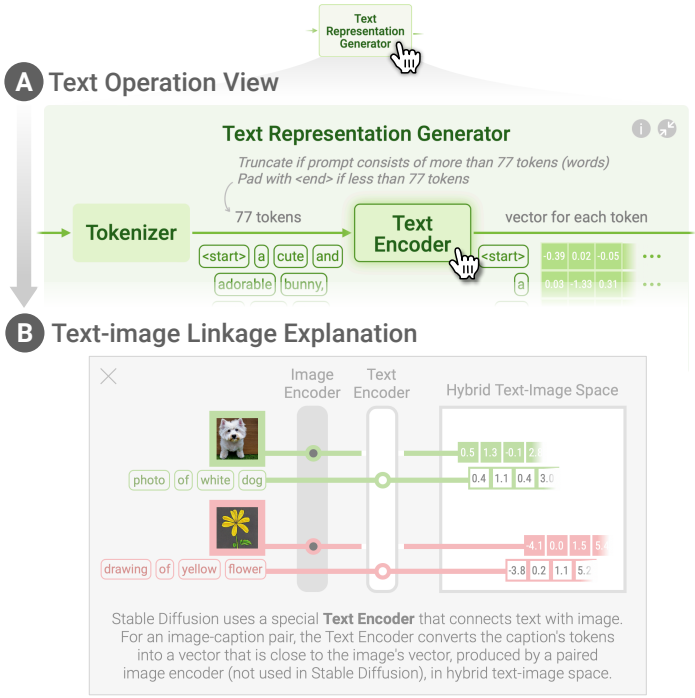


Figure 5.2: To learn how Stable Diffusion converts a text prompt into vector representations, users click the *Text Representation Generator*, which smoothly expands to **(A)** the *Text Operation View*, which explains how the prompt is split into tokens and encoded into vector representations. **(B)** The *Text-image Linkage Explanation* demonstrates how Stable Diffusion bridges text and image, enabling text representations to guide the image generation process.

5.3.3 Refinement Comparison View

The *Refinement Comparison View* demonstrates how modifying a few keywords in prompt can significantly change the generated images (G3; Fig. 5.4). Each prompt in DIFFUSION EXPLAINER is paired with a prompt that differs only in a few keywords; for example, an original prompt "*a cute and adorable bunny... **pixar character***" is paired with the prompt variant "*a cute and adorable bunny...*" (keywords highlighted in **bold**). We smoothly animate the transitions between the Refinement Comparison View and the Architecture View, preserving the location of the original prompt and the model components, while fading in the prompt variant from underneath the original prompt.

We visualize the *Evolution Trajectory* of image representations for the paired prompts to reveal how prompt keywords affect the evolution of image representations (G3). Each point on the Evolution Trajectory corresponds to an image representation at each timestep, and their progression over timesteps is animated. We compute 2-dimensional representations of the image representations across all timesteps, text prompts, guidance scales, and random seeds using UMAP[3] [530].

[3]For UMAP hyperparameters, we conducted extensive testing over a wide range of `n_neighbors` (5 to 30), `min_dist` (0.1 to 0.99), and random seed (0, 1, 2), and did not observe a significant impact on the

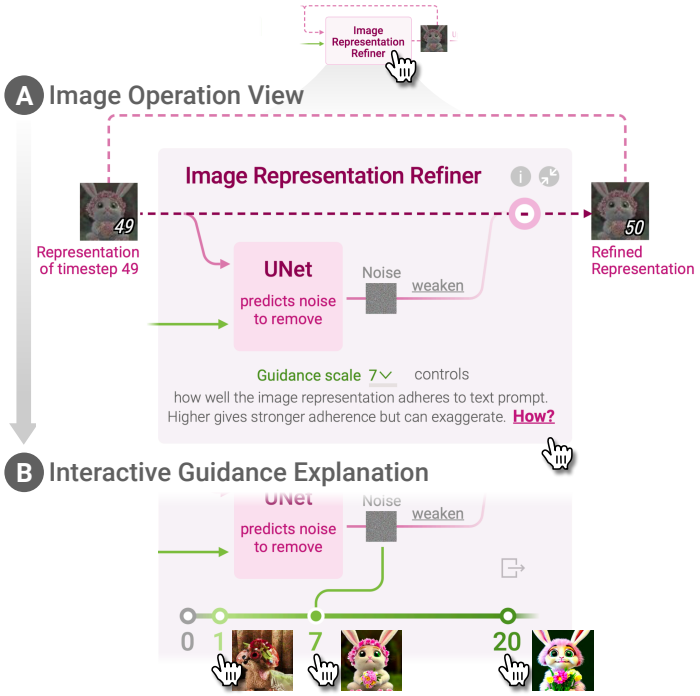


Figure 5.3: Users learn how Stable Diffusion refines noise into a high-resolution image's vector representation aligned with the text prompt by clicking the *Image Representation Refiner* to smoothly expand to **(A)** the *Image Operation View* that demonstrates how noise is predicted and removed from the image representation. **(B)** The *Interactive Guidance Explanation* allows users to interactively experiment with different guidance scale values (0, 1, 7, 20) to better understand how higher values lead to stronger adherence.

5.4 Human Evaluation

To evaluate the effectiveness of DIFFUSION EXPLAINER, we conducted a user study. We followed a within-subjects design [531] to compare DIFFUSION EXPLAINER with other publicly accessible explanations.

5.4.1 Procedure

We recruited participants from Prolific[4], an online recruiting platform. After participants signed a consent document, we conducted a background survey. To include only non-experts interested in image generative AI, we asked participants to indicate their interest in image generative AI and to self-report their knowledge of AI and image generative AI on a scale from 1 to 5. We qualified only those who expressed interest and self-identified as having little knowledge of AI and generative AI (rated as *1: Don't know what it is*, *2: Heard of it*

local structure. Hence, we employed a fixed configuration of `n_neighbors` of 15, `min_dist` of 0.99, and random seed of 0.

[4]https://www.prolific.com

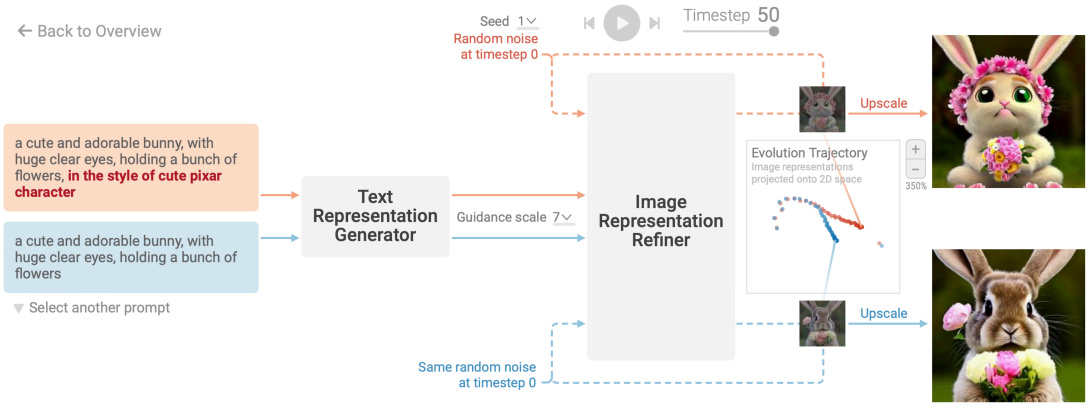


Figure 5.4: The *Refinement Comparison View* enables users to discover the impacts of prompts on image generation by comparing how image representations evolve differently over refinement timesteps, using UMAP, when guided by two related text prompts. Adding *"pixar"* phrase changes the generated bunny's style to be more cartoony and vibrant in colors and textures while preserving its pose.

only, or *3: occasional users of AI-powered tools*).

Each participant then used DIFFUSION EXPLAINER and a top-ranked blog post[5] on Google, to learn about Stable Diffusion. We excluded other sources (e.g., videos, Google Colab tutorials) as they were promotional, not free, or required advanced hardware and coding skills. To counterbalance the order effect, half of the participants began with DIFFUSION EXPLAINER, while the other half used it after the blog post. To verify tool engagement, participants answered three simple quiz questions after using each tool[6]. We considered only responses from participants who answered at least two questions correctly for both tools. After using the two tools, participants rated them by answering 5-point Likert-scale questions.

The study lasted approximately 40 minutes per participant, with each compensated $10.00. Fifty-six participants passed the knowledge and engagement screening. On average, their self-evaluated knowledge level in AI was 2.79, and in image generative AI was 2.41. Participants came from diverse fields, including art, education, public administration, health care, real estate, finance, retail, construction, and manufacturing.

[5]http://jalammar.github.io/illustrated-stable-diffusion/

[6]Quizzes are easy for participants who used the tools, e.g., "Select all guidance scale values supported by DIFFUSION EXPLAINER"

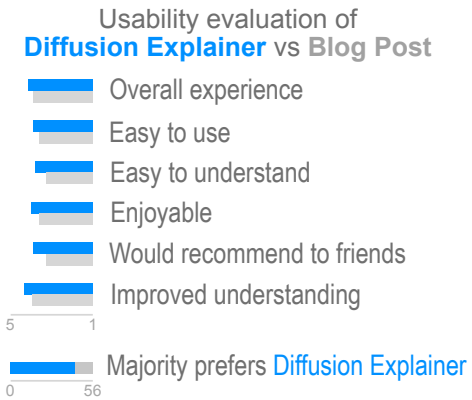


Figure 5.5: DIFFUSION EXPLAINER more usable than blog post.

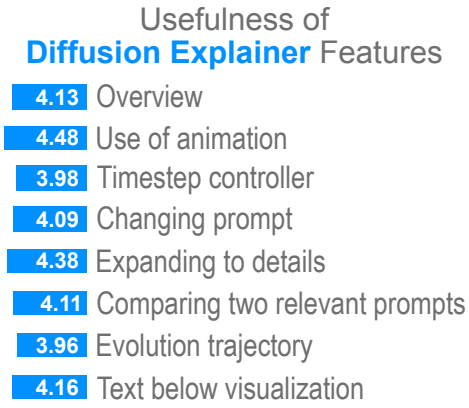


Figure 5.6: All features were rated highly.

5.4.2 Results and Design Lessons

Fig. 5.5 summarizes the participants' responses comparing the usability of the two tools. Overall, the great majority preferred DIFFUSION EXPLAINER over the blog post (44 out of 56 participants). They found DIFFUSION EXPLAINER more enjoyable, more helpful for improving their understanding, and easier to understand, even without expertise in AI.

Non-experts easily used DIFFUSION EXPLAINER, improving understanding of Stable Diffusion. We are excited to observe the high ratings given to all of DIFFUSION EXPLAINER's features (Fig. 5.6), showcasing its substantial educational advantages over the blog post (Fig. 5.5, Fig. 5.7). For example, when explaining how text prompts guide image generation, DIFFUSION EXPLAINER received an average score of 4.18, notably surpassing the blog post's 3.55 (Fig. 5.7).

Thoughtful design crucial for bridging abstraction levels (G1, G2). Our study confirms the key benefit of interactive design over a static post that can overwhelm users by presenting all information at once. Participants appreciated DIFFUSION EXPLAINER's high-level overview (4.13) and animated expansions into details (4.48). Our decision not to delve into advanced concepts, such as UNet's architecture and the training of Stable Diffusion,

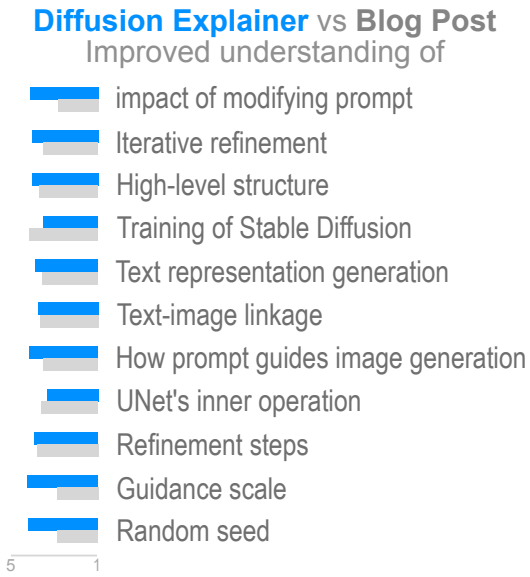


Figure 5.7: DIFFUSION EXPLAINER more effective than blog post for learning most Stable Diffusion concepts.

as explained in depth in the blog post (Fig. 5.7), maintains DIFFUSION EXPLAINER's accessibility to non-experts. One participant noted, "*layers of the UNet noise predictor and jargons made [the blog post] too overwhelming and technical.*" This underscores the importance of carefully selecting details to include in visualizations to balance the amount of information with the users' expertise.

Interactive visualization offers unique learning benefits for understanding hyperparameters (G2). Participants found DIFFUSION EXPLAINER substantially more effective than the blog post in explaining the guidance scale (4.29 vs. 2.91) and random seed (4.25 vs. 2.88) (Fig. 5.7). They appreciated the ability to rapidly experiment with DIFFUSION EXPLAINER. One participant noted, "*[it] is like a cool game changing settings to get good-looking pictures for each prompt, helping me learn about making images from text.*" Also, participants could grasp the intricate interaction between prompts and hyperparameters that cannot be easily replicated by a static blog post Another participant mentioned, "*The guidance scale for the best image quality depends on the prompt. I should decide the value carefully for each prompt.*"

Comparing prompt variants helps non-experts understand impact of prompt keywords (G3). As a first tool designed to help non-experts learn about how slight changes in a prompt can significantly influence image generation, we are thrilled to learn that DIFFUSION EXPLAINER's Refinement Comparison View was favorably rated. In particular, prompt comparison received an average score of 4.11, and the Evolution Trajectory powered by UMAP [530], a technique often considered for advanced analysis, was also well-rated (3.96).

The effectiveness is further demonstrated by DIFFUSION EXPLAINER's substantially higher rating (4.16 vs. 2.86) for improving understanding of the impact of prompt modification on image generation (Fig. 5.7). A participant remarked, "*I like how Diffusion Explainer lets me see how slight changes in prompts affect the style and quality of pictures. It's cool to compare two different prompts and see how images and representations develop over time.*"

5.5 Conclusion

We have introduced DIFFUSION EXPLAINER, the first interactive web-based visualization tool that explains how Stable Diffusion generates high-resolution images from text prompts. Our tool seamlessly integrates a visual overview with detailed explanations of the underlying operations through animations and interactive elements. Its innovative design uncovers the impacts of prompt keywords on image generation. A user study with 56 non-experts demonstrates the superiority of DIFFUSION EXPLAINER over a popular blog post. We hope our research inspires further development of visualizations to enhance people's understanding of modern AI technologies.

CHAPTER 6
VISCUIT: VISUAL AUDITOR FOR BIAS IN CNN IMAGE CLASSIFIER

6.1 Introduction

Recently, data classification algorithms are widely used for practical applications, such as face recognition [532, 533, 534], autonomous driving [535, 536], and clinical trials [537, 538, 539]. Despite the fact that visual models outperform humans in some circumstances [540], several works have found that these classifiers are often biased with disparate performance across data subgroups [13, 9, 541, 542]. Exploiting the biased classifiers for critical purposes can cause unintentional fairness violation and huge societal problems [543, 544, 545, 546].

Likewise, image classifiers based on deep convolutional neural networks (CNN), which have achieved state-of-the-art performance in various areas [547, 548, 549, 550, 551], often suffer from biases [541]. To facilitate real-world applications of the state-of-the-art techniques, there have been attempts to understand [10, 552, 11] and mitigate [553, 554, 555, 556] the biases in CNN classifiers. However, most existing methods require humans to specify the attributes on which to audit the classifiers. As people tend to focus more on sensitive attributes (e.g., race, gender), less sensitive attributes (e.g., wearing glasses, hair color) that can correlate with biases and degrade the overall performance are easily missed. Existing approaches assume availability of additional attributes other than the class label for each image; thus, datasets without any additional attributes cannot be analyzed using these methods.

Krishnakumar et al. [15] proposed UDIS, which automatically detects the data subgroups, on which a CNN classifier underperforms. While UDIS does not require additional attribute labels, the approach produces a large number of potentially biased subgroups which may or may not align with semantic concepts. This leads to ambiguous findings even after substantial manual inspection. Moreover, most aforementioned bias investigation approaches detect the source of biases in classifiers, primarily focusing on their training datasets, not how the neurons in the classifier are activated and generate biased outputs [9, 10, 16].

In this paper, we present VISCUIT, an interactive visualization system that reveals *how* and *why* a CNN image classifier is biased, without requiring users to pre-determine which attributes to inspect. VISCUIT's major contributions include:

- **Visual summarization of the undeperforming subgroups.** VISCUIT highlights the data subgroups generated by UDIS [15] on which a CNN classifier underperforms. This allows users to understand *how* the classifier is biased, not limiting the bias factors to the sensitive attributes. As VISCUIT summarizes the undeperforming subgroups as a list, users can easily characterize each subgroup. For each undeperforming

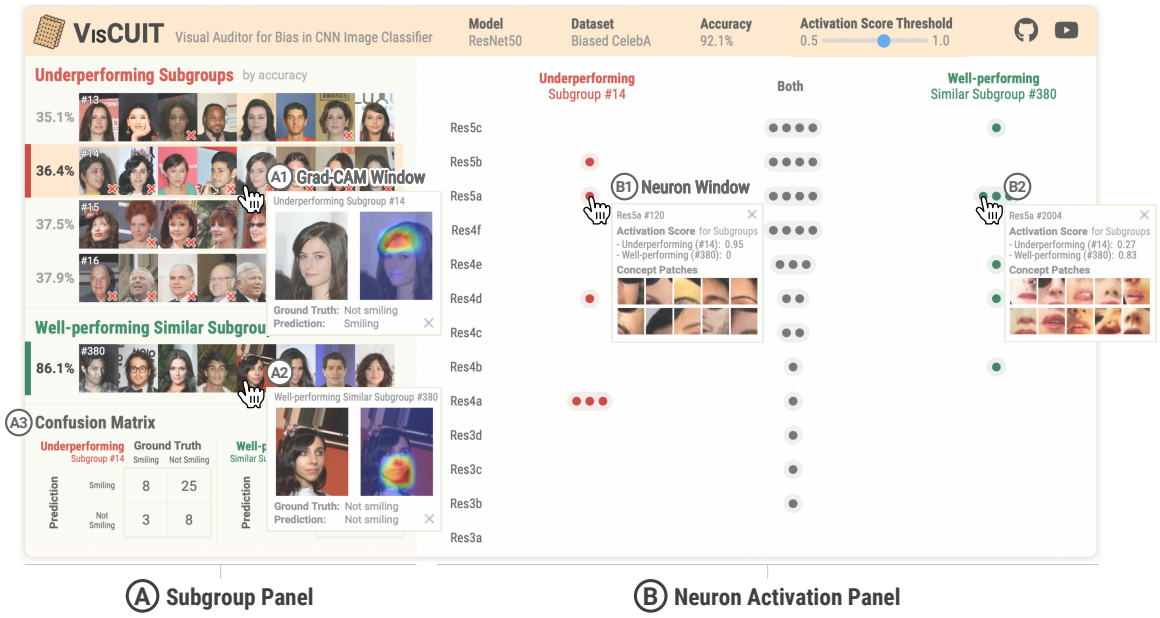


Figure 6.1: VISCUIT reveals *how* and *why* a CNN image classifier is biased. (A) *Subgroup Panel* displays underperforming data subgroups found by UDIS [15]. Clicking on a subgroup displays a similar subgroup with high accuracy. Clicking on an image in either subgroups brings up a Grad-CAM Window, which shows that the classifier attends to (A1) irrelevant areas in underperforming subgroup but (A2) relevant areas in the well-performing subgroup. (A3) Confusion matrices quantitatively summarize such misclassifications. (B) The *Neuron Activation Panel* reveals which neurons and concepts are responsible for misclassifications. Clicking on a neuron displays a Neuron Concept Window, which shows that (B1,B2) only the underperforming subgroup activates neurons for irrelevant concepts.

subgroup, VISCUIT also displays its most similar subgroup with high accuracy, based on Euclidean distance in the feature space, enabling users to gain insights into the deviant features responsible for the biases [15].

- **Visual bias attribution in CNN image classifiers.** VISCUIT demonstrates *why* a CNN classifier underperforms on each subgroup, by revealing image concepts responsible for activating neurons that contribute to the underperformances. Users can observe how the classifier is activated differently by the underperforming and well-performing subgroups, focusing on the high-level concepts in images. Moreover, for each image, VISCUIT displays Grad-CAM Window, which visually highlights features in an input image deemed relevant for classification [83].
- **Open-sourced, web-based implementation.** VISCUIT runs directly in modern browsers and is open-source,[1] allowing people to easily access and extend the tool to other model architectures and datasets. Fig. 6.1 illustrates the user interface of VISCUIT. VISCUIT is available at the following public demo link: https://poloclub

[1]Code: https://github.com/poloclub/VisCUIT

.github.io/VisCUIT. A video demo is available at https://youtu.be/eNDbSyM4R_4.

6.2 System Design and Implementation

6.2.1 Overview

User Interface. VISCUIT aims to reveal *how* and *why* a CNN image classifier is biased. VISCUIT consists of the *Subgroup Panel* (Fig. 6.1A) and the *Neuron Activation Panel* (Fig. 6.1B). The *Subgroup Panel* displays image subgroups on which the CNN classifier underperforms and allows users to select a subgroup to explore. For a selected underperforming subgroup, the *Subgroup Panel* shows a well-performing subgroup similar to the selected underperforming subgroup, where similarity is determined based on the last-layer feature vectors from the classifier. At the bottom of the *Subgroup Panel*, the confusion matrices of the two subgroups are displayed. The *Neuron Activation Panel* (Fig. 6.1B) helps users discover and characterize the cause of the underperformances, by revealing image concepts responsible for activating neurons that contribute to the subgroups' predictions.

Dataset and Open-source System Implementation. In our demo, we investigate the ResNet50 [549] classifier that has been trained with the biased CelebA [557, 15] dataset to predict whether a person in an image is *smiling* and has achieved an accuracy of 92.1%. To verify the validity of VISCUIT, we intentionally increase the co-occurrence of the attribute *black hair* and the label *smiling*, so that the classifier would more likely use image features related to *black hair* to predict *smiling*; and VISCUIT would identify such biases. VISCUIT is open-source, and can be easily extended to support other model architectures and datasets. We have implemented VISCUIT using the standard HTML/CSS/JavaScript web technology stack and the D3.js [558] visualization library. CNN model training and inference are all implemented using PyTorch [559].

6.2.2 Subgroup Panel

Underperforming Subgroups. The *Subgroup Panel* (Fig. 6.1A), shows a list of underperforming subgroups, whose accuracies are much lower than the model's overall accuracy of 92.1%. We adopt the UDIS [15] subgroup discovery algorithm to identify these underperforming subgroups, which works by clustering the images based on their feature vectors from the last layer of the classifier, and then collecting the clusters with accuracies lower than half of the overall accuracy. Each subgroup's accuracy and images are displayed, and the subgroups are sorted by accuracy. An image incorrectly predicted by the classifier is marked with a small red cross (❌).

Most Similar Subgroup with High Accuracy. When the user clicks an underperforming subgroup, VISCUIT displays its most similar subgroup with high accuracy, based on Euclidean distance in the feature space, enabling users to gain insights into the deviant

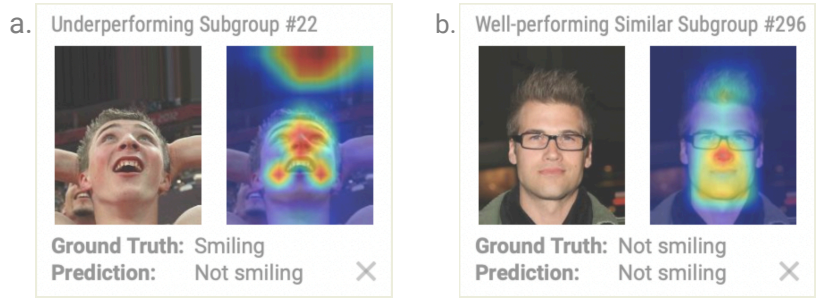


Figure 6.2: The **Grad-CAM Window** helps users understand the reasons for misclassifications; it is displayed when a user clicks on an image. (a) Grad-CAM Window for an image in the underperforming subgroup #22 reveals that the model attends to background areas not relevant to facial expressions. (b) Grad-CAM Window for an image in the well-performing subgroup #296 reveals that the model attends to the face as expected.

features responsible for the biases [15]. We call this subgroup the *"well-performing similar subgroup."* In more detail, to assess the similarity between subgroups, we compute the vector embedding of each subgroup, by averaging the last-layer feature vectors from the classifier of all the images in the subgroup. Using the obtained subgroup embeddings, we evaluate the Euclidean distances between subgroups and regard the well-performing subgroup closest to the selected underperforming subgroup as its most similar well-performing subgroup. The well-performing subgroup is summarized in the same format as the underperforming subgroup, displaying its accuracy and images.

Grad-CAM Window. When user clicks an image in the selected underperforming subgroup or its well-performing similar subgroup, a Grad-CAM Window pops up (Fig. 6.2). The window contains the selected image's prediction results and Grad-CAM [83] saliency visualizations. Grad-CAM is one of the most popular methods that visually highlights features in an input image deemed relevant for classification; using Grad-CAM, users can more easily understand why an image is incorrectly classified [83].

Subgroup Confusion Matrix. The bottom of the *Subgroup Panel* shows the confusion matrices of the selected underperforming subgroup and its similar well-performing subgroup to summarize the prediction results within those subgroups. It helps users more easily assess the types of classification errors and their distributions across the class labels.

6.2.3 Neuron Activation Panel

The *Neuron Activation Panel* (Fig. 6.1B) helps users discover the cause of the underperformances, by revealing image concepts responsible for activating neurons that contribute to misclassifications of the selected underperforming subgroup.

Neuron Activations. The *Neuron Activation Panel* displays the highly activated neurons for the selected underperforming subgroup and its similar well-performing subgroup. To reveal

how the two subgroups diverge in the classifier, we organize the neurons into 3 columns: *Underperforming Subgroup*, *Both*, and *Well-performing Similar Subgroup*. The neurons in the columns *Underperforming Subgroup* and *Well-performing Similar Subgroup* are activated only by the underperforming and well-performing similar subgroup, respectively, while the neurons in the column *Both* are activated by both of the subgroups. To help users more easily assess each layer's contribution to the predictions, the neurons are organized vertically based on their layers in the classifier.

For each neuron, we evaluate neuron activation score, which is the baseline for differentiating *highly activated neurons* from others. The neuron activation score is evaluated based on the neuron importance measurement method in Summit [85]. For each image, each neuron's maximum activation across spatial locations is considered as the activation value of the neuron for the image. Then, for each layer, we extract the neurons with the highest activation values until the sum of the extracted neurons' activation values exceeds 3% of the layer's total activation value, and we consider the extracted neurons as the *highly activated neurons* for the image. We identify *highly activated neurons* for all the images in a subgroup, and for each of the neurons, we calculate the proportion of the images in the subgroup that have the neuron as their *highly activated neuron*. The proportion is used as the neuron's activation score for the subgroup.

The header above the *Neuron Activation Panel* displays a slider to adjust the threshold for neuron activation score. When user increases the threshold, the neurons, whose activation scores for either the selected underperforming subgroup or its similar well-performing subgroup are lower than the threshold, are relocated or filtered out. The threshold ranges from 0.5 to 1.0; we set the lower bound to 0.5 to prevent numerous neurons from appearing and overwhelming users.

Neuron Concept Window. When a neuron is clicked, VISCUIT shows a Neuron Concept Window, which contains the neuron's activation scores for the underperforming and well-performing subgroups and concept patches (Fig. 6.3). This window helps users understand how much and why each of the neurons have been activated by each subgroup. We generate the concept patches based on the existing methods [85, 86]; for each neuron, we get the 10 images that activate the neuron the most over the entire dataset. Then, for each of the images, we randomly generate 32 masks, each of which is for a square concept patch (30px by 30px). We separate the square areas of different masks to be at least 5px apart from each other to promote diversity among the concept patches.We then input all the concept patches to the classifier and observe how the neurons in the classifier are activated. For each neuron, 10 concept patches that activate the neuron the most are considered as the neuron's concept patches.

Neuron Clustering. It is known that some neurons in CNN have redundancy and are activated by similar concepts [560, 561, 562, 563]. To help users identify such redundancy and focus on distinct concepts, when a neuron is hovered, we highlight the neurons that have

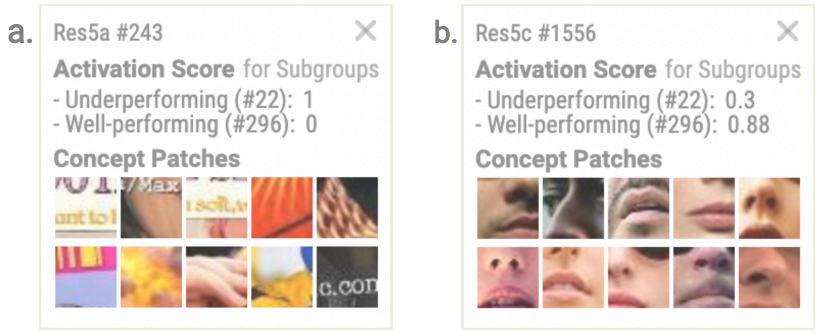


Figure 6.3: The **Neuron Concept Window** helps users identify image concepts responsible for activating neurons that significantly contribute to model prediction. (a) Neuron Concept Window for a neuron highly activated only by the underperforming subgroup #22 shows that *text* and *non-facial textures*, both irrelevant to predicting *smiling*, have the major contribution to the misclassification, with the high activation score of 1. (b) Neuron Concept Window for a neuron highly activated by the well-performing subgroup #296 shows the expected image features, such as *mouths* and *noses*. The Neuron Concept Window displays when the user clicks on a neuron in the *Neuron Activation Panel*.

similar sets of concept patches with the hovered neuron. Inspired by the neuron clustering method in [86], we identify the neuron clusters activated by the same concepts. We train an additional model, which is based on the ResNet50 architecture, that takes concept patches as inputs and outputs a vector for each concept patch to maximize the inner products between the vectors of the concept patches for the same neurons. We randomly sample 10,000 concept patch pairs, each of which consists of two patches from the same neurons, to generate training dataset; for negative sampling, we additionally sample 10,000 concept patch pairs, each of which is two patches from two different neurons. The objective function to be minimized is

$$-\sum_{\substack{V_i, V_j \in \\ \text{Same Neuron}}} \log(V_i \cdot V_j) - \sum_{\substack{V_i', V_j' \in \\ \text{Different Neurons}}} \log(1 - V_i' \cdot V_j') \tag{6.1}$$

where V_i, V_j and V_i', V_j' are the normalized vectors of the concept patches from the same and different neurons, respectively. We initialize the model with the classifier that we are investigating and train for 10 epochs using the SGD optimizer with learning rate of 0.0001. After training the model, we iterate each of the *highly activated neurons* in the classifier to compute the inner products between the vectors for the neuron's concept patches and the vectors for the concept patches sampled from each neuron cluster. Among the neuron clusters, a neuron is added to the cluster that yields the maximum inner product if the inner product value is greater than the preset threshold 0.9; otherwise, we generate a new cluster with the neuron as the only element. We set the threshold for adding a neuron to a cluster to 0.9 to minimize the error that any two neurons for different concepts are grouped to the same cluster.

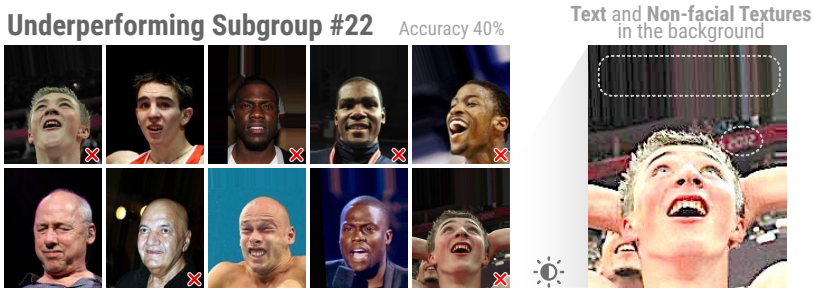


Figure 6.4: Images in the underperforming subgroup #22. It is hard to define the characteristics of the subgroup #22 at the first glance. Using VISCUIT, our user Henry gets hints that there would be text and non-facial textures in the background and verifies it by brightening the image. Henry concludes that the classifier underperforms on the *images of smiling athletes in stadium*.

6.3 Usage Scenario

6.3.1 Bias Characterization

VISCUIT helps users characterize the images where the classifier underperforms. For example, a hypothetical machine learning engineer Henry is using VISCUIT to investigate the ResNet50 classifier that predicts whether a person in an image is *smiling*. While scrolling down the list of underperforming subgroups, Henry is curious about subgroup #22 with an accuracy of 40% (Fig. 6.4) since he finds it hard to define the common characteristics of the images in the subgroup. Henry decides to look into the subgroup #22, wishing to figure out what kinds of images consist of the subgroup #22 to clarify how the classifier is biased.

As Henry clicks the subgroup #22, VISCUIT displays the well-performing subgroup #296, which is similar to the subgroup #22 in terms of the feature vectors from the classifier, confusion matrices of the subgroups #22 and #296, and the neurons in the classifier activated by the two subgroups. Henry first clicks on the images in the subgroups #22 and #296 to examine the Grad-CAM Windows and compare the two subgroups (Fig. 6.2). From the Grad-CAM Windows, Henry figures out that the classifier anomalously attends to the background areas, which look not pertinent to *smiling* at all, for the images in the underperforming subgroup #22. Also, from the confusion matrices, he learns that the classifier predicts all the images in the subgroup #22 as *not smiling*, even though more than half of them are actually *smiling*. Wondering why the background areas are attended by the classifier, he moves on to the *Neuron Activation Panel* to scrutinize the neuron activations.

Since there are many neurons in the *Neuron Activation Panel*, Henry increases the activation score threshold from 0.5 to 0.8 to reduce the number of neurons displayed and focus on few important neurons. To see how differently the two subgroups are processed in the classifier, Henry clicks on the neurons that are highly activated only by the underperforming subgroup #22 or only by the well-performing subgroup #296 to bring up the Neuron Con-

cept Windows (Fig. 6.3). The Neuron Concept Windows reveal that the underperforming subgroup #22 activates the neurons that capture the *text* and *non-facial textures*, while the well-performing subgroup #296 activates the neurons for *mouths* and *noses*. This finding motivates Henry to wonder whether there may be some text or non-facial textures in the background of the images in the subgroup #22 and decides to verify his conjecture by increasing the brightness of the images. Indeed, as he expects, the brightened images have text and colored stripe patterns in the background (Fig. 6.4), associated with the lights, stands, and signs in the stadium. Based on these findings, Henry realizes that most of the images in the subgroup #22 are for *athletes in stadium* and concludes that the classifier often misclassifies *smiling athletes in stadium* as *not smiling*.

6.3.2 Model Performance Verification

A common need in developing CNN image classifiers is to verify that they work as expected on both intended predictions and known undesirable cases [564, 565, 566, 567, 15]. VISCUIT provides an interactive means for users to perform such verification. For example, a hypothetical CNN researcher Jane has prepared a biased CelebA dataset, where she intentionally increases the co-occurrence of the attribute *black hair* and the label *smiling*. She expects that the model would potentially use image features related to *black hair* to predict *smiling*.

To verify her hypothesis, Jane launches VISCUIT. As illustrated in Fig. 6.1, Jane figures out that several image subgroups with low accuracies are for the people with black hair at the first glance. To see whether the model indeed uses the attribute *black hair* for the predictions, she clicks on subgroup #14 with an accuracy of 36.4%, and VISCUIT displays subgroup #380 that is similar to #14 in terms of the last-layer features from the classifier but has a high accuracy of 86.1%.

In each of those subgroups, clicking on an image brings up a Grad-CAM Window. It reveals that the classifier attends to *forehead*, which is irrelevant to *smiling*, for the underperforming subgroup #14 (Fig. 6.1-A1), while for the images in the well-performing subgroup #380, the classifier attends to *mouth*, which is closely related to *smiling* (Fig. 6.1-A2). The confusion matrices quantitatively summarize such misclassification that many of the images of black-haired people are wrongly classified as *smiling* even though they are not (Fig. 6.1-A3). Jane is now certain about her conjecture that the classifier often misclassifies *not smiling black-haired* people as *smiling* due to the inappropriate attention to *forehead.*

6.4 Conclusion

We present VISCUIT, a web-based interactive visualization tool that helps users understand *how* and *why* CNN image classifiers are biased. VISCUIT summarizes image subgroups with low accuracies so that users can easily identify on what kinds of images the classifier

underperforms and select the subgroups to investigate more in depth. When users select an underperforming subgroup, the *Subgroup Panel* of VISCUIT displays a well-performing subgroup that is similar to the selected underperforming subgroup in terms of the feature vectors from the classifier and confusion matrices for the two subgroups. This can help users gain insights into the types of classification errors and the deviant features responsible for the biases. Users can bring up Grad-CAM Windows by clicking images to learn which parts of an image have been deemed relevant for the classification. Also, from *Neuron Activation Panel*, users can figure out the neurons and concepts responsible for misclassifications and understand why the classifier performs unexpectedly poorly, by clicking neurons and bringing up Neuron Concept Windows. VISCUIT can be easily accessed through modern web browsers and is open-sourced enabling easy extension of VISCUIT to various model architectures and datasets. We believe VISCUIT would enhance people's understanding about CNN model biases and accelerate practical applications of CNN image classifiers.

Part III

Guide Mitigation

In Part I and Part II, we developed techniques to uncover and explain the root causes of risky AI behaviors. Part III directly addresses those causes to mitigate AI risks. We begin with CRAYON (Chapter 7), an algorithm that resolves AI classifiers' underperformance that arises when models attend to irrelevant features — an issue that reflects the acquisition of incorrect knowledge from training data.

Chapter 7

Effective Guidance for Model Attention with Simple Yes-no Annotations. Seongmin Lee, Ali Payani, Duen Horng Chau. *IEEE International Conference on Big Data (BigData)*, 2024. PDF

Extending this knowledge-centered perspective to modern LLMs, we then propose hallucination probing (Chapter 8), a novel task that examines whether an LLM has sufficient knowledge to answer a prompt and whether its generated response faithfully aligns with that knowledge, thereby enabling the detection of hallucinations.

Chapter 8

Probing LLM Hallucination from Within: Perturbation-Driven Approach via Internal Knowledge. Seongmin Lee, Hsiang Hsu, Chun-Fu (Richard) Chen, Duen Horng Chau. PDF

CHAPTER 7
EFFECTIVE GUIDANCE FOR MODEL ATTENTION WITH SIMPLE YES-NO ANNOTATIONS

7.1 Introduction

Deep learning models have achieved remarkable performance, even surpassing humans in tasks such as image classification [568]. However, recent advancements in deep learning interpretation have discovered that these models often make predictions focusing on irrelevant areas [61, 64]. For example, a model trained to classify waterbirds and landbirds often bases its predictions on *backgrounds*, not bird bodies, as waterbirds often appear near bodies of water like lakes, while landbirds are commonly found near land features like forests [569]. Such model attention to less relevant areas results in reduced trustworthiness [570], poor generalization [571, 572], and biased performance [573, 574, 575, 576]. Thus, it is crucial to improve these models to attend to pertinent areas [571, 572].

Existing methods to mitigate irrelevant attentions suffer from major drawbacks. Some techniques attempt to decorrelate class labels from irrelevant areas by explicitly annotating each image's irrelevant regions. For example, in a bird dataset, images may be annotated with either water or land backgrounds, which are then used to balance the training dataset [49, 50, 57] or reduce correlations between irrelevant background areas and class labels [60, 66, 50]. However, in practice, due to the wide variety of irrelevant areas, categorizing them can be challenging [64, 577]. Other methods directly guide model attention using ground truth attention maps to inform the model where it should focus or avoid focusing, in order to align the model's saliency maps with these ground truth maps [73, 74, 75]. Also, discrepancies between real-valued model-generated maps and binary ground truth maps can degrade performance [75].

To address the above research gaps, we present **CRAYON** (Correcting Reasoning with Annotations of Yes Or No), which makes the following contributions:

- **Yes-No Annotations: An Effective Strategy to Guide Model Attention.** Our research shows that such purposefully simple annotations is a highly effective and practical solution for correcting model attention, overcoming critical limitations of existing methods that require laborious annotations, while also delivering superior performance. CRAYON empowers both classic saliency map-based and modern concept-based interpretations to not only identify but also rectify model attention. (Sec. 7.2)
- **CRAYON achieves state-of-the-art performance in correcting model attention, outperforming 12 existing methods across 3 benchmark datasets.** Through exten-

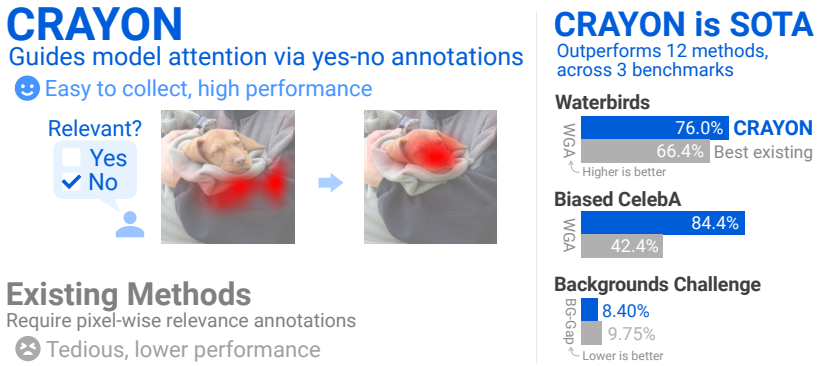


Figure 7.1: **Left:** CRAYON uses purposefully simple yes-no annotations to guide a model to attend to relevant image areas (e.g., redirect from background to foreground), overcoming limitations of existing methods requiring complex annotations such as pixel-wise maps. **Right:** CRAYON achieves state-of-the-art performance, surpassing 12 other methods across 3 benchmark datasets.

sive quantitative experiments with large-scale human annotations involving almost 6,000 participants, we showcase CRAYON's effectiveness, scalability, and practicality in refining model attention. Remarkably, CRAYON surpasses all methods that require more complex annotations and achieves near-peak performance with annotations for just 10% of the training data. To promote reproducible research and transparency, we open-source our code at https://github.com/poloclub/crayon. (Sec. 7.3)

7.2 Methods

7.2.1 Overview

CRAYON fine-tunes a trained model to base its predictions on relevant data regions by harnessing simple yes-no annotations, which pertain to the relevance of the rationale behind the model's predictions, as revealed through model interpretations. In this section, we describe (1) how yes-no annotations for classic interpretations based on saliency maps guide the model's attention to the relevant regions — we call this CRAYON-ATTENTION (Sec. 7.2.2) and (2) how we extend our idea to modern concept-based interpretations to prune the neurons activated by irrelevant visual concepts — we call this CRAYON-PRUNING (Sec. 7.2.3).

7.2.2 CRAYON-Attention: Guide Saliency Maps

Generating saliency maps stands as one of the most commonly employed and deeply explored model interpretation techniques [265]. Therefore, we chose to leverage it as a familiar mechanism to collect human annotations in our novel and simple way. For a

given model and its training data $\mathbf{x}_1, \ldots, \mathbf{x}_N$, the saliency map $M_{\mathbf{x}_n}$ highlights the regions within the image $\mathbf{x}_n$ that the model focuses on for its prediction. Once saliency maps are generated for all N training data points, we proceed to gather yes-no annotations regarding the relevance of each map to the prediction task. We denote the set of indices corresponding to training data with relevant and irrelevant maps as R and I, respectively.

To refine the model using the yes-no annotations, we introduce a loss function based on the energy loss [578]. For the data point $\mathbf{x}_n$ whose saliency map $M_{\mathbf{x}_n}$ highlights the relevant regions (i.e., $n \in R$), the model should generate similar saliency maps following the refinement. Hence, we formulate the loss function $\mathcal{L}_{rel,n}$ as follows:

$$\mathcal{L}_{rel,n} = \sum_{h=1}^{H} \sum_{w=1}^{W} [M'_{\mathbf{x}_n}]_{hw}(1 - [M_{\mathbf{x}_n}]_{hw}) \tag{7.1}$$

where H and W represent the height and width of the saliency maps, respectively, and $M'_{\mathbf{x}_n}$ is the saliency map for the model being trained and the data point $\mathbf{x}_n$. We clarify that $M_{\mathbf{x}_n}$ is the saliency map for the original model before refining, and $M'_{\mathbf{x}_n}$ is for the model being fine-tuned. For better stability of the loss function, we normalize both $M_{\mathbf{x}_n}$ and $M'_{\mathbf{x}_n}$, scaling their values between 0 and 1 by dividing each map by its maximum value.

For the data point $\mathbf{x}_n$ with irrelevant saliency map (i.e., $n \in I$), the model should attend to the regions that are not highlighted in the map $M_{\mathbf{x}_n}$. In this regard, we construct the loss function $\mathcal{L}_{irrel,n}$ as follows:

$$\mathcal{L}_{irrel,n} = \sum_{h=1}^{H} \sum_{w=1}^{W} [M'_{\mathbf{x}_n}]_{hw}[M_{\mathbf{x}_n}]_{hw} \tag{7.2}$$

While guiding the model to attend to the right regions, we need to preserve the accuracy of the model's predictions. Therefore, we incorporate the prediction loss $\mathcal{L}_{pred,n}$ for the data point $\mathbf{x}_n$:

$$\mathcal{L}_{pred,n} = \sum_{k=1}^{K} -y_{nk} \log \hat{y}_{nk} \tag{7.3}$$

where y_{nk} is 1 if the label of the data $\mathbf{x}_n$ is k and 0 otherwise and $\hat{y}_{nk}$ is the probability of the data $\mathbf{x}_n$ being labeled as k computed by the model being trained.

Summing up the loss functions, we obtain the loss $\mathcal{L}_{att}$ that guides a model with yes-no annotations on saliency maps,

$$\mathcal{L}_{att} = \sum_{n=1}^{N} \mathcal{L}_{pred,n} + \frac{\alpha}{|R|} \sum_{n \in R} \mathcal{L}_{rel,n} + \frac{\beta}{|I|} \sum_{n \in I} \mathcal{L}_{irrel,n} \tag{7.4}$$

where α and β are the hyperparameters that control the weights of the loss terms, and $|R|$

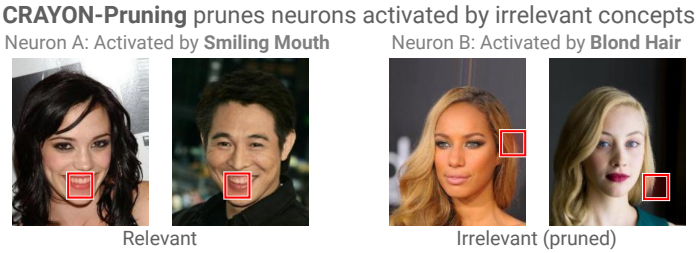


Figure 7.2: CRAYON-PRUNING prunes the neurons activated by irrelevant concepts in the penultimate layer and fine-tunes the last layer. For example, in a smile classifier, **Left:** a neuron is activated by *smiling mouth*, which is relevant to smiling, while **Right:** another neuron is activated by irrelevant *blond hair* and is pruned.

and $|I|$ are the number of relevant and irrelevant training data points respectively.

7.2.3 CRAYON-Pruning: Prune Irrelevant Neurons

Neurons, also referred to as *channels*, in the penultimate layer of CNN models are activated by specific high-level visual concepts in the input data [579]. Based on this finding, a model interpretation method that summarizes the concepts responsible for a neuron's activation as a collection of image patches has recently been proposed [580]. These patches are identified by selecting the images that most highly activate the neuron and cropping out the corresponding region. For example, a neuron in the penultimate layer of a smile classifier has patches corresponding to the *mouth* concept, indicating that the neuron's activation is attributed to the presence of a mouth (Fig. 7.2, left), while another neuron has patches for *hair* (Fig. 7.2, right).

CRAYON-PRUNING identifies the neurons in the penultimate layer that are activated by irrelevant concepts by presenting the image patches of each neuron and collecting yes-no annotations on their relevance. For instance, for the smile classifier in the previous example, the neuron activated by the *mouth* is relevant, while the neuron activated by the *hair* is irrelevant. We prune the irrelevant neurons and fine-tune the last fully-connected layer of the model to remove the effect of the irrelevant concept on the model's prediction. For this fine-tuning process, we use the prediction loss in Eq. 7.3.

7.2.4 Time Complexity

Both CRAYON-ATTENTION and CRAYON-PRUNING share the same general time complexity of $\mathcal{O}(NEM)$, where N is the number of training data points, E is the number of training epochs, and M corresponds to time for loss computation. For CRAYON-ATTENTION, the loss computation is based on Grad-CAM (as described in Sec. 7.2.2), whereas for CRAYON-PRUNING, it uses the conventional cross-entropy loss. This analysis indicates that the runtimes of both CRAYON-ATTENTION and CRAYON-PRUNING scale linearly with the

number of training data points, as we will empirically demonstrate in our experiments in Sec. 7.3.7.

7.3 Evaluation with Human Annotations

To demonstrate CRAYON's effectiveness, scalability, and practicality in rectifying model attention, we collect yes-no human annotations for three benchmark datasets from 5,893 participants on Amazon Mechanical Turk (MTurk).

7.3.1 Benchmark Datasets and Model Training

Waterbirds [569] consists of bird images, where waterbirds and landbirds are more commonly seen in water (e.g., lake) and land (e.g., forest) backgrounds, respectively. Its training set consists of:

- 1,057 *waterbirds* on *water* backgrounds,
- 3,498 *landbirds* on *land* backgrounds,
- 56 *waterbirds* on *land* backgrounds, and
- 184 *landbirds* on *water* backgrounds.

Bird classifiers trained on this data tend to classify birds based on the backgrounds rather than their bird features. The test set consists of 1,284 waterbirds and 4,510 landbirds; half of the waterbird images and half of the landbird images have water backgrounds, while the other half have land backgrounds.

Biased CelebA [581] demonstrates a correlation where most training images of individuals with *black hair* exhibit *smiling* expressions, while those with *blond hair* are mostly *not smiling*. There are a total of 20,200 training instances:

- 10,000 with *black hair* and *smiling* attributes,
- 10,000 with *blond hair* and *not smiling* attributes,
- 100 with *black hair* and *not smiling* attributes, and
- 100 with *blond hair* and *smiling* attributes.

This correlation leads the smile classifier trained on this dataset to make incorrect associations between smile predictions and hair color. The test set contains a total of 8,000 data instances, with 2,000 instances per group.

Backgrounds Challenge [64] addresses the problem that classifiers trained on the ImageNet [582] dataset often inappropriately base their predictions on image backgrounds, rather than foreground objects. Aiming to correct models to base their predictions on the foreground objects, the challenge introduces the ImageNet-9 (IN-9) dataset, a subset of

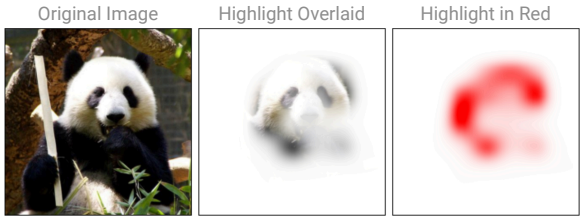


Figure 7.3: Example visualization shown to participants for attention annotations.

ImageNet with nine coarse-grained classes (e.g., dog, bird, vehicle). The IN-9 dataset's training and test set consist of 5,045 and 450 images for each class, respectively (45,405 and 4,050 in total).

For each dataset, we train a ResNet50 [583] classifier, pretrained with ImageNet [582], with the cross-entropy loss (Eq. 7.3) for 10 epochs, and refer to it as the *Original* model throughout this paper, indicating no refinement has been applied and thus it may attend to irrelevant areas. We train the *Original* model for the Waterbirds dataset with Adam optimizer [584] with a learning rate of 0.001, a weight decay of 0.0001, and a batch size of 256 for 50 epochs. Images are resized to 256×256, center-cropped to 224×224, and then normalized. For the Biased CelebA, we use Adam optimizer with a learning rate of 0.001, a weight decay of 0.0001, and a batch size of 128 for 10 epochs. Images are resized to 274×224. For the Backgrounds Challenge, we use the model provided in the challenge repository as the Original model. Images are resized to 256×256, center cropped to 224×224, and then normalized.

7.3.2 Annotating Visualized Interpretations

We collect yes-no annotations for the relevance of saliency maps and neuron concepts on Amazon MTurk. To collect human annotations for CRAYON-ATTENTION, we present participants with a visualization for each training image, as the example shown in Fig. 7.3, consisting of:

- *Original* training image
- *Highlight Overlaid* depicting regions receiving higher Grad-CAM attention with higher opacity
- *Highlight in Red* coloring model-attended areas in red

This visualization design was informed by our pilot study, where we learned that relying solely on *Highlight in Red*, which has conventionally been used [585], could obscure image contents, making it hard for participants to assess relevance. Similarly, identifying model-attended areas solely with *Highlight Overlaid* is challenging when the attended areas are white in color. We present all three visualization components for Waterbirds and

Backgrounds Challenge, while only *Highlight Overlaid* is presented for Biased CelebA, as most areas in these face images are in skin colors, and many participants have found it sufficient to determine relevance. To maximize quality and efficiency, we present each visualization to two participants [586] and ask each participant a single yes-no question:

- Waterbirds: Is the strong highlight mainly on the bird?
- Biased CelebA: Can you determine if the person in the image is smiling?
- Backgrounds Challenge[1]: Is the strong highlight mainly on X? (X is the image's class label, e.g., dog, wheeled vehicle, fish)

We carefully set our questions to include the word "mainly" for the Waterbirds and Backgrounds Challenge datasets so that the participants would pick "*yes*" only if they strongly agree with the question, rather than just checking for the existence of highlight. Images that receive "*yes*" responses (e.g., highlight only on the bird for Waterbirds) from both participants are annotated as having a *relevant* saliency map. Conversely, if both respond with "*no*," the image is annotated as *irrelevant*. Images receiving mixed responses, suggesting possible Grad-CAM ambiguity, are not used to guide the model's attention.

For CRAYON-PRUNING, we generate image patches that represent the visual concepts of each neuron in the model's penultimate layer. Since a neuron's concepts are often determined by a combination of multiple patches [580], we generate three patches for each neuron, following convention [587]. These patches are created by identifying the top three data points that yield the highest activation for the neuron, along with the activation's spatial position. As an image patch could strongly activate multiple neurons, we eliminate such duplicates, resulting in a set of 2,060 unique patches for the Waterbirds dataset, 4,627 for the Biased CelebA dataset, and 3,042 for the Backgrounds Challenge dataset. We visualize each patch by overlaying a red rectangle on the corresponding position of the image (Fig. 7.2). Each patch is shown to an individual participant, and we inquire whether the red highlight is on the bird body for the Waterbirds dataset and whether the area covered by the patch implies the presence of a smile for the Biased CelebA dataset. For the Backgrounds Challenge dataset, we ask whether the red highlight is on X, where X is the image's class label. A neuron is annotated as *relevant* if the majority of its patches receive *yes* responses, and as *irrelevant* if the majority receive *no* responses.

7.3.3 CRAYON Configurations

We outline the hyperparameter configurations for CRAYON used in our experiments. For the Waterbirds dataset, CRAYON-ATTENTION fine-tunes the *Original* bird classifier for 10

[1]As the Backgrounds Challenge is a large dataset with 45,045 training images, we collected annotations incrementally and discovered that annotating a small random subset of just 10% (4,500) was already sufficient to achieve SOTA results, as presented in Sec. 7.3.6.

epochs with a batch size of 128, using the Adam optimizer [584] with a learning rate of 5e-5 and a weight decay of 1e-4. We set the hyperparameters α and β in Eq. 7.4 set to 1e+7 and 2e+5, respectively. For Biased CelebA, the *Original* smile classifier is fine-tuned for 10 epochs with a batch size of 64, using the Adam optimizer with a learning rate of 1e-5 and a weight decay of 1e-4. We set the hyperparameters α to 5e+7 and β to 1e+6. For Backgrounds Challenge, we fine-tune the classifier for 10 epochs with a batch size of 256 using the SGD optimizer [588] with a learning rate of 5e-6 and a weight decay of 1e-1. The hyperparameters α and β set to 5e+3 and 5e+2, respectively.

For CRAYON-PRUNING, we prune 1,034 irrelevant neurons from the penultimate layer of the *Original* model trained on the Waterbirds dataset, and then fine-tunes the last fully connected layer for 10 epochs with a learning rate of 5e-5. For the Biased CelebA dataset, we prune 1,871 irrelevant neurons and train the last layer for 50 epochs with a learning rate of 5e-6. For the Backgrounds Challenge dataset, 407 neurons are pruned, and the last layer is trained for 10 epochs with a learning rate of 1e-6.

We also evaluate CRAYON-PRUNING+ATTENTION, which prunes irrelevant neurons in the penultimate layer and then fine-tunes the entire model with attention guidance[2]. For Waterbirds, we set α to 1e+7 and β to 2e+5, while for Biased CelebA, we set α and β to 5e+7 and 1e+6, respectively. For Backgrounds Challenge, we set α to 1e+3 and β to 5e+1. For the Waterbirds and Biased CelebA datasets, we use the Adam optimizer with the same learning rate, weight decay, and number of epochs as with CRAYON-ATTENTION, while we use the SGD optimizer with a learning rate of 5e-5 and weight decay to 1e-1 for the Backgrounds Challenge dataset. We select these values base on the performance of training data after testing with a wide range of hyperparameter values (details in Sec. 7.3.11).

7.3.4 Compared Methods

To demonstrate the importance of human annotations, we compare CRAYON with some of the most recent, best-performing methods aimed at rectifying model attention without human annotations:

- *JtT* [51] upweights the loss of training data points misclassified by the *Original* model.
- *MaskTune* [65] guides the model to learn diverse features by masking regions highly attended by the *Original* model.
- *LfF* [50] introduces an auxiliary model that heavily relies on irrelevant areas to identify the data points that are likely to receive irrelevant attention.
- *SoftCon* [589] uses the auxiliary model to closely locate data points with the same class labels but significant disparities in irrelevant areas.

[2]Neuron pruning must be applied before using attention annotations, as annotated image patches stem from *Original* model. Fine-tuning the model with CRAYON-ATTENTION may change the concepts detected by each neuron and new image patches may appear.

Table 7.1: CRAYON achieves the **best**, state-of-the-art performance in correcting model attention, outperforming all existing methods across 3 benchmark datasets, even surpassing methods that require more complex annotations.

Method	Annotation	Waterbirds		Biased CelebA		Backgrounds	
		WGA↑	MGA↑	WGA↑	MGA↑	MR↑	BG-Gap↓
Original	-	28.35	72.08	32.60	73.71	78.27	12.99
CRAYON-ATTENTION	Yes-No	72.31$_{\pm 0.89}$	85.23$_{\pm 0.17}$	83.29$_{\pm 2.07}$	89.61$_{\pm 0.10}$	80.85$_{\pm 0.23}$	8.52$_{\pm 0.29}$
CRAYON-PRUNING	Yes-No	68.97$_{\pm 0.43}$	83.13$_{\pm 0.08}$	69.51$_{\pm 2.20}$	86.75$_{\pm 0.40}$	78.61$_{\pm 0.15}$	12.18$_{\pm 0.11}$
CRAYON-PRUNING+ATTENTION	Yes-No	**76.04**$_{\pm 3.38}$	**86.03**$_{\pm 0.45}$	**84.38**$_{\pm 1.49}$	**90.13**$_{\pm 0.36}$	**81.66**$_{\pm 0.14}$	**8.40**$_{\pm 0.31}$
JtT [51]	-	46.88$_{\pm 1.69}$	78.29$_{\pm 0.92}$	35.25$_{\pm 1.90}$	74.29$_{\pm 0.31}$	77.61$_{\pm 0.20}$	12.99$_{\pm 0.09}$
MaskTune [65]	-	45.67$_{\pm 2.17}$	79.13$_{\pm 0.36}$	37.72$_{\pm 1.37}$	78.85$_{\pm 0.28}$	78.14$_{\pm 0.08}$	12.54$_{\pm 0.07}$
LfF [50]	-	44.64$_{\pm 0.85}$	77.24$_{\pm 0.27}$	44.35$_{\pm 1.76}$	77.05$_{\pm 0.40}$	78.23$_{\pm 0.05}$	12.41$_{\pm 0.08}$
SoftCon [589]	-	46.10$_{\pm 4.56}$	79.93$_{\pm 1.28}$	42.38$_{\pm 13.99}$	76.17$_{\pm 3.23}$	73.14$_{\pm 2.02}$	11.58$_{\pm 0.75}$
FLAC [577]	-	40.68$_{\pm 10.58}$	80.77$_{\pm 3.59}$	39.31$_{\pm 11.84}$	76.73$_{\pm 3.51}$	79.91$_{\pm 0.36}$	9.75$_{\pm 0.35}$
LC [66]	-	61.65$_{\pm 3.70}$	80.43$_{\pm 0.67}$	64.98$_{\pm 5.09}$	84.80$_{\pm 0.88}$	74.78$_{\pm 1.81}$	13.35$_{\pm 0.55}$
CnC [67]	-	46.98$_{\pm 1.12}$	77.80$_{\pm 0.47}$	37.76$_{\pm 0.65}$	75.34$_{\pm 0.19}$	77.87$_{\pm 0.42}$	12.91$_{\pm 0.27}$
RRR [73]	Map	53.96$_{\pm 3.46}$	82.29$_{\pm 0.83}$	42.16$_{\pm 4.22}$	78.64$_{\pm 0.82}$	80.08$_{\pm 0.25}$	10.68$_{\pm 0.22}$
GradMask [590]	Map	58.38$_{\pm 3.83}$	82.78$_{\pm 0.66}$	8.43$_{\pm 6.70}$	65.72$_{\pm 2.45}$	80.11$_{\pm 0.30}$	10.63$_{\pm 0.30}$
ActDiff [591]	Map	64.58$_{\pm 1.66}$	84.54$_{\pm 0.23}$	41.29$_{\pm 4.69}$	79.08$_{\pm 1.20}$	75.84$_{\pm 5.97}$	11.73$_{\pm 3.06}$
GradIA [74]	Yes-No, Map	60.87$_{\pm 2.86}$	83.17$_{\pm 0.51}$	41.80$_{\pm 7.82}$	76.58$_{\pm 2.09}$	79.79$_{\pm 0.19}$	11.26$_{\pm 0.15}$
Bounding Box [81]	Bounding Box	66.36$_{\pm 2.17}$	85.85$_{\pm 0.46}$	33.34$_{\pm 1.96}$	73.03$_{\pm 0.38}$	80.82$_{\pm 0.13}$	9.81$_{\pm 0.96}$
ERM	-	22.49$_{\pm 12.00}$	70.23$_{\pm 5.10}$	37.82$_{\pm 9.54}$	71.86$_{\pm 3.46}$	77.86$_{\pm 0.26}$	12.88$_{\pm 0.21}$

- *FLAC* [577] maximizes the dissimilarity between the feature distributions of the *Original* model and those of the auxiliary model.
- *LC* [66] corrects logits of the *Original* model using the auxiliary model outputs.
- *CnC* [67] replaces the auxiliary model's predictions required by SoftCon with the *Original* model's predictions.

We also assess five methods that require complex human annotations by using the *ground truth* maps provided in the dataset. We note that this comparison poses CRAYON disadvantages since it uses *simple human-provided* annotations, not *complex ground truth*. For Biased CelebA, which lacks segmentation maps, we generate maps covering the eyes and mouth, relevant to smiling, as the images in this dataset are aligned with respect to the eyes' locations [592]. Bounding boxes are generated to enclose these maps. It is important to note that we could not collect human annotations for the existing methods because they either require proprietary apparatus [75, 74] or are not evaluated with human annotators [73, 81]. The methods[3] are:

- *RRR* [73] collects ground truth maps that annotate irrelevant pixels in the images and guides the model not to attend to these irrelevant pixels.

[3]We tried RES [75] on our datasets and determined that it was computationally prohibitive. The algorithm did not finish one iteration even after 3 hours on an NVIDIA A6000 GPU; 2,205 iterations are needed for the Biased CelebA dataset.

- *GradMask* [590] collects ground truth maps and penalizes attention outside the relevant pixels.
- *ActDiff* [591] collects ground truth maps to mask the background of each image and closely aligns the masked image's representation with the unmasked one.
- *GRADIA* [74] collects yes-no annotations to identify images where the original model generates irrelevant saliency maps or incorrect predictions, collects ground truth maps, and aligns the model's attention accordingly.
- *Bounding Box* [81] collects bounding boxes that cover the relevant regions of each image and guides the model to keep its attention within these boxes.

To ensure attention correction is not solely due to extended training, we examine the naive *empirical risk minimization (ERM)* [569] approach that simply minimizes classification loss by training the model for more epochs. Hyperparameter values for all compared methods are determined through a comprehensive systematic search around the values reported in their papers. We use the values that yield the highest performance. We report each method's training configurations in our repository[4].

7.3.5 Evaluation Metrics

As the Waterbirds and Biased CelebA provide the labels for irrelevant areas, we employ *worst* and *mean group accuracy* as the evaluation metrics in accordance with the practice in literature [569]. Specifically, we first evaluate the model accuracy for each group created by intersecting irrelevant area labels and class labels; for example, Waterbirds consists of four groups: waterbirds on water backgrounds, waterbirds on land backgrounds, landbirds on water backgrounds, and landbirds on land backgrounds. We then compute the minimum and mean accuracy values across these groups and denote them as *worst group accuracy* (WGA) and *mean group accuracy* (MGA), respectively.

For Backgrounds Challenge, we employ the metrics proposed by the challenge itself [64], as it is difficult to categorize its image backgrounds to form image groups. These metrics are based on two datasets, *Mixed-Same* and *Mixed-Rand*, which are created by transforming image backgrounds. The *Mixed-Same* dataset shuffles backgrounds across the images with the same class label, while the *Mixed-Rand* dataset shuffles the backgrounds across all images to decorrelate backgrounds and class labels. We evaluate:

- Accuracy on the Mixed-Rand dataset to examine whether the model makes correct predictions when the class label is decorrelated from background.
- Difference between the accuracies on the Mixed-Same and Mixed-Rand datasets to assess the impact of backgrounds on the model predictions; we name it as *BG-Gap*. A smaller BG-Gap implies less reliance on backgrounds.

[4]https://github.com/poloclub/crayon/blob/main/misc/training_config.md

7.3.6 Results: CRAYON is Effective and SOTA

CRAYON achieves state-of-the-art performance in correcting model attention, outperforming 12 existing methods across three benchmark datasets, including those that require more complex annotations. We run each method five times with different random seeds and report the average and standard deviation of the performance values in Table 7.1.

For Waterbirds, compared to the unrefined original model (Row 1), CRAYON-ATTENTION (Row 2) substantially enhances WGA by 43.96 percentage points (pp), raising it from 28.35% to 72.31%; and MGA by 13.15pp, from 72.08% to 85.23%. CRAYON-PRUNING (Row 3) also demonstrates improvements, achieving a 40.62pp increase in WGA and an 11.05pp increase in MGA. CRAYON-PRUNING+ATTENTION (Row 4), which combines both *attention* and *pruning* approaches, further elevates both WGA and MGA beyond the capabilities of each individual approach, raising WGA to 76.04% and MGA to 86.03%.

CRAYON is also effective for Biased CelebA, significantly outperforming all 12 methods. which is expected for the methods that do not leverage human annotations (Table 7.1, Row 4-10). JtT and LfF, which upweight the training data points misclassified by the original model, exhibit marginal effects as there are few misclassified training data points due to the severe imbalance in training data. Similarly, CnC, which uses misclassified data as positive and negative samples for contrastive learning, achieves only limited improvement. For the methods that use human annotations, their performances were on par with those without human annotations. We believe this is due to the discrepancies between real-valued model-generated maps and binary ground truth maps [75]. Moreover, we observe that GradMask (Row 13) often fails because its masking significantly alters the face areas other than mouth and eyes, making it hard for the model to detect the image as a face. Bounding Box (Row 16) shows limited performance, as rectangular boxes surrounding eyes and mouth inevitably include irrelevant areas like nose and cheeks, where hair frequently appears.

For Backgrounds Challenge, CRAYON achieves the highest accuracy on the Mixed-Rand dataset and smallest BG-Gap, demonstrating its effectiveness in multi-label classification tasks. It improves the accuracy on the Mixed-Rand dataset from 78.27% to 81.66% and reduces the BG-Gap from 12.99% to 8.40%. While CRAYON-PRUNING exhibits marginal performance, since a concept irrelevant to one class can be relevant to another in multi-label classification, combining it with attention guidance complements such limitation and effectively mitigates the model reliance on image backgrounds. The underperformance of most approaches that do not use any annotations demonstrates their limitations in the challenging setting where irrelevant attributes correlated with class labels are unclear [66]. Specifically, the underperformance of JtT and CnC is attributed to the small number (only 2%) of training data points misclassified by the original model, while MaskTune's limited performance suggests its potential reliance on large training data, as higher performance was reported when four times the training data points were used [65].

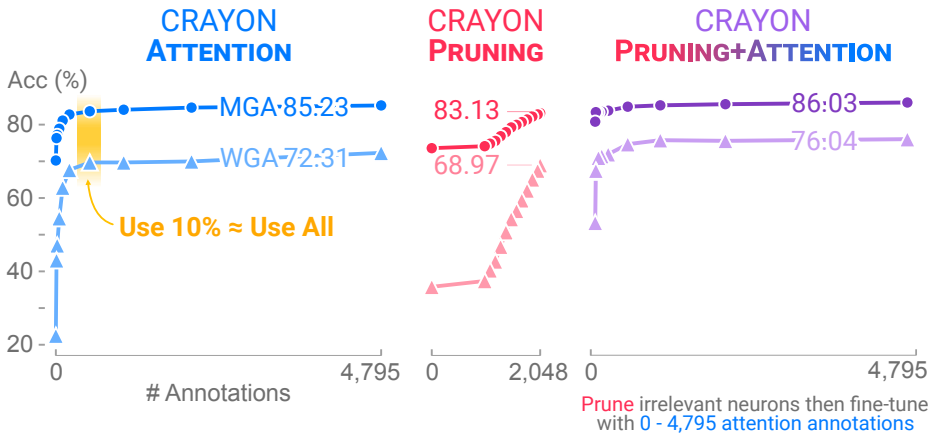


Figure 7.4: For the Waterbirds dataset, even with annotations for just 10% of the training data points, both CRAYON-ATTENTION and CRAYON-PRUNING+ATTENTION nearly reach their peak performance. CRAYON-PRUNING realizes its full effectiveness when annotations are provided for most neurons in the penultimate layer. The performance of CRAYON-PRUNING+ATTENTION with no attention annotations differs from CRAYON-PRUNING's peak performance, as the latter retrains the entire model, while the former fine-tunes only the last layer. For each annotation number n, we run each method five times with different random seeds and report the average MGA and WGA values.

It is noteworthy that CRAYON-ATTENTION surpasses all methods using more complex annotations, which often contain richer information. We attribute CRAYON's superior performance to its ability to overcome the shortcomings of binary ground truth maps and boxes, which are represented as 0s and 1s, while model-generated saliency maps consist of continuous real numbers. This inconsistency degrades the performance of model attention guidance [75]. Additionally, experiments with the Biased CelebA dataset reveal that the shape constraint of bounding boxes significantly degrades performance. CRAYON addresses this challenge by using the saliency maps of the original model instead of binary ground truth.

7.3.7 Scalability: Runtime Scales Linearly with Data Points

We run all CRAYON algorithms on an NVIDIA A6000 GPU with 40GB RAM. For the Waterbirds dataset with 4,795 training data points, CRAYON-ATTENTION takes 288 seconds on average, CRAYON-PRUNING takes 111 seconds, and CRAYON-PRUNING+ATTENTION takes 295 seconds. For the Biased CelebA dataset with 20,200 training data points, CRAYON-ATTENTION takes 1368 seconds, CRAYON-PRUNING takes 1241 seconds, and CRAYON-PRUNING+ATTENTION takes 1401 seconds. For the Backgrounds Challenge dataset, which consists of 45,405 training data points, CRAYON-ATTENTION takes 4512 seconds, CRAYON-PRUNING takes 4118 seconds, and CRAYON-PRUNING+ATTENTION takes 4540 seconds. Overall, CRAYON-PRUNING takes much less time as it fine-tunes only the last fully con-

Table 7.2: Ablation study demonstrates that $\mathcal{L}_{rel}$ guides model attention significantly stronger than $\mathcal{L}_{irrel}$ which directs attention away from irrelevant image regions.

$\mathcal{L}_{rel}$	$\mathcal{L}_{irrel}$	**Waterbirds**		**Biased CelebA**		**Backgrounds**	
		WGA↑	MGA↑	WGA↑	MGA↑	MR↑	BG-Gap↓
✓	✓	72.31	85.23	83.29	89.61	80.85	8.52
✓		67.38	83.64	77.48	88.28	80.52	8.69
	✓	46.57	76.33	67.32	84.59	78.66	11.86
		22.49	70.23	37.82	71.86	77.86	12.88

nected layer. Additionally, these runtimes align with our time complexity analysis of CRAYON that it scales linearly with the number of training data points (Sec. 7.2.4).

7.3.8 Varying Number of Annotations

We evaluate how the number of annotations n affects CRAYON's performance. For CRAYON-ATTENTION, we randomly sample n images from the training set of the Waterbirds dataset and compute both $\mathcal{L}_{rel}$ and $\mathcal{L}_{irrel}$ for these n images along with their annotations. For CRAYON-PRUNING, we randomly sample n neurons from the penultimate layer and prune the *irrelevant* neurons within this sampled group of n. We additionally investigate CRAYON-PRUNING+ATTENTION, where we use all 2,048 annotations for neuron relevance while varying the number of annotations for model attention[5].

Fig. 7.4 presents the results for Waterbirds. The performance of all CRAYON methods improves as the number of annotations increases. Notably, CRAYON-ATTENTION and CRAYON-PRUNING+ATTENTION are effective even with a small number of annotations, achieving nearly peak performance when annotations are available for only 10% of the training data points. Specifically, with attention annotations for 500 out of 4,795 training data points, CRAYON-ATTENTION enhances the WGA and MGA to 83.64% and 69.69%, respectively, while CRAYON-PRUNING+ATTENTION enhances the WGA and MGA to 84.89% and 74.67%, respectively. These values are only marginally lower than the performance achieved with annotations for all training data points. In contrast, the performance of CRAYON-PRUNING is constrained unless a substantial portion of neurons is annotated, underscoring the importance of acquiring annotations for all neurons in the penultimate layer. In our repository, we include additional results for Biased CelebA and Backgrounds Challenge, which show similar overall trends as presented above for Waterbirds[6].

[5]We elect to focus on varying the number of attention annotations based on our observation that almost all pruning annotations need to be used for CRAYON-PRUNING to be fully effective.

[6]https://github.com/poloclub/crayon/blob/main/README.md

Table 7.3: The attention annotations for the *Original* ResNet50 model effectively refine ConvNeXt trained on 3 datasets.

	Waterbirds		**Biased CelebA**		**Backgrounds**	
	WGA↑	MGA↑	WGA↑	MGA↑	MR↑	BG-Gap↓
Original	66.23	84.73	26.76	71.32	87.18	5.87
Refined	76.73	88.59	49.50	79.40	89.49	5.56

7.3.9 Ablation Study

We conduct an ablation study to assess the impact of the two proposed loss terms, $\mathcal{L}_{rel}$ and $\mathcal{L}_{irrel}$, on CRAYON-ATTENTION's performance. We deactivate one of the two loss terms by setting either α or β in Equation 7.4 to 0. Table 7.2 summarizes the results. Overall, removing either loss term degrades the performance of CRAYON-ATTENTION, highlighting the contributions of both $\mathcal{L}_{rel}$ and $\mathcal{L}_{irrel}$ in guiding model attention. When we deactivate $\mathcal{L}_{irrel}$ and rely solely on $\mathcal{L}_{rel}$ (Row 2), WGA and MGA experience declines of 4.93pp (72.31% to 67.38%) and 1.59pp (85.23% to 83.64%) for Waterbirds and 5.81pp (83.29% to 77.48%) and 1.33pp (89.61% to 88.28%) for Biased CelebA. For Backgrounds Challenge, the accuracies on the Mixed-Rand datasets decrease from 80.85% to 80.52% and the BG-Gap increases from 8.52% to 8.69%, respectively.

Excluding $\mathcal{L}_{rel}$ from $\mathcal{L}_{att}$ (Row 3) also significantly impairs the performance. For Waterbirds, WGA decreases by 25.74pp (72.31% to 46.57%) and MGA by 8.90pp (85.23% to 76.33%), and for Biased CelebA, WGA declines by 15.97pp (83.29% to 67.32%) and MGA by 5.02pp (89.61% to 84.59%). Likewise, for Backgrounds Challenge, the accuracy on the Mixed-Rand dataset decreases by 2.19pp (80.85% to 78.66%) and the BG-Gap increases by 3.34pp (8.52% to 11.86%). These results show that $\mathcal{L}_{rel}$ plays a significant role in guiding model attention, while $\mathcal{L}_{irrel}$ provides additional guidance by directing attention away from irrelevant areas.

7.3.10 Transferability of Attention Annotations

Although attention annotations have been collected with respect to the saliency maps from the *Original* model, CRAYON-ATTENTION can effectively exploit these annotated saliency maps to refine other models, as these annotated maps serve as indicators of where a model should or should not attend. To assess the transferability of attention annotations, for each dataset, we train five ConvNeXt [593] models, each initialized with a different random seed value. We then refine these models using CRAYON-ATTENTION with the saliency maps from the *Original* model and their relevance annotations. In other words, we train the ConvNeXt models using the saliency maps from the *Original* model as $M_{\mathbf{x}_n}$ in Eq. 7.1 and 7.2, and their relevance annotations to decide R and I in Eq. 7.4. We note that $M'_{\mathbf{x}_n}$ in

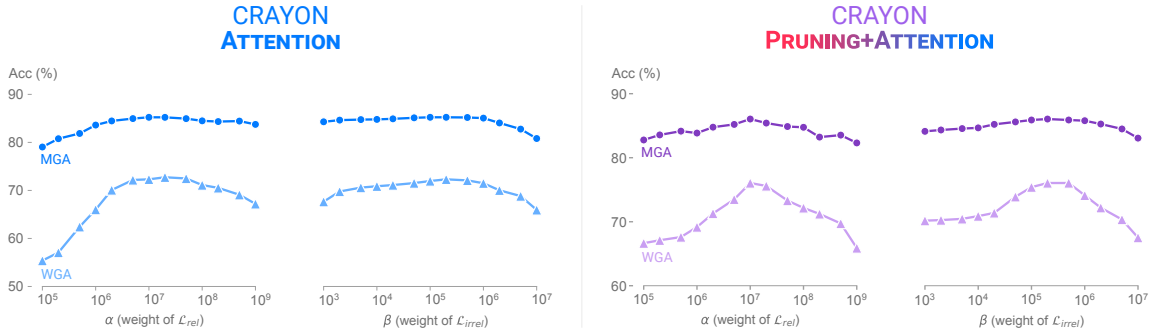


Figure 7.5: For the Waterbirds dataset, both CRAYON-ATTENTION (left) and CRAYON-PRUNING+ATTENTION (right) effectively correct model attention for wide range of α and β.

Eq. 7.1 and 7.2 is the saliency map for the ConvNeXt models being trained. We run the CRAYON-ATTENTION for each model five times, each with a different random seed, and report the average performance in Table 7.3.

The attention annotations for the *Original* model's saliency maps effectively guide the attention of the other models. For the Waterbirds dataset, the average WGA of the five models, which was 66.23% before refining, increases to 76.73%, and the average MGA, which was 84.73%, is elevated to 88.59%. Similarly, for the Biased CelebA dataset, the average WGA improves from 26.76% to 49.50%, and MGA from 71.32% to 79.40%. For the Backgrounds dataset, the average accuracy on the Mixed-Rand dataset is enhanced from 87.18% to 89.49%, and the BG-Gap is reduced from 5.87% to 5.56%. These results demonstrate that our attention annotations can be effectively transferred across different models.

7.3.11 Hyperparameter Sensitivity

We examine how CRAYON's two hyperparameters, α and β, affect the performance of CRAYON-ATTENTION and CRAYON-PRUNING+ATTENTION. We evaluate the performance of the two methods on the Waterbirds dataset with different values of α over the wide range from 1e+5 to 1e+9 while consistently using β of 2e+5 as in Sec. 7.3.3. Similarly, we vary β from 1e+3 to 1e+7, while keeping α as 1e+7. Fig. 7.5 visualizes the results.

We observe that the performance of CRAYON-ATTENTION stabilizes for various values of α from 5e+6 to 5e+7. The performance is even more stable with different values of β ranging from 2e+3 to 5e+5. The performance of CRAYON-PRUNING+ATTENTION is relatively more sensitive to these hyperparameters; changing α by a scale of 2 can impact the performance, while the performance stabilizes for β from 1e+5 to 5e+5. This demonstrates that both CRAYON-ATTENTION and CRAYON-PRUNING+ATTENTION achieve its effectiveness with various values of the hyperparameters, while they would benefit from a careful hyperparameter tuning. Our repository provides the results of the hyperparameter

sensitivity experiments for the Biased CelebA and Backgrounds Challenge dataset, which show similar trends to the Waterbirds dataset[7].

7.4 Conclusion

We introduce CRAYON, the state-of-the-art approach for correcting model attention using simple yes-no annotations annotations. Extensive quantitative evaluation with large-scale human annotations demonstrates CRAYON's effectiveness, scalability, and practicality. It outperforms all existing methods across three benchmark datasets, surpassing approaches that rely on more complex annotations.

[7] https://github.com/poloclub/crayon/blob/main/README.md

CHAPTER 8
PROBING LLM HALLUCINATION FROM WITHIN: PERTURBATION-DRIVEN APPROACH VIA INTERNAL KNOWLEDGE

8.1 Introduction

Large language models (LLMs) excel at text generation [594, 595] but often produce hallucinations — incorrect or unverifiable content — posing significant risks for practical use [596]. Detecting such hallucinations is crucial yet challenging due to their plausible appearance [597, 598] and diverse types [490, 597]. One common detection approach involves comparing outputs with external sources (e.g., Wikipedia) [18, 19, 20], but they fail when such sources are unavailable. Alternative approaches include fine-tuning LLMs to reject questions likely to produce hallucinations [24, 25, 26] or training classifiers to detect them [22, 21, 23], both requiring expensive supervised training on large labeled datasets. To address these limitations, some recent efforts forgo external knowledge or supervised training. These approaches include checking consistency across multiple generations [30], prompting LLMs to assess correctness [27, 29, 599], or estimating LLM's uncertainty about its generation [36, 40, 600, 33, 31].

However, the methods above have a key limitation: they do not assess whether an LLM possesses sufficient knowledge to generate accurate responses — an essential factor in ensuring accurate detection. To be specific, consistency checks cannot identify hallucinations that the model repeatedly generates [330], while prompting-based approaches may result in random guesses when the LLM lacks knowledge. Moreover, uncertainty estimation methods assume all hallucinations stem from uncertainty, neglecting factors such as randomness in token sampling or beam search variability [45, 46].

These limitations highlight the critical need to assess the LLM's knowledge and its alignment with generated content. We address this gap with the following key contributions:

- **A novel hallucination probing task** that classifies LLM-generated text into three categories — *aligned* (LLM has sufficient knowledge and responds faithfully), *misaligned* (LLM has knowledge but generates contradictions), and *fabricated* (LLM lacks relevant knowledge). Our categorization refines hallucination types beyond existing work [601, 490] (Sec. 8.2), significantly boosting detection performance (Fig. 8.1).
- **Entity Perturbation Impact discovery**, our key finding revealing that perturbing key entities in prompts affects LLM outputs differently depending on the type of hallucination present (Sec. 8.2). While prior studies have identified that hallucinations arise from errors while retrieving knowledge about the key entities [602], they did not leverage this for hallucination detection. We systematically perturb key entities to assess their impact

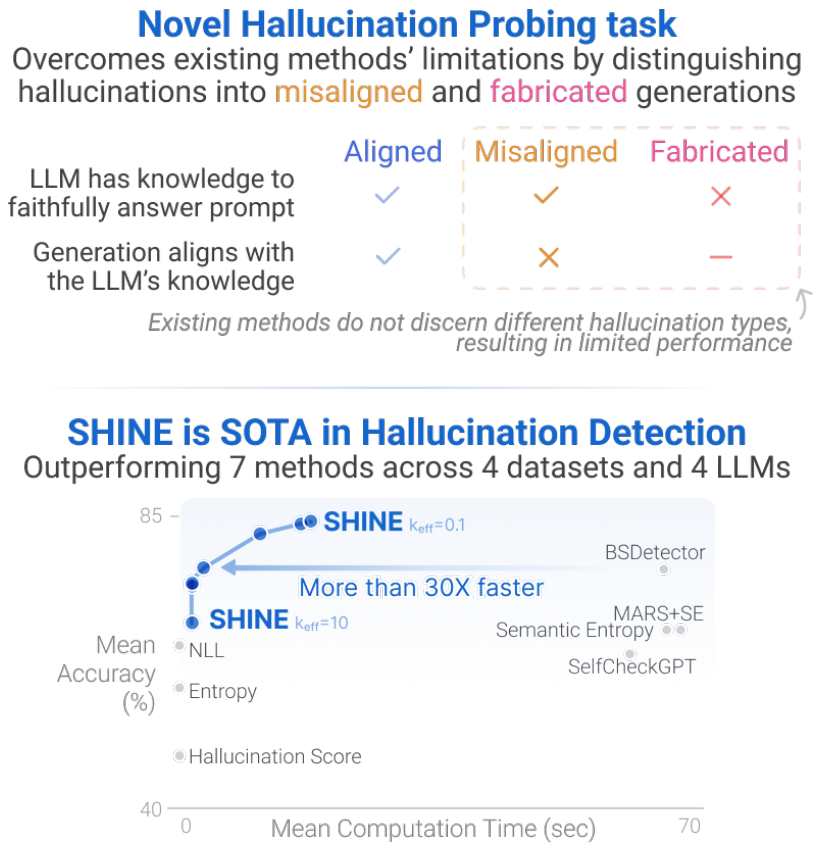


Figure 8.1: We introduce a novel **hallucination probing** task that distinguishes hallucination into *misaligned* and *fabricated* text. SHINE, our novel hallucination probing method, achieves **best** performance in hallucination detection, outperforming 7 methods across 4 datasets and 4 LLMs (Table 8.2, 8.3).

on generated tokens, demonstrating how these perturbations can determine whether the generated tokens stem from the LLM's knowledge and align with it. This provides the foundation for distinguishing between aligned, misaligned, and fabricated responses for hallucination probing.

- **SHINE, a novel hallucination probing method** leveraging our *Entity Perturbation Impact* discovery to predict whether an LLM possesses sufficient knowledge to generate faithful response, and whether the response aligns with the LLM's knowledge (Sec. 8.3, Fig. 8.3). SHINE operates without external knowledge, supervised training, or LLM fine-tuning. SHINE stands for **S**ystematic **H**allucination **I**nspection with **N**oisy **E**ntity.
- **Extensive experiments demonstrating the superiority of our method** on four modern LLMs (LLaMA2-13B-Chat, LLaMA3-8B-Instruct, Mistral-7B-Instruct, Qwen2.5-7B-Instruct) across four datasets for diverse text generation tasks. Our method effectively classifies aligned, misaligned, and fabricated text (Sec. 8.4.1) and outperforms all existing algorithms on hallucination detection (Sec. 8.4.2), underscoring the importance of

hallucination probing.

8.2 Discovery: Entity Perturbation Impact for Hallucination Probing

Inspired by the prior work that explores GPT-2's knowledge by perturbing entities in templated aligned prompts [364], we ask: "*Is LLM-hallucinated text, ungrounded in model knowledge, as sensitive to perturbations as aligned text?*" This drives our novel Entity Perturbation Impact discovery, forming the basis of SHINE, our hallucination probing method (Sec. 8.3).

To explore this, we extend the analysis to modern LLMs, specifically, LLaMA2-13B-Chat-GPTQ [489] (henceforth, LLaMA2), and more flexible prompts that ask factual questions about specific entities, which have been the primary focus of recent research [603, 604, 602, 605]. We use the NEC benchmark [606], which includes prompts about existent and non-existent entities across six topics; real-entity prompts yield aligned or misaligned responses, whereas fictional ones generate fabricated responses.

Below, we outline an LLM's inference process, then introduce our perturbation and evaluation method, which leads to our key discovery that perturbing key entities affects LLM's generation of aligned, misaligned, and fabricated responses differently, and ultimately gives rise to SHINE (Sec. 8.3).

LLM Inference without Perturbation. For a prompt P and text G, a tokenizer with a token set $\mathcal{T}$ splits P into M tokens $t_{1:M} = [t_1, \ldots, t_M]$ and G into N tokens $t_{M+1:M+N} = [t_{M+1}, \ldots, t_{M+N}]$, where $t_i \in \mathcal{T}$. Each token t_i is mapped to a d-dimensional embedding vector $\mathbf{e}_i \in \mathbb{R}^d$ by a token embedding map. An LLM f takes the embedding vector sequence $\mathbf{e}_{1:M+N} = [\mathbf{e}_1, \ldots, \mathbf{e}_{M+N}]$ as an input and computes the probability of each token appearing after each token position i; i.e., $f(\mathbf{e}_{1:M+N}) = \mathbf{P}_{1:M+N} \in \mathbb{R}^{(M+N)\times|\mathcal{T}|}$, where $\mathbf{P}_i \in \mathbb{R}^{|\mathcal{T}|}$ and $\mathbf{P}_i(t)$ is the probability of the token t to be generated at the position $i+1$.

Entity Perturbation. For each prompt P, we manually identify the key entity S, tokenized as $t^S_{1:K}$. We locate all token positions I_S where S appears in P and the LLM's response G, i.e., $I_S = \{i | t_{i:i+K-1} = S\}$. To perturb the embeddings of S, we add Gaussian noise $\boldsymbol{\epsilon} \sim \mathcal{N}(0, (3\boldsymbol{\sigma}_0)^2) \in \mathbb{R}^{K\times d}$ to all occurrences of S, where $\boldsymbol{\sigma}_0$ is the standard deviation of token embeddings in the NEC dataset [364]; i.e., $\hat{\mathbf{e}}_{i:i+K-1} = \mathbf{e}_{i:i+K-1} + \boldsymbol{\epsilon}$ for $i \in I_S$, while leaving other token embeddings unchanged. Then, we input $\hat{\mathbf{e}}_{1:M+N}$ to the LLM f, obtaining a perturbed probability distribution $\hat{\mathbf{P}}_{1:M+N} = f(\hat{\mathbf{e}}_{1:M+N}) \in \mathbb{R}^{(M+N)\times|\mathcal{T}|}$. For each token position i in G, we evaluate the Entity Perturbation Impact using: (1) Kullback-Leibler (KL) divergence between $\mathbf{P}_i$ and $\hat{\mathbf{P}}_i$ (i.e., $KL(\mathbf{P}_i \| \hat{\mathbf{P}}_i)$) to capture global effects across the entire vocabulary and (2) change in the generation probability of token t_i (i.e., $\Delta P_i = \hat{\mathbf{P}}_i(t_i) - \mathbf{P}_i(t_i)$), focusing on generated tokens. To mitigate the impact of random Gaussian noise, we repeat this process 10 times with different random seeds.

We note that our approach differs from the methods that detect hallucinations via

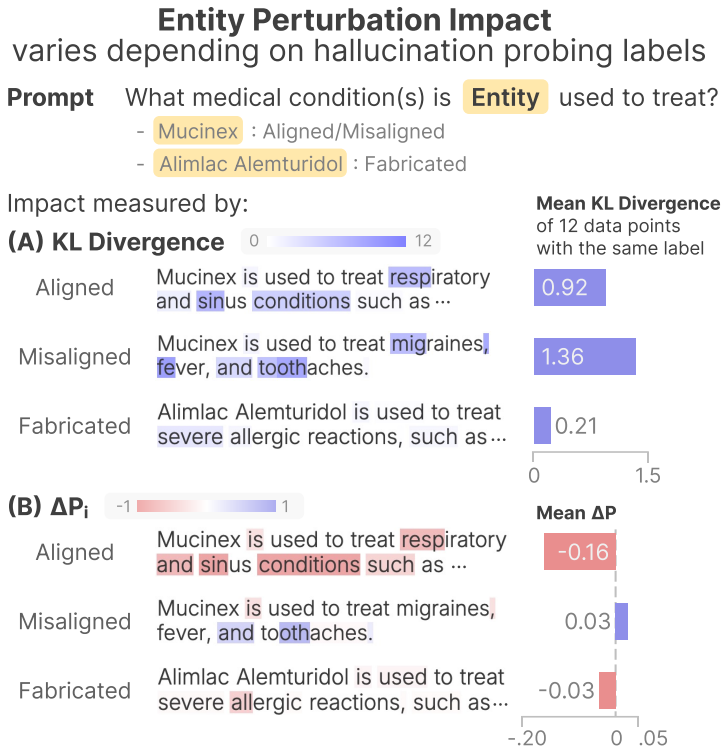


Figure 8.2: Entity Perturbation Impact, measured by KL divergence and ΔP_i, varies across aligned, misaligned, and fabricated text. **(A)** Tokens in aligned and misaligned text, strongly linked to the LLM's knowledge of the entity, have high KL divergence, while most tokens in fabricated text have near-zero values, indicating weak association with the LLM's knowledge. The mean KL divergence of fabricated text is significantly lower than aligned and misaligned text. **(B)** While tokens in aligned and fabricated text generally show negative or near-zero ΔP_i, misaligned text has tokens with positive ΔP_i, particularly those in the contradictory phrase, leading the mean ΔP of misaligned text to surpass that of aligned and fabricated text.

inconsistencies across generations. While their inconsistencies mean that there is a token position i with multiple inconsistent token ts with high generation probability $\mathbf{P}_i(t)$ (i.e., $\mathbf{P}_i$ has high entropy), we measure the KL divergence between original and perturbed distributions $\mathbf{P}_i$ and $\hat{\mathbf{P}}_i$.

Experiments. From the NEC dataset, we randomly sample two prompts for each topic for both existent and non-existent entities, yielding 12 aligned, 12 misaligned, and 12 fabricated data points. This analysis focuses on a small subset, as it requires manual identification of key entities; results for full dataset using our automated approach are in Sec. 8.4. In addition to tokenwise analysis, we evaluate each text's Entity Perturbation Impact by taking mean across all tokens in G except those of the perturbed entity, and refer to the results as the KL divergence and ΔP of G.

Fig. 8.2 demonstrates the Entity Perturbation Impact of aligned, misaligned, and fabri-

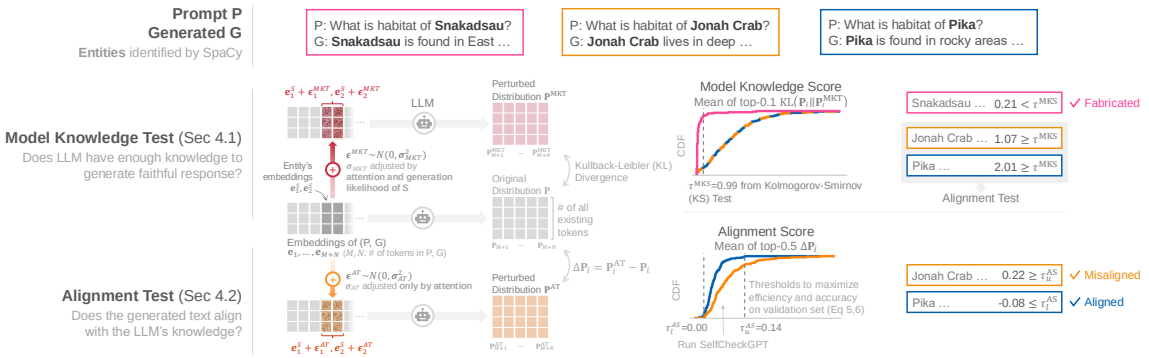


Figure 8.3: SHINE classifies LLM-generated text into aligned, misaligned, and fabricated without external knowledge or supervised training through two steps: Model Knowledge Test (Sec. 8.3.1) and Alignment Test (Sec. 8.3.2). The Model Knowledge Test identifies whether the LLM has sufficient knowledge to generate a faithful response by perturbing entities in the prompt and evaluating the impact; fabricated text is differentiated at this stage. For the data that the LLM has enough knowledge, we conduct the Alignment Test to examine whether the text aligns with the LLM's knowledge.

cated text. Fabricated text shows a much lower mean KL divergence (0.21) than aligned (0.92) and misaligned (1.36) text (Fig. 8.2A). For example, in the aligned response to the prompt "*What medical condition(s) is Mucinex used to treat?*", tokens for "*respiratory*" and "*sinus*", which are relevant to *Mucinex*, have high KL divergence, whereas fabricated text's tokens exhibit near-zero KL divergence, indicating their weak association with the LLM's knowledge. Moreover, misaligned text contains tokens with positive ΔP_i, with a mean ΔP of 0.03 — higher than aligned (-0.16) and fabricated (-0.03) text, where most tokens have negative or near-zero ΔP_i (Fig. 8.2B). In the misaligned text about *Mucinex*, tokens in the misaligned phrase "*fever, and toothaches*" show positive ΔP_i, suggesting $\Delta P_i > 0$ as a misalignment indicator.

Key Takeaways. Low Entity Perturbation Impact measured by KL divergence strongly indicates fabricated text, as its tokens are weakly tied to key entities. In contrast, a rise in generation probability after perturbation (i.e., $\Delta P > 0$) signals misalignment. This is because aligned tokens are grounded in model knowledge of the key entities, so perturbation lowers their probability ($\Delta P < 0$), while tokens in fabricated text remains unaffected ($\Delta P \approx 0$). In Sec. 8.3, we introduce SHINE, a hallucination probing method that leverages Entity Perturbation Impact to distinguish between aligned, misaligned, and fabricated text.

8.3 SHINE: Hallucination Probing Method

Leveraging our Entity Perturbation Impact discovery (Sec. 8.2), we develop a two-stage workflow for hallucination probing, consisting of *Model Knowledge Test* and *Alignment Test* (Fig. 8.3). The Model Knowledge Test first assesses whether the LLM has enough

knowledge to generate faithful response for the prompt and distinguishes fabricated text from other two types (Sec. 8.3.1). Then, the Alignment Test examines whether the generated text aligns with the LLM's knowledge, classifying it as either aligned or misaligned (Sec. 8.3.2).

8.3.1 Model Knowledge Test

Model Knowledge Test is inspired by our discovery that the Entity Perturbation Impact of fabricated text, measured by KL divergence, is much lower than that of aligned or misaligned text. To systematically identify key entities, we detect all entities in a prompt and perturb them with varying strength, determined by the attention they receive. The Model Knowledge Test involves four steps: (1) identifying entities in a prompt, (2) determining each entity's perturbation strength, (3) perturbing the entities' embeddings, and (4) measuring the perturbation's impact.

Step 1. Entity Identification. Model Knowledge Test extracts all entities $S_1, \ldots, S_L$ in P using the named entity (`doc.ents`) and noun chunk (`doc.noun_chunks`) extraction features of SpaCy [607] library.

Step 2. Perturbation Strength Computation. To focus on the entities that significantly contribute to generation, we apply stronger perturbations to entities with higher attention. To calculate the attention $att_l \in \mathbb{R}^+$ of each entity S_l [274], we concatenate M tokens in P and N tokens in G into $t_{1:M+N}$, feed the token sequence into the LLM, and sum the attention values received by the tokens of S_l while generating $t_{M+1:M+N}$. We then scale att_l into $a_l \in [0, 1]$:

$$a_l = \frac{\exp\left(k_{att} \cdot att_l\right)}{\exp\left(k_{att} \cdot \max\left(att_1, \ldots, att_L\right)\right)}, \tag{8.1}$$

where $k_{att} \geq 0$ controls the impact of attention values; $k_{att} = 0$ weighs all entities equally, while larger values emphasize highly attended entities.

Since LLMs tend to fabricate for entities with low frequency in the training data [608, 609, 19, 610], we further adjust perturbation strength based on the likelihood of each entity S_l being generated by the LLM, which serves as a good proxy for frequency [611]. Specifically, we use the negative log-likelihood, scaled by token position to consider later tokens' importance. For an entity S_l, tokenized as $t^{S_l}_{1:K_l}$, the perturbation strength $w_l \in \mathbb{R}^+$ is calculated as

$$w_l = a_l \cdot \left(1 - \frac{1}{K_l} \sum_{k=1}^{K_l} \sqrt{k-1} \log Pr(t^{S_l}_k | t^{S_l}_{1:k-1})\right)^{-1}. \tag{8.2}$$

Step 3. Entity Perturbation. We perturb L entities by adding Gaussian noise to the embeddings of the entities' tokens. For entity $S_l = t^{S_l}_{1:K_l}$, we identify all token positions I_{S_l} where it appears in (P, G): $I_{S_l} = \{i | t_{i:i+K_l-1} = S_l\}$. We add Gaussian noise $\boldsymbol{\epsilon}^{MKT}_l \sim$

$\mathcal{N}(0, (\boldsymbol{\sigma}_l^{MKT})^2) \in \mathbb{R}^{K_l \times d}$ to all $i \in I_{S_l}$, i.e., $\mathbf{e}^{MKT}_{i:i+K_l-1} = \mathbf{e}_{i:i+K_l-1} + \boldsymbol{\epsilon}_l^{MKT}$, while leaving other tokens unchanged; the standard deviation $\boldsymbol{\sigma}_l^{MKT} = w_l \cdot \boldsymbol{\sigma}$, where $\boldsymbol{\sigma}$ is a hyperparameter (details in Sec. 8.4). We input the perturbed embeddings $\mathbf{e}^{MKT}_{1:M+N}$ to the LLM f to compute the perturbed probability distribution $\mathbf{P}^{MKT} = f(\mathbf{e}^{MKT}_{1:M+N}) \in \mathbb{R}^{(M+N)\times|\mathcal{T}|}$.

Step 4. Model Knowledge Score Evaluation. We compute KL divergence between $\mathbf{P}$ and $\mathbf{P}^{MKT}$ at each token position i in the generated text G and refer to it as Model Knowledge Score at the position i, which is denoted as $MKS_i = KL(\mathbf{P}_i \| \mathbf{P}_i^{MKT})$. As the dependency on the previous tokens can reduce the KL divergence values of some tokens, we evaluate MKS, the Model Knowledge Score of G, by taking the mean of top-p^{MKS} values of MKS_i for $i = M+1, \ldots, M+N$ ($0 < p^{MKS} \leq 1$):

$$MKS = \frac{1}{\lceil p^{MKS} \cdot N \rceil} \sum_{i=1}^{\lceil p^{MKS} \cdot N \rceil} MKS_i^{sort}, \tag{8.3}$$

where $MKS_1^{sort} \geq \cdots \geq MKS_N^{sort}$ is a permutation of $MKS_{M+1}, \ldots, MKS_{M+N}$ sorted in descending order. We repeat this 10 times with different random seeds.

If the Model Knowledge Score is lower than a threshold, we classify the text as fabricated; otherwise, we proceed to the Alignment Test (Sec. 8.3.2). We determine the threshold τ^{MKS} using the Kolmogorov-Smirnov (KS) test [612] on the validation set: $\tau^{MKS} = \arg\max_{\tau \in [0,\infty)} (\mathbf{F}_f^{MKS}(\tau) - \mathbf{F}_{a,m}^{MKS}(\tau))$, where $\mathbf{F}_f^{MKS}$ is the cumulative probability of fabricated data's Model Knowledge Score and $\mathbf{F}_{a,m}^{MKS}$ is that of aligned or misaligned data.

8.3.2 Alignment Test

After ensuring the LLM has enough knowledge about (P, G) via Model Knowledge Test, we check if the text G aligns with the LLM's knowledge. Specifically, we add noise $\boldsymbol{\epsilon}_l^{AT} \sim \mathcal{N}(0, (\boldsymbol{\sigma}_l^{AT})^2)$ to the embeddings of each entity S_l to compute the perturbed embeddings $\mathbf{e}^{AT}_{1:M+N}$; since Model Knowledge Test has already verified that the model is knowledgeable about S_l, we do not use the likelihood but adjust $\boldsymbol{\sigma}_l^{AT}$ only with a_l as $\boldsymbol{\sigma}_l^{AT} = a_l \cdot \boldsymbol{\sigma}$, where $\boldsymbol{\sigma}$ is the same hyperparameter used in Model Knowledge Test. We then compute the perturbed probability distribution $\mathbf{P}^{AT} = f(\mathbf{e}^{AT}_{1:M+N}) \in \mathbb{R}^{(M+N)\times|\mathcal{T}|}$ and $\Delta P_i = \mathbf{P}_i^{AT}(t_i) - \mathbf{P}_i(t_i)$ for $i = M+1, \ldots, M+N$

Since some tokens in the misaligned text can still be aligned with model knowledge, while all tokens in the aligned text should be aligned, we evaluate Alignment Score (AS) by taking the mean of top-p^{AS} ΔP_i values ($0 < p^{AS} \leq 1$):

$$AS = \frac{1}{\lceil p^{AS} \cdot N \rceil} \sum_{i=1}^{\lceil p^{AS} \cdot N \rceil} \Delta P_i^{sort}, \tag{8.4}$$

where $\Delta P_1^{sort} \geq \cdots \geq \Delta P_N^{sort}$ is a permutation of $\Delta P_{M+1}, \ldots, \Delta P_{M+N}$ sorted in descending order.

While large negative and large positive AS values are strong indicators of alignment and misalignment, respectively, $AS \approx 0$ can imply either. To handle near-zero AS, we use SelfCheckGPT [30], which takes some computational cost but effectively differentiates text about which a model has enough knowledge. Integrating our Alignment Test with SelfCheckGPT reduces its computational cost by checking only a few data points with AS near zero, while achieving high accuracy. We set lower and upper thresholds, τ_l^{AS} and τ_u^{AS}, to run SelfCheckGPT only on data within these thresholds; data points with AS lower than τ_l^{AS} and higher than τ_u^{AS} are directly classified as *aligned* and *misaligned*, respectively. The thresholds are determined to maximize correct predictions and minimize incorrect predictions on the validation set as

$$\tau_l^{AS} = \arg\max_x \left(\frac{1 + \mathbf{F}_a^{AS}(x)}{1 + \mathbf{F}_m^{AS}(x)/k_{eff}} \right) \tag{8.5}$$

$$\tau_u^{AS} = \arg\max_x \left(\frac{1 + (1 - \mathbf{F}_m^{AS}(x))}{1 + (1 - \mathbf{F}_a^{AS}(x))/k_{eff}} \right), \tag{8.6}$$

where $\mathbf{F}_a^{AS}$ and $\mathbf{F}_m^{AS}$ are the cumulative probabilities of the Alignment Score for aligned and misaligned data, respectively, and k_{eff} is a hyperparameter that controls efficiency; larger k_{eff} reduces penalization for incorrect classifications, allowing more data points to be classified by the Alignment Score; Sec. 8.4.3 further demonstrates how k_{eff} balances the efficiency and performance.

8.4 Experiments

This section demonstrates SHINE's effectiveness in hallucination probing (Sec. 8.4.1) and detection (Sec. 8.4.2), highlighting its superiority over existing algorithms and the importance of probing. Then, we show how our method improves efficiency while maintaining high performance (Sec. 8.4.3), directly compare our Model Knowledge Score with existing approaches (Sec. 8.4.4), and examine the robustness of SHINE to its hyperparameters (Sec. 8.4.5).

8.4.1 Hallucination Probing

We demonstrate SHINE's effectiveness on three modern LLMs: LLaMA2-13B-Chat-GPTQ [489], LLaMA3-8B-Instruct [613], and Mistral-7B-Instruct v0.3 [614], henceforth referred to as LLaMA2, LLaMA3, and Mistral. We use `en_core_web_sm` model for SpaCy, LLM temperature of 1.0, $\boldsymbol{\sigma} = 10\boldsymbol{\sigma}_0$, where $\boldsymbol{\sigma}_0$ is the standard deviation of all token embeddings in each dataset, k_{att} of 0.1, and k_{eff} of 0.1.

Dataset

Our evaluation requires datasets of prompts, LLM-generated responses, and labels (aligned, misaligned or fabricated). Since existing datasets only provide binary labels (hallucinated or not), we create two new datasets — NEC and Biography — featuring trinary labels, using existing datasets [606, 19].

The **NEC dataset** contains questions on existent and non-existent concepts and LLM-generated responses, with across six different topics (foods, sports, countries, animals, medicines, generic). For each model, we identify questions for which the model has enough knowledge and generate both aligned and misaligned responses. The same number of questions and responses about non-existent concepts are sampled to create fabricated data points and ensure balanced dataset. This results in 359 aligned, 359 misaligned, and 359 fabricated responses for the LLaMA2 model, 358 responses for each category for LLaMA3, and 476 for Mistral. Within each topic and label, the data is split evenly into validation and test sets.

The **Biography dataset** builds on top of an existing dataset of people's names on Wikipedia [19] to generate a biography of each person. After identifying the names that each LLM knows well, we pair each person with the correct job and prompt the LLM to generate the person's aligned and misaligned biographies. To create fabricated biographies, we add model-unknown people's names and pair them with incorrect random jobs so that LLMs would completely fabricate. As a result, we collect 109 aligned, 109 misaligned, and 129 fabricated biographies for LLaMA2; 174 aligned, 174 misaligned, and 174 fabricated biographies for LLaMA3; and 128 aligned, 128 misaligned, and 128 fabricated biographies for Mistral.

Each dataset is split evenly into validation and test sets within each label.

Results

We evaluate the Model Knowledge Score and Alignment Score by visualizing their cumulative distribution functions (CDFs) on the validation set of the NEC dataset for the LLaMA2 model (Fig. 8.3). The distributions of the Model Knowledge Score for fabricated and non-fabricated text (i.e., aligned and misaligned) are substantially distinct, demonstrating the score's effectiveness in detecting fabrication. Specifically, the KS statistic ($\max_{\tau\in[0,\infty)} \mathbf{F}_f^{MKS}(\tau) - \mathbf{F}_{a,m}^{MKS}(\tau)$) is 0.82 at τ^{MKS} of 0.99 (p-value 1.53e-80).[1] Alignment Score distributions for aligned and misaligned data also exhibit a clear gap, enabling us to classify data with Alignment Score below $\tau_l^{AS} = 0.00$ as aligned and above $\tau_u^{AS} = 0.14$ as misaligned, and use SelfCheckGPT for the remaining data.

Table 8.1 shows the confusion matrices of SHINE on the NEC and biography datasets

[1] τ^{MKS} is highly consistent across datasets, with values of 0.99 on NEC and 1.01 on the biography for LLaMA2.

Table 8.1: Confusion matrix showing SHINE's hallucination probing performance (%) on LLaMA2, LLaMA3, and Mistral across two datasets: NEC / biography.

			Actual		
			Aligned	Misaligned	Fabricated
LLaMA2	Predicted	Aligned	**75.53 / 81.82**	13.53 / 5.45	9.89 / 4.62
		Misaligned	11.17 / 9.09	**73.53 / 87.27**	1.65 / 9.23
		Fabricated	13.30 / 9.09	12.94 / 7.27	**88.46 / 86.15**
LLaMA3	Predicted	Aligned	**72.93 / 83.91**	20.77 / 6.90	6.04 / 1.15
		Misaligned	14.92 / 8.05	**68.31 / 75.86**	8.79 / 1.15
		Fabricated	12.15 / 8.05	10.93 / 17.24	**85.16 / 97.70**
Mistral	Predicted	Aligned	**73.44 / 89.06**	11.62 / 3.12	12.45 / 1.56
		Misaligned	5.81 / 4.69	**65.15 / 87.50**	4.15 / 3.12
		Fabricated	20.75 / 6.25	23.24 / 9.38	**83.40 / 95.31**

across the LLaMA2, LLaMA3, and Mistral models. Since no existing methods differentiate text into these three types, we cannot directly compare our approach with others. Overall, SHINE effectively differentiates the three types across both datasets and all three LLMs. For LLaMA2, the Model Knowledge Test correctly detects 88.46% of fabricated data points in the NEC dataset and 86.15% in the biography dataset. The Alignment Test further differentiates aligned and misaligned text, correctly classifying 75.53% of aligned and 73.53% of misaligned data points in the NEC dataset and 81.82% of aligned and 87.27% of misaligned data points in the biography dataset. Similarly, for LLaMA3 and Mistral, SHINE consistently classifies most data points correctly, demonstrating its robust effectiveness in hallucination probing.

8.4.2 Hallucination Detection

We evaluate SHINE's effectiveness in conventional hallucination detection (i.e., binary classification for faithful vs. hallucinated) by comparing it with two detection methods (SelfCheckGPT [30], BSDetector [599]), and five uncertainty estimation methods (negative log-likelihood (NLL) [615], entropy [327], Hallucination Score [36], Semantic Entropy [40], MARS [42][2]). For a fair comparison, we adapt SHINE's predictions to binary by combining its misaligned and fabricated classes into the *hallucinated* class, and mapping aligned to *faithful*. We do not experiment with methods outside our scope, like those requiring external knowledge [18, 19, 20], LLM fine-tuning [24, 25, 26], or supervised training with hallucination-labeled data [21, 22, 23, 43], or those with different objective of predicting hallucination likelihood from prompts rather than analyzing generated responses [33, 337].

[2]Among MARS variants, we use MARS+SE (Semantic Entropy), which exhibits the best performance.

Table 8.2: SHINE (in **bold**) outperforms all existing approaches in hallucination detection (%), especially for fabricated text, across the NEC / Biography datasets. For a fair comparison, we adapt SHINE's predictions to binary by combining its misaligned and fabricated classes into the conventional *hallucinated* class, and mapping its aligned class to the *faithful* class.

			Aligned	Misaligned	Fabricated
		Overall	*Faithful*	*Hallucinated*	
LLaMA2	**SHINE (Ours)**	**83.69 / 90.58**	73.40 / 81.82	86.47 / 94.55	91.21 / 95.38
	SelfCheckGPT	63.58 / 76.92	86.70 / 81.82	86.47 / 98.18	17.58 / 50.77
	BSDetector	76.69 / 73.80	85.64 / 70.91	82.35 / 98.18	62.09 / 52.31
	NLL	65.00 / 57.58	82.98 / 89.09	90.59 / 83.64	21.43 / 0.00
	Entropy	58.37 / 55.06	67.55 / 98.18	72.94 / 65.45	34.62 / 1.54
	Hallucination Score	48.15 / 68.02	93.62 / 89.09	37.65 / 87.27	13.19 / 27.69
	Semantic Entropy	67.46 / 78.32	39.89 / 85.45	97.65 / 61.82	64.84 / 87.69
	MARS+SE	67.46 / 76.08	39.89 / 81.82	97.65 / 61.82	64.84 / 84.62
LLaMA3	**SHINE (Ours)**	**82.04 / 90.80**	72.93 / 80.46	79.23 / 93.10	93.96 / 98.85
	SelfCheckGPT	68.14 / 80.84	78.45 / 87.36	75.96 / 91.95	50.00 / 63.22
	BSDetector	81.88 / 80.08	85.64 / 86.21	78.69 / 78.16	81.32 / 75.86
	NLL	71.41 / 50.19	66.30 / 74.71	74.86 / 70.11	73.08 / 5.75
	Entropy	76.57 / 70.11	75.69 / 55.17	69.40 / 87.36	84.62 / 67.82
	Hallucination Score	67.39 / 79.69	65.19 / 75.86	69.95 / 90.80	67.03 / 72.41
	Semantic Entropy	60.50 / 77.39	71.27 / 73.56	38.25 / 60.92	71.98 / 97.70
	MARS+SE	60.07 / 75.86	72.53 / 74.71	35.16 / 55.17	72.53 / 97.70
Mistral	**SHINE (Ours)**	**83.13 / 94.79**	73.44 / 89.06	88.38 / 96.77	87.55 / 98.44
	SelfCheckGPT	71.09 / 69.27	89.63 / 93.75	83.82 / 98.44	39.83 / 15.63
	BSDetector	82.85 / 88.54	88.80 / 93.75	85.06 / 98.44	74.69 / 73.44
	NLL	74.14 / 65.10	90.04 / 93.75	84.23 / 95.31	48.13 / 6.25
	Entropy	65.01 / 71.35	77.18 / 54.69	65.15 / 100.0	52.70 / 59.38
	Hallucination Score	63.21 / 67.19	60.58 / 65.62	72.20 / 100.0	56.85 / 35.94
	Semantic Entropy	64.32 / 84.90	73.03 / 78.12	35.68 / 84.38	84.23 / 92.19
	MARS+SE	59.75 / 83.85	74.27 / 85.94	25.31 / 79.69	79.67 / 85.94

Models and Datasets

To assess detection performance across a wider range of models and tasks, we also evaluate Qwen2.5-7B-Instruct [616] (referred to as Qwen)[3]and two benchmark hallucination detection datasets: HaluEval [28] and TriviaQA [617], in addition to the NEC and biography datasets (Sec. 8.4.1).

The **HaluEval dataset** includes 10,000 tuples of a question, relevant context, and faithful and hallucinated responses, resulting in 10,000 faithful and 10,000 hallucinated data points. The **TriviaQA dataset** contains 318 trivia questions that can be verified using Wikipedia, resulting in 257 faithful and 318 hallucinated data points for LLaMA2, 263 faithful and 318 hallucinated for LLaMA3, 268 faithful and 318 hallucinated for Mistral, and 209 faithful and 318 hallucinated for Qwen. We use half of each dataset as validation data, and the other

[3]We could not experiment with Qwen on the NEC and biography datasets since Qwen's ability to reject the unanswerable questions resulted in only a few fabricated data points.

half as test data.

As all HaluEval prompts include relevant context, we conduct only the Alignment Test without the Model Knowledge Test. For TriviaQA, which does not distinguish between misaligned and fabricated data and thus cannot decide τ^{MKS}, we set τ^{MKS} for each model to the mean from the NEC and biography datasets.

Detection Results

Table 8.2 shows the mean and class-wise accuracies[4]of each method on the NEC and biography datasets; class-wise accuracies measure the ratio of aligned text predicted as faithful, misaligned text predicted as hallucinated, and fabricated text predicted as hallucinated; mean accuracy is the average of these three values. SHINE consistently achieves the highest mean accuracies across all models and datasets. Comparing SHINE with the two hallucination detection methods (SelfCheckGPT, BSDetector), which misclassify much of the fabricated text, underscores the importance of differentiating fabricated text from misaligned text for accurate detection. For instance, for LLaMA2, SHINE correctly classifies 91.21% of fabricated text in the NEC dataset, much higher than the 17.58% and 62.09% by SelfCheckGPT and BSDetector, respectively; the underperformance of SelfCheckGPT and BSDetector are attributable to LLM's consistent generation of fabricated text [330]. The underperformance of uncertainty estimation methods (NLL, entropy, Hallucination Score, Semantic Entropy, MARS+SE) aligns with recent findings that LLMs often exhibit overconfidence in fabricated content [330]. SHINE's superior performance extends to LLaMA3 and Mistral models, reaffirming its effectiveness.

Table 8.3 demonstrates SHINE's superiority on the HaluEval and TriviaQA datasets across four models, showcasing its effectiveness with and without relevant context provided in the prompt. Notably, SHINE is the only method that correctly classifies certain hallucinated data in TriviaQA. For example, for the question "Melanie Molitor is the mom of which tennis world No 1?", LLaMA2 consistently generates the incorrect response "Serena Williams" across ten generations. While all other methods misclassify this as faithful, SHINE correctly classifying it as fabricated (i.e., hallucinated) with a Model Knowledge Score of 0.0369.

8.4.3 Efficiency of SHINE

SHINE incorporates the hyperparameter k_{eff} to control the efficiency of the Alignment Test by changing the number of data points inspected by SelfCheckGPT. While SelfCheckGPT takes 60.44 seconds to classify a single NEC data point using LLaMA2 on an NVIDIA A10G GPU, SHINE's Alignment Test completes the same task in less than 2 seconds, which

[4]We could not compare receiver operating characteristic (ROC) curves as SHINE detects hallucinations using two scores, not one.

Table 8.3: SHINE (**bolded**) outperforms all existing methods in hallucination detection (%) on HaluEval/TriviaQA.

	Type	Overall	Faithful	Hallucinated
LLaMA2	**SHINE (Ours)**	**86.69 / 93.20**	83.32 / 96.90	90.96 / 89.51
	SelfCheckGPT	76.56 / 92.74	77.30 / 98.45	75.82 / 87.04
	BSDetector	80.94 / 88.22	84.96 / 84.62	76.92 / 91.82
	NLL	62.65 / 75.29	45.66 / 73.85	79.64 / 76.73
	Entropy	63.29 / 72.18	51.50 / 77.69	78.08 / 66.67
	Hallucination Score	55.06 / 72.53	11.90 / 81.54	98.22 / 63.52
	Semantic Entropy	76.79 / 90.63	76.72 / 86.92	80.96 / 94.34
	MARS+SE	76.76 / 90.63	72.82 / 86.92	80.70 / 94.34
LLaMA3	**SHINE (Ours)**	**87.95 / 87.34**	81.00 / 87.88	94.90 / 86.79
	SelfCheckGPT	78.77 / 87.15	80.08 / 89.39	77.46 / 84.91
	BSDetector	84.96 / 86.69	86.56 / 84.09	83.36 / 89.31
	NLL	32.00 / 57.58	41.20 / 48.48	22.80 / 66.67
	Entropy	82.79 / 63.20	78.86 / 62.88	86.72 / 63.52
	Hallucination Score	70.28 / 71.49	68.38 / 56.82	72.18 / 86.16
	Semantic Entropy	81.12 / 83.68	81.98 / 81.82	80.26 / 85.53
	MARS+SE	81.12 / 84.06	81.98 / 82.58	80.26 / 85.53
Mistral	**SHINE (Ours)**	**83.13 / 89.51**	89.46 / 86.57	76.80 / 92.45
	SelfCheckGPT	77.97 / 89.14	78.88 / 85.82	77.06 / 92.45
	BSDetector	81.92 / 85.19	85.32 / 83.58	78.52 / 86.79
	NLL	79.05 / 78.25	64.68 / 87.31	93.42 / 69.18
	Entropy	67.65 / 72.58	55.08 / 67.16	80.22 / 77.99
	Hallucination Score	43.78 / 68.36	75.82 / 45.52	11.74 / 91.19
	Semantic Entropy	79.18 / 82.46	78.60 / 76.87	79.76 / 88.05
	MARS+SE	79.35 / 82.63	78.98 / 77.21	79.72 / 88.05
Qwen	**SHINE (Ours)**	**89.40 / 85.81**	82.74 / 92.38	96.06 / 79.25
	SelfCheckGPT	76.51 / 85.18	76.60 / 92.38	76.42 / 77.99
	BSDetector	85.51 / 84.81	84.22 / 79.05	86.80 / 90.57
	NLL	59.47 / 48.60	25.94 / 69.52	93.00 / 27.67
	Entropy	64.27 / 50.07	45.52 / 48.57	83.12 / 51.57
	Hallucination Score	71.76 / 68.18	55.64 / 53.33	87.88 / 83.02
	Semantic Entropy	80.15 / 83.90	78.58 / 86.67	81.72 / 81.13
	MARS+SE	80.15 / 83.90	78.58 / 86.67	81.72 / 81.13

is over 30 times faster. We evaluate SHINE's performance and computation time on the NEC dataset and the LLaMA2 model for k_{eff} values ranging from 0.1 to 10 (Fig. 8.1), with comparisons to other methods. Results for other datasets and models, which follow similar trends.

SHINE's performance improves with increased computation time, indicating that SelfCheckGPT often helps the Alignment Test. Even at low k_{eff} values, SHINE outperforms BSDetector, the best-performing compared method, highlighting its superiority in both efficiency and accuracy. Table 8.4 additionally shows the number of data points handled by SelfCheckGPT and the overall accuracy, while varying the hyperparameter k_{eff} from 0.1 to 1.0. SHINE applies SelfCheckGPT to only 30 out of 332 data points when k_{eff} is 1.0, while achieving the overall accuracy of 77.08%, which exceeds the state-of-the-art performance of 76.69%. Lowering k_{eff} further improves the accuracy, offering a flexible balance between

Table 8.4: We vary k_{eff} for the LLaMA2 model on the 322 NEC data points evaluated with the Alignment Test. We report the number of data points examined by SelfCheckGPT (# SelfCheckGPT) and the corresponding accuracy. Larger k_{eff} values offer higher efficiency while preserving state-of-the-art accuracy, whereas smaller k_{eff} values yield additional accuracy gains.

k_{eff}	# SelfCheckGPT	Accuracy (%)
1.0	30/322	77.08
0.5	160/322	82.26
0.2	256/322	83.21
0.1	284/322	83.69

efficiency and effectiveness.

8.4.4 Effectiveness of Model Knowledge Test

We assess the effectiveness of Model Knowledge Score in distinguishing fabricated from non-fabricated (i.e., aligned and misaligned) text by comparing its receiver operating characteristic (ROC) curves against existing methods on the NEC dataset for LLaMA2, LLaMA3, and Mistral models (Fig. 8.4). For a comprehensive comparison, we include hallucination detection methods, even though they are not designed to differentiate misaligned and fabricated text.

The Model Knowledge Score achieves the highest ROC area under the curve (AUC) across all models: 0.9410 for LLaMA2, 0.9016 for LLaMA3, and 0.8730 for Mistral. Limited AUC of state-of-the-art uncertainty estimation methods (Semantic Entropy, MARS+SE) highlights their weakness when the LLM is overconfident and consistently generates identical fabricated responses across multiple generations. Notably, the Model Knowledge Score without likelihood scaling (Eq. 8.2) still provides an informative measure of model

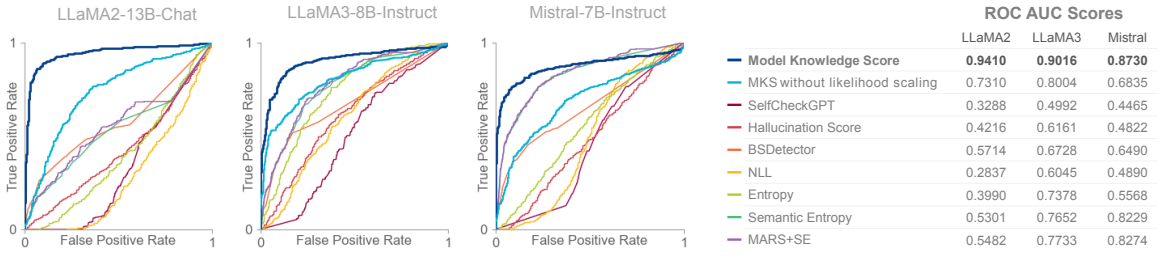


Figure 8.4: ROC curves of the Model Knowledge Score and existing methods for differentiating fabricated and non-fabricated (i.e., aligned and misaligned) text on the NEC dataset generated by the LLaMA2, LLaMA3, and Mistral models. The Model Knowledge Score achieves the highest ROC AUC values, demonstrating its superiority in identifying whether a model has sufficient knowledge about the prompt and text.

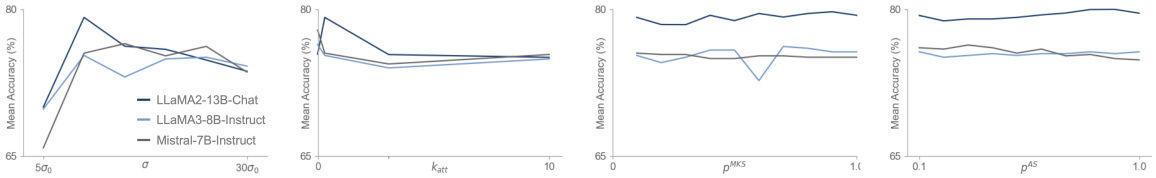


Figure 8.5: SHINE's mean accuracy on the NEC dataset for the LLaMA2, LLaMA3, and Mistral models while varying each of $\boldsymbol{\sigma}$, k_{att}, p^{MKS}, and p^{AS}.

knowledge, while incorporating likelihood scaling significantly improves performance.

8.4.5 Hyperparameter Sensitivity

SHINE involves four hyperparameters, $\boldsymbol{\sigma}$, k_{att}, p^{MKS}, and p_{AS}, which we set to $10\boldsymbol{\sigma}_0$, 0.1, 0.5, and 0.1, respectively, to achieve the best performance. To assess SHINE'd robustness to these hyperparameters, we evaluate its performance on the LLaMA2, LLaMA3, and Mistral models, while varying each hyperparameter, and visualize the results on the NEC dataset in Fig. 8.5.

Our hyperparameter configuration mostly yields the best overall performance. Notably, p^{MKS}, and p_{AS} values have little impact on SHINE, confirming its robustness to these parameters. For k_{att}, lower values provide the strongest results for each model (0.1 for LLaMA2, 0 for LLaMA3 and Mistral), though higher values still achieve state-of-the-art accuracies. For $\boldsymbol{\sigma}$, values between $10\boldsymbol{\sigma}_0$ and $20\boldsymbol{\sigma}_0$ perform well, while extremes — too low ($5\boldsymbol{\sigma}_0$) or too high ($30\boldsymbol{\sigma}_0$) — degrade the performance.

8.5 Conclusion

We develop SHINE, a novel method to classify LLM-generated text into *aligned*, *misaligned*, and *fabricated* to identify the causes of hallucinations and improve existing detection methods. Model Knowledge Test effectively detects fabricated text and Alignment Test further differentiates aligned and misaligned text. We aim to develop a more efficient and effective technique and to evaluate our method on a broader range of tasks.

Part IV

Conclusions

CHAPTER 9
CONCLUSIONS AND FUTURE DIRECTIONS

9.1 Research Contributions

My thesis contributes to the fields of human-computer interaction, machine learning, visualization, and their intersections by developing approaches to **attribute** risks (Part I), **explain** model failures (Part II), and **guide** risk mitigation (Part III).

Bridging Interpretability and Safety: A New Paradigm for Responsible and Safe AI

- We present the **first comprehensive survey** that connects LLM interpretation and safety (Chapter 3). We introduce a unified framework that summarizes safety-focused interpretation methods, interpretation-informed safety enhancement strategies, and practical tools that operationalize them. This timely survey provides researchers and practitioners with a clear roadmap of key advancements, establishing interpretation as a cornerstone for building safer LLMs.

- LLM Attributor (Chapter 4) elevates training data attribution, a powerful yet underexplored interpretation technique, into a **practical safety tool**. By enabling practitioners to directly visualize and inspect the training data most responsible for an output, it delivers interactive, evidence-based reliability checks that can be leveraged to diagnose unsafe generations.

- VisCUIT (Chapter 6) transforms interpretation into **fast, intuitive diagnostics of spurious model behaviors**, exposing how and why models produce unreliable predictions. VisCUIT empowers practitioners to identify and preempt risks before deployment in high-stakes applications.

- CRAYON (Chapter 7) pioneers an algorithmic bridge between interpretation and intervention, offering a **practical and scalable solution to rectify improper model attention**. CRAYON enables both classic and modern interpretation methods to directly enhance model robustness and reliability.

Revolutionizing AI Understanding: Global Reach through Interactive Visualization

- Diffusion Explainer (Chapter 5) demonstrates the **substantial learning benefits of interactive visualization** for understanding generative AI compared to traditional resources, validated by a 56-participant user study. Together with the follow-up Transformer Explainer, these tools have reached over 638,000 users across 200+

countries and been integrated into top university courses (e.g., MIT, Columbia), establishing interactive visualization as a proven approach for democratizing AI education.

- LLM Attributor (Chapter 4) offers **visualizations for computational notebooks**, making attribution fast, legible, and seamlessly integrated into workflows. Features such as influential example visualization and side-by-side comparison allow practitioners to interactively explore how outputs arise, simplifying adoption of attribution.

- VisCUIT (Chapter 6) advances **interactive exploration of model internals**, enabling researchers and practitioners to easily probe and interpret complex attention patterns. Its visualization design provides an accessible yet powerful interface for understanding model behaviors, complementing its role as a diagnostic tool for safety.

Breakthrough Algorithms Driving Safer Models

- CRAYON (Chapter 7) introduces the **first practical and scalable method to correct model attention** with simple yes-no feedback. Validated by a large-scale evaluation with nearly 6,000 participants, CRAYON demonstrates effectiveness, scalability, and practicality in refining model attention.

- SHINE (Chapter 8) establishes a **new paradigm for probing LLM hallucinations through model knowledge**. Without external knowledge, supervised training, or fine-tuning, SHINE predicts whether an LLM possesses sufficient knowledge to generate a faithful response, and whether the response aligns with the model's knowledge, providing a powerful diagnostic for hallucination detection and analysis.

Open-source and Extensible Systems Accelerating Global AI Innovation

- Across this thesis, I have released **many open-source, paradigm-shifting tools**— from web-based visual interfaces (Chapter 4, 5, 6) to scalable algorithms to improve AI safety (Chapter 7, 8)— that are freely available for adoption and extension. This work has empowered researchers, practitioners, and educators worldwide to build upon my designs, accelerating innovation and advancing the responsible use of AI.

9.2 Impact

My thesis research is already making significant impact on industry and society.

- Diffusion Explainer (Chapter 5) and its followup work Transformer Explainer have pioneered generative AI education and literacy, being used by more than **638,000 novices across 200+ countries** and integrated into top university courses (e.g., MIT,

Columbia). These contributions have been recognized by prestigious honors, including being named an NCWIT AiC Collegiate Award Finalist and winning the IEEE VIS Best Poster Award.

- My visualization designs for safety and interpretability, exemplified in LLM Attributor (Chapter 4) and VisCUIT (Chapter 6), led to **Cisco's OpenRAI library for responsible AI**, and informed the creation of a visualization interface with AVAST to strengthen web security.
- My research was recognized with the **Korean Honor Scholarship**, a competitive fellowship awarded by the Korean government to only 36 Korean students studying in the US.
- My works have been recognized with multiple awards: VisCUIT (Chapter 6) received the **Best Project Award Honorable Mention** (2nd place) in CS7001, a course taken by over 60 first-year CS PhD students at Georgia Tech; SHINE (Chapter 8) received the NeurIPS SoLaR Workshop Travel Award; CRAYON (Chapter 7) received the IEEE BigData Student Travel Award. Beyond academia, these ideas were translated into industry impact, with my work winning the **Best Poster Award** in the Adobe Firefly intern poster session.
- My thesis ideas contributed to securing two highly competitive funded grant awards: the concept of democratizing AI (Chapter 5 and 6) formed the foundation of a funded **$1.2M NSF Medium Grant**, while educational innovation in the era of generative AI (Chapter 5) received a **Google Research Award**.

9.3 Future Directions

While this dissertation makes significant contributions and demonstrates impact across academia, industry, and society, it also opens several promising research directions in developing safer, human-centered AI systems.

9.3.1 Unified Systems From Risk Diagnosis to Automatic Mitigation

For safer AI, this dissertation addresses three complementary capabilities, attributing risks, explaining them, and guiding corrective interventions. While each capability is valuable on its own, future progress may integrate them into unified end-to-end systems that monitor and explain model behavior, incorporate human feedback, and automatically respond with appropriate interventions. By connecting attribution, explanation, and mitigation in a closed loop, these systems would make AI safety approaches more actionable, scalable, and practical for real-world deployments, where failures must often be addressed quickly and at scale.

9.3.2 Adaptive Explanations for Diverse Users and Contexts

The explanations in this dissertation are designed to be effective for specific target audiences, yet they remain largely static: they present the same explanation regardless of a user's background, expertise, or goals. Future research should move toward adaptive explanation systems that personalize content, interaction style, and level of detail according to users' needs and situational context. For example, beginners seeking to learn basic concepts may benefit from high-level visuals and step-by-step walkthroughs, whereas machine learning practitioners aiming to mitigate model risks may prefer detailed information about model internals, such as attribution results or neuron-level diagnostics. Future interfaces could infer these needs through explicit preferences or interaction behavior, then dynamically tailor explanations in real time, making AI understanding more effective, efficient, and broadly accessible.

9.3.3 Cross-model Interfaces for Rapidly Evolving AI Ecosystems

Modern AI ecosystems are evolving rapidly: new architectures emerge continuously, and real-world applications increasingly rely on heterogeneous pipelines that combine multiple models within a single workflow. This challenge is especially pronounced in multimodal and agentic systems, which coordinate diverse modalities (e.g., language, vision, audio) and modules (e.g., retrieval modules, external tools). Future research should therefore move beyond model-specific tools toward more general frameworks that support diverse models and modalities. Future interfaces should reveal not only how individual components behave, but also how information flows across components and where failures arise within the broader pipeline. Building such cross-model interfaces could make interpretability more scalable, transferable, and sustainable in an era of continuous AI innovation.

9.3.4 Refining AI Safety Taxonomies for Emerging Risks

Much of current AI safety research focuses on established risk types such as hallucination, harmful generation, bias, and privacy leakage. While these concerns remain important, future AI systems may introduce risks that do not fit neatly within existing taxonomies. In particular, as models become increasingly embedded in physical environments, autonomous agents, and complex decision-making workflows, safety frameworks must continually evolve to capture new forms of harm.

9.3.5 Evaluating the Effectiveness of Interpretability Methods

Interpretability methods are often motivated by the promise of improving AI safety, yet explanations are not inherently reliable or beneficial. Different methods can vary substantially in their faithfulness, robustness, and usefulness to real users. Moreover, the way

explanations are presented can significantly affect whether users are helped or misled. Thus, future work should place greater emphasis on systematically evaluating interpretability methods not only as technical algorithms, but also as practical tools experienced and used by people in real-world settings.

9.4 Conclusion

I believe that AI should not only become more capable, but also safer, more reliable, and aligned with human values. As AI systems increasingly shape decisions, creativity, and everyday life, my continued mission is to develop human-centered methods that help various users uncover model risks, understand risky behaviors, and fix when risks arise. This dissertation is an initial step toward this mission, and addresses the fundamental and practical challenges of fortifying AI safety through visual and algorithmic explanations.

REFERENCES

[1] D. Spiegelhalter, "Should We Trust Algorithms?" *Harvard Data Science Review*, vol. 2, no. 1, Jan. 2020, https://hdsr.mitpress.mit.edu/pub/56lnenzj.

[2] J. O. Arowosegbe, "Data bias, intelligent systems and criminal justice outcomes," *International Journal of Law and Information Technology*, vol. 31, no. 1, pp. 22–45, 2023.

[3] M. Raghavan, S. Barocas, J. Kleinberg, and K. Levy, "Mitigating bias in algorithmic hiring: Evaluating claims and practices," in *Proceedings of the 2020 conference on fairness, accountability, and transparency*, 2020, pp. 469–481.

[4] C. Soares, "It's too easy to make ai chatbots lie about health information, study finds," *Reuters*, Jul. 2025.

[5] G. Burke, "Chatbots' inaccurate, misleading responses about us elections threaten to keep voters from polls," *Associated Press (AP)*, Feb. 2024.

[6] S. v. d. Linden, "Ai-generated fake news is coming to an election near you," *WIRED*, Jan. 2024.

[7] P. Khatiwada, G. Donaher, J. Navarro, and L. Bhatta, *Culling misinformation from gen ai: Toward ethical curation and refinement*, 2025. arXiv: 2507.14242 [cs.CY].

[8] J. Zou and L. Schiebinger, "Design ai so that it's fair," *Nature*, vol. 559, pp. 324–326, 2018.

[9] J. Buolamwini and T. Gebru, "Gender shades: Intersectional accuracy disparities in commercial gender classification," in *Conference on Fairness, Accountability and Transparency, FAT 2018, 23-24 February 2018, New York, NY, USA*, ser. Proceedings of Machine Learning Research, vol. 81, PMLR, 2018, pp. 77–91.

[10] V. Albiero, K. K. S, K. Vangara, K. Zhang, M. C. King, and K. W. Bowyer, "Analysis of gender inequality in face recognition accuracy," in *IEEE Winter Applications of Computer Vision Workshops, WACV Workshops 2020, Snowmass Village, CO, USA, March 1-5, 2020*, IEEE, 2020, pp. 81–89.

[11] J. G. Cavazos, P. J. Phillips, C. D. Castillo, and A. J. O'Toole, "Accuracy comparison across face recognition algorithms: Where are we on measuring race bias?" *IEEE Trans. Biom. Behav. Identity Sci.*, vol. 3, no. 1, pp. 101–111, 2021.

[12] A. Lambrecht and C. Tucker, "Algorithmic bias? an empirical study of apparent gender-based discrimination in the display of stem career ads," *Management Science*, vol. 65, no. 7, pp. 2966–2981, 2019.

[13] J. Angwin, J. Larson, S. Mattu, and L. Kirchner, *Machine bias: There's software used across the country to predict future criminals. and it's biased against blacks.* 2016.

[14] K. K. Singh, D. Mahajan, K. Grauman, Y. J. Lee, M. Feiszli, and D. Ghadiyaram, "Don't judge an object by its context: Learning to overcome contextual bias," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, Computer Vision Foundation / IEEE, 2020, pp. 11 067–11 075.

[15] 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)*, 2021.

[16] E. Derman, "Dataset bias mitigation through analysis of CNN training scores," *CoRR*, vol. abs/2106.14829, 2021. arXiv: 2106.14829.

[17] C. Qian et al., "Towards tracing trustworthiness dynamics: Revisiting pre-training period of large language models," in *Findings of the Association for Computational Linguistics: ACL 2024*, L.-W. Ku, A. Martins, and V. Srikumar, Eds., Bangkok, Thailand: Association for Computational Linguistics, Aug. 2024, pp. 4864–4888.

[18] S. Lin, J. Hilton, and O. Evans, "Truthfulqa: Measuring how models mimic human falsehoods," *arXiv preprint arXiv:2109.07958*, 2021.

[19] S. Min et al., "Factscore: Fine-grained atomic evaluation of factual precision in long form text generation," *arXiv preprint arXiv:2305.14251*, 2023.

[20] L. Tang, P. Laban, and G. Durrett, "MiniCheck: Efficient Fact-Checking of LLMs on Grounding Documents," *arXiv preprint arXiv:2404.10774*, 2024.

[21] Y. Chen et al., "Hallucination detection: Robustly discerning reliable answers in large language models," in *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, 2023, pp. 245–255.

[22] A. Azaria and T. Mitchell, "The internal state of an llm knows when it's lying," *arXiv preprint arXiv:2304.13734*, 2023.

[23] W. Su et al., "Unsupervised real-time hallucination detection based on the internal states of large language models," *arXiv preprint arXiv:2403.06448*, 2024.

[24] H. Zhang et al., "R-tuning: Teaching large language models to refuse unknown questions," *arXiv preprint arXiv:2311.09677*, 2023.

[25] H. Xu et al., "Rejection improves reliability: Training llms to refuse unknown questions using rl from knowledge feedback," *arXiv preprint arXiv:2403.18349*, 2024.

[26] J. Li, Y. Tang, and Y. Yang, "Know the unknown: An uncertainty-sensitive method for llm instruction tuning," *arXiv preprint arXiv:2406.10099*, 2024.

[27] R. Friel and A. Sanyal, "Chainpoll: A high efficacy method for llm hallucination detection," *arXiv preprint arXiv:2310.18344*, 2023.

[28] J. Li, X. Cheng, W. X. Zhao, J.-Y. Nie, and J.-R. Wen, "Halueval: A large-scale hallucination evaluation benchmark for large language models," *arXiv preprint arXiv:2305.11747*, 2023.

[29] X. Zhang et al., "Self-alignment for factuality: Mitigating hallucinations in llms via self-evaluation," *arXiv preprint arXiv:2402.09267*, 2024.

[30] P. Manakul, A. Liusie, and M. J. Gales, "Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models," *arXiv preprint arXiv:2303.08896*, 2023.

[31] Y. A. Yadkori, I. Kuzborskij, A. György, and C. Szepesvári, "To Believe or Not to Believe Your LLM," *arXiv preprint arXiv:2406.02543*, 2024.

[32] Y. A. Yadkori et al., "Mitigating llm hallucinations via conformal abstention," *arXiv preprint arXiv:2405.01563*, 2024.

[33] C. Chen et al., "INSIDE: LLMs' internal states retain the power of hallucination detection," in *The Twelfth International Conference on Learning Representations*, 2024.

[34] S. Bengio, O. Vinyals, N. Jaitly, and N. Shazeer, "Scheduled sampling for sequence prediction with recurrent neural networks," *Advances in neural information processing systems*, vol. 28, 2015.

[35] T. Iqbal and S. Qureshi, "The survey: Text generation models in deep learning," *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 6, pp. 2515–2528, 2022.

[36] T. Zhang et al., "Enhancing uncertainty-based hallucination detection with stronger focus," *arXiv preprint arXiv:2311.13230*, 2023.

[37] U. Kamath, K. Keenan, G. Somers, and S. Sorenson, "LLM challenges and solutions," in *Large Language Models: A Deep Dive: Bridging Theory and Practice*, Springer, 2024, pp. 219–274.

[38] Y. Huang, J. Song, Z. Wang, H. Chen, and L. Ma, "Look before you leap: An exploratory study of uncertainty measurement for large language models," *arXiv preprint arXiv:2307.10236*, 2023.

[39] E. Quevedo, J. Yero, R. Koerner, P. Rivas, and T. Cerny, "Detecting hallucinations in large language model generation: A token probability approach," *arXiv preprint arXiv:2405.19648*, 2024.

[40] S. Farquhar, J. Kossen, L. Kuhn, and Y. Gal, "Detecting hallucinations in large language models using semantic entropy," *Nature*, vol. 630, no. 8017, pp. 625–630, 2024.

[41] B. Hou, Y. Zhang, J. Andreas, and S. Chang, "A probabilistic framework for llm hallucination detection via belief tree propagation," *arXiv preprint arXiv:2406.06950*, 2024.

[42] Y. F. Bakman, D. N. Yaldiz, B. Buyukates, C. Tao, D. Dimitriadis, and S. Avestimehr, "Mars: Meaning-aware response scoring for uncertainty estimation in generative llms," *arXiv preprint arXiv:2402.11756*, 2024.

[43] D. N. Yaldiz et al., "Do not design, learn: A trainable scoring function for uncertainty estimation in generative llms," *arXiv preprint arXiv:2406.11278*, 2024.

[44] L. Liu, R. Pourreza, S. Panchal, A. Bhattacharyya, Y. Qin, and R. Memisevic, "Enhancing hallucination detection through noise injection," *arXiv preprint arXiv:2502.03799*, 2025.

[45] F. Stahlberg and B. Byrne, "On NMT search errors and model errors: Cat got your tongue?" In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, K. Inui, J. Jiang, V. Ng, and X. Wan, Eds., Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 3356–3362.

[46] A. Holtzman, J. Buys, L. Du, M. Forbes, and Y. Choi, "The curious case of neural text degeneration," *arXiv preprint arXiv:1904.09751*, 2019.

[47] S. Zhang, T. Yu, and Y. Feng, "TruthX: Alleviating hallucinations by editing large language models in truthful space," in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, L.-W. Ku, A.

Martins, and V. Srikumar, Eds., Bangkok, Thailand: Association for Computational Linguistics, Aug. 2024, pp. 8908–8949.

[48] S. Sagawa, A. Raghunathan, P. W. Koh, and P. Liang, "An investigation of why overparameterization exacerbates spurious correlations," in *International Conference on Machine Learning*, PMLR, 2020, pp. 8346–8356.

[49] Y. Li and N. Vasconcelos, "Repair: Removing representation bias by dataset resampling," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 9572–9581.

[50] 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.

[51] E. Z. Liu et al., "Just train twice: Improving group robustness without training group information," in *International Conference on Machine Learning*, PMLR, 2021, pp. 6781–6792.

[52] 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.

[53] 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.

[54] J. Lee, E. Kim, J. Lee, J. Lee, and J. Choo, "Learning debiased representation via disentangled feature augmentation," *Advances in Neural Information Processing Systems*, vol. 34, pp. 25 123–25 133, 2021.

[55] E. Kim, J. Lee, and J. Choo, "Biaswap: Removing dataset bias with bias-tailored swapping augmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 14 992–15 001.

[56] M.-C. Chiu, P.-Y. Chen, and X. Ma, "Better may not be fairer: Can data augmentation mitigate subgroup degradation?" *arXiv preprint arXiv:2212.08649*, 2022.

[57] H. Yao et al., "Improving out-of-distribution robustness via selective augmentation," in *International Conference on Machine Learning*, PMLR, 2022, pp. 25 407–25 437.

[58] Z. Han et al., "Umix: Improving importance weighting for subpopulation shift via uncertainty-aware mixup," *arXiv preprint arXiv:2209.08928*, 2022.

[59] R. Schwartz and G. Stanovsky, "On the limitations of dataset balancing: The lost battle against spurious correlations," *arXiv preprint arXiv:2204.12708*, 2022.

[60] B. Kim, H. Kim, K. Kim, S. Kim, and J. Kim, "Learning not to learn: Training deep neural networks with biased data," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 9012–9020.

[61] S. Singla and S. Feizi, "Salient imagenet: How to discover spurious features in deep learning?" *arXiv preprint arXiv:2110.04301*, 2021.

[62] Y. Zhang et al., "Causaladv: Adversarial robustness through the lens of causality," *arXiv preprint arXiv:2106.06196*, 2021.

[63] S. Nagpal, M. Singh, R. Singh, and M. Vatsa, "Attribute aware filter-drop for bias invariant classification," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020, pp. 32–33.

[64] 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.

[65] S. Asgari et al., "Masktune: Mitigating spurious correlations by forcing to explore," in *Advances in Neural Information Processing Systems*, 2022.

[66] S. Liu, X. Zhang, N. Sekhar, Y. Wu, P. Singhal, and C. Fernandez-Granda, "Avoiding spurious correlations via logit correction," *arXiv preprint arXiv:2212.01433*, 2022.

[67] 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.

[68] T. Kulesza, M. Burnett, W.-K. Wong, and S. Stumpf, "Principles of explanatory debugging to personalize interactive machine learning," in *Proceedings of the 20th international conference on intelligent user interfaces*, 2015, pp. 126–137.

[69] S. Teso and K. Kersting, "Explanatory interactive machine learning," in *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, 2019, pp. 239–245.

[70] P. Schramowski et al., "Making deep neural networks right for the right scientific reasons by interacting with their explanations," *Nature Machine Intelligence*, vol. 2, no. 8, pp. 476–486, 2020.

[71] G. Saha and K. Roy, "Saliency guided experience packing for replay in continual learning," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2023, pp. 5273–5283.

[72] J. Schneider and M. Vlachos, "Reflective-net: Learning from explanations," *Data Mining and Knowledge Discovery*, pp. 1–22, 2023.

[73] A. S. Ross, M. C. Hughes, and F. Doshi-Velez, "Right for the right reasons: Training differentiable models by constraining their explanations," *arXiv preprint arXiv:1703.03717*, 2017.

[74] 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.

[75] Y. Gao, T. S. Sun, G. Bai, S. Gu, S. R. Hong, and Z. Liang, "Res: A robust framework for guiding visual explanation," in *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2022, pp. 432–442.

[76] 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.

[77] G. Plumb, M. T. Ribeiro, and A. Talwalkar, "Finding and fixing spurious patterns with explanations," *arXiv preprint arXiv:2106.02112*, 2021.

[78] W. Stammer, P. Schramowski, and K. Kersting, "Right for the right concept: Revising neuro-symbolic concepts by interacting with their explanations," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 3619–3629.

[79] C. Ma et al., "Eye-gaze-guided vision transformer for rectifying shortcut learning," *IEEE Transactions on Medical Imaging*, 2023.

[80] H. Shen et al., "Human-ai interactive and continuous sensemaking: A case study of image classification using scribble attention maps," in *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*, 2021, pp. 1–8.

[81] S. Rao, M. Böhle, A. Parchami-Araghi, and B. Schiele, "Studying how to efficiently and effectively guide models with explanations," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 1922–1933.

[82] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," in *2nd International Conference on Learning Representations, ICLR Workshop Track Proceedings*, 2014.

[83] 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 *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, IEEE Computer Society, 2017, pp. 618–626.

[84] R. Fong and A. Vedaldi, "Net2vec: Quantifying and explaining how concepts are encoded by filters in deep neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8730–8738.

[85] F. Hohman, H. Park, C. Robinson, and D. H. (Chau, "Summit: Scaling deep learning interpretability by visualizing activation and attribution summarizations," *IEEE Trans. Vis. Comput. Graph.*, vol. 26, no. 1, pp. 1096–1106, 2020.

[86] H. Park et al., "NeuroCartography: Scalable Automatic Visual Summarization of Concepts in Deep Neural Networks," 2021.

[87] N. Das et al., "Bluff: Interactively deciphering adversarial attacks on deep neural networks," 2020.

[88] G. D. Cantareira, R. F. Mello, and F. V. Paulovich, "Explainable adversarial attacks in deep neural networks using activation profiles," *arXiv preprint arXiv:2103.10229*, 2021.

[89] X. Ma et al., "Understanding adversarial attacks on deep learning based medical image analysis systems," *Pattern Recognition*, vol. 110, p. 107 332, 2021.

[90] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," in *Proceedings of the 25rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019.

[91] H. Joo, C. Bao, I. Sen, F. Huang, and L. Battle, "Guided hyperparameter tuning through visualization and inference," *arXiv preprint arXiv:2105.11516*, 2021.

[92] H. Jin, *Keras documentation: Visualize the hyperparameter tuning process*, 2021.

[93] S. Mostafa, D. Mondal, M. Beck, C. Bidinosti, C. Henry, and I. Stavness, "Visualizing feature maps for model selection in convolutional neural networks," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 1362–1371.

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

[95] T. Räuker, A. Ho, S. Casper, and D. Hadfield-Menell, "Toward transparent ai: A survey on interpreting the inner structures of deep neural networks," in *2023 ieee conference on secure and trustworthy machine learning (satml)*, IEEE, 2023, pp. 464–483.

[96] W. Hsieh et al., "A comprehensive guide to explainable ai: From classical models to llms," *arXiv preprint arXiv:2412.00800*, 2024.

[97] C. Singh, J. P. Inala, M. Galley, R. Caruana, and J. Gao, "Rethinking interpretability in the era of large language models," *arXiv preprint arXiv:2402.01761*, 2024.

[98] A. McGrath and A. Jonker, *What is ai interpretability?* 2024.

[99] R. Ajwani, S. R. Javaji, F. Rudzicz, and Z. Zhu, "Llm-generated black-box explanations can be adversarially helpful," *arXiv preprint arXiv:2405.06800*, 2024.

[100] A. Madsen, S. Reddy, and S. Chandar, "Post-hoc interpretability for neural nlp: A survey," *ACM Comput. Surv.*, vol. 55, no. 8, Dec. 2022.

[101] P. W. Koh and P. Liang, "Understanding black-box predictions via influence functions," in *International conference on machine learning*, PMLR, 2017, pp. 1885–1894.

[102] G. Pruthi, F. Liu, S. Kale, and M. Sundararajan, "Estimating training data influence by tracing gradient descent," *Advances in Neural Information Processing Systems*, vol. 33, pp. 19 920–19 930, 2020.

[103] R. Grosse et al., "Studying large language model generalization with influence functions," *arXiv preprint arXiv:2308.03296*, 2023.

[104] A. Ghorbani and J. Zou, "Data shapley: Equitable valuation of data for machine learning," in *International conference on machine learning*, PMLR, 2019, pp. 2242–2251.

[105] A. Ilyas, S. M. Park, L. Engstrom, G. Leclerc, and A. Madry, "Datamodels: Predicting predictions from training data," in *ArXiv preprint arXiv:2202.00622*, 2022.

[106] X. Han and Y. Tsvetkov, "Orca: Interpreting prompted language models via locating supporting data evidence in the ocean of pretraining data," *arXiv preprint arXiv:2205.12600*, 2022.

[107] K. Guu, A. Webson, E. Pavlick, L. Dixon, I. Tenney, and T. Bolukbasi, "Simfluence: Modeling the influence of individual training examples by simulating training runs," *arXiv preprint arXiv:2303.08114*, 2023.

[108] R. D. Cook and S. Weisberg, "Characterizations of an empirical influence function for detecting influential cases in regression," *Technometrics*, vol. 22, no. 4, pp. 495–508, 1980.

[109] S. M. Park, K. Georgiev, A. Ilyas, G. Leclerc, and A. Madry, "Trak: Attributing model behavior at scale," in *Proceedings of the 40th International Conference on Machine Learning*, ser. ICML'23, Honolulu, Hawaii, USA: JMLR.org, 2023.

[110] Y. Kwon, E. Wu, K. Wu, and J. Zou, "Datainf: Efficiently estimating data influence in lora-tuned llms and diffusion models," *arXiv:2310.00902*, 2023.

[111] A. Karpathy, *ConvNetJS: Deep Learning in your browser*, https://cs.stanford.edu/people/karpathy/convnetjs/index.html, Accessed on: 2023-11-26, 2014.

[112] D. Smilkov, S. Carter, D. Sculley, F. B. Viégas, and M. Wattenberg, "Direct-manipulation Visualization of Deep Networks," *arXiv preprint arXiv:1708.03788*, 2017.

[113] G. Goh, "Why Momentum Really Works," *Distill*, vol. 2, no. 4, e6, 2017.

[114] S. Carter and M. Nielsen, "Using artificial intelligence to augment human intelligence," *Distill*, 2017, https://distill.pub/2017/aia.

[115] A. Madsen, "Visualizing memorization in rnns," *Distill*, 2019, https://distill.pub/2019/memorization-in-rnns.

[116] A. Agnihotri and N. Batra, "Exploring Bayesian Optimization," *Distill*, vol. 5, no. 5, e26, 2020.

[117] B. Sanchez-Lengeling, E. Reif, A. Pearce, and A. B. Wiltschko, "A Gentle Introduction to Graph Neural Networks," *Distill*, vol. 6, no. 9, e33, 2021.

[118] C. Olah, *colahś blog*, http://colah.github.io, Accessed on: 2023-04-30, 2023.

[119] Z. J. Wang et al., "CNN Explainer: Learning Convolutional Neural Networks with Interactive Visualization," *IEEE Conference on Visual Analytics Science and Technology (VAST)*, 2020.

[120] M. Kahng, N. Thorat, D. H. Chau, F. Viégas, and M. Wattenberg, "GAN Lab: Understanding Complex Deep Generative Models using Interactive Visual Experimentation," *IEEE Transactions on Visualization and Computer Graphics*, vol. 25, no. 1, 2019.

[121] A. P. Norton and Y. Qi, "Adversarial-Playground: A Visualization Suite Showing how Adversarial Examples Fool Deep Learning," in *Visualization for Cyber Security (VizSec), 2017 IEEE Symposium on*, IEEE, 2017, pp. 1–4.

[122] C. Pierse, *Transformers interpret: Explainability for any transformers models in 2 lines*, https://github.com/cdpierse/transformers-interpret?tab=readme-ov-file, 2021.

[123] G. Sarti, N. Feldhus, L. Sickert, and O. van der Wal, "Inseq: An interpretability toolkit for sequence generation models," in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstra-*

tions), D. Bollegala, R. Huang, and A. Ritter, Eds., Toronto, Canada: Association for Computational Linguistics, Jul. 2023, pp. 421–435.

[124] Google, *Analyze model behavior with interpretability tools*, https://ai.google.dev/responsible/model_behavior, 2024.

[125] I. Tenney et al., "The language interpretability tool: Extensible, interactive visualizations and analysis for NLP models," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, Q. Liu and D. Schlangen, Eds., Online: Association for Computational Linguistics, Oct. 2020, pp. 107–118.

[126] J. Alammar, "Ecco: An open source library for the explainability of transformer language models," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, Association for Computational Linguistics, 2021.

[127] T. Worledge, J. H. Shen, N. Meister, C. Winston, and C. Guestrin, "Unifying corroborative and contributive attributions in large language models," *arXiv preprint arXiv:2311.12233*, 2023.

[128] J. Adebayo, "Fairml : Toolbox for diagnosing bias in predictive modeling," 2016.

[129] P. Saleiro et al., "Aequitas: A bias and fairness audit toolkit," *CoRR*, vol. abs/1811.05577, 2018. arXiv: 1811.05577.

[130] R. K. E. Bellamy et al., "AI fairness 360: An extensible toolkit for detecting and mitigating algorithmic bias," *IBM J. Res. Dev.*, vol. 63, no. 4/5, 4:1–4:15, 2019.

[131] 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 *14th IEEE Conference on Visual Analytics Science and Technology, IEEE VAST 2019, Vancouver, BC, Canada, October 20-25, 2019*, IEEE, 2019, pp. 46–56.

[132] S. Lee, A. Cho, G. C. Kim, S. Peng, M. Phute, and D. H. Chau, *Interpretation meets safety: A survey on interpretation methods and tools for improving llm safety*, 2025. arXiv: 2506.05451 [cs.SE].

[133] S. Lee et al., "Llm attributor: Interactive visual attribution for llm generation," in *Proceedings of the Thirty-Ninth AAAI Conference on Artificial Intelligence and Thirty-Seventh Conference on Innovative Applications of Artificial Intelligence and Fifteenth Symposium on Educational Advances in Artificial Intelligence*, ser. AAAI'25/IAAI'25/EAAI'25, AAAI Press, 2025, ISBN: 978-1-57735-897-8.

[134] F. Yu, L. Quartey, and F. Schilder, "Legal prompting: Teaching a language model to think like a lawyer," *arXiv preprint arXiv:2212.01326*, 2022.

[135] K. Singhal et al., "Large language models encode clinical knowledge," *Nature*, vol. 620, no. 7972, pp. 172–180, 2023.

[136] A. V. Sadybekov and V. Katritch, "Computational approaches streamlining drug discovery," *Nature*, vol. 616, no. 7958, pp. 673–685, 2023.

[137] S. Wang, Z. Zhao, X. Ouyang, T. Liu, Q. Wang, and D. Shen, "Interactive computer-aided diagnosis on medical image using large language models," *Communications Engineering*, vol. 3, no. 1, p. 133, 2024.

[138] Q. Tang and X. Li, "An investigation of large language models and their vulnerabilities in spam detection," *arXiv preprint arXiv:2504.09776*, 2025.

[139] H. Zhao et al., "Explainability for large language models: A survey," *ACM Transactions on Intelligent Systems and Technology*, vol. 15, no. 2, pp. 1–38, 2024.

[140] J. Ferrando, G. Sarti, A. Bisazza, and M. R. Costa-Jussà, "A primer on the inner workings of transformer-based language models," *arXiv preprint arXiv:2405.00208*, 2024.

[141] H. Zhao, F. Yang, B. Shen, H. Lakkaraju, and M. Du, "Towards uncovering how large language model works: An explainability perspective," *arXiv preprint arXiv:2402.10688*, 2024.

[142] N. Calderon and R. Reichart, "On behalf of the stakeholders: Trends in NLP model interpretability in the era of LLMs," in *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, L. Chiruzzo, A. Ritter, and L. Wang, Eds., Albuquerque, New Mexico: Association for Computational Linguistics, Apr. 2025, pp. 656–693, ISBN: 979-8-89176-189-6.

[143] X. Huang et al., "A survey of safety and trustworthiness of large language models through the lens of verification and validation," *Artif. Intell. Rev.*, vol. 57, p. 175, 2023.

[144] S. G. Ayyamperumal and L. Ge, "Current state of llm risks and ai guardrails," *arXiv preprint arXiv:2406.12934*, 2024.

[145] D. Shi et al., "Large language model safety: A holistic survey," *arXiv preprint arXiv:2412.17686*, 2024.

[146] J. Chua, Y. Li, S. Yang, C. Wang, and L. Yao, "Ai safety in generative ai large language models: A survey," *arXiv preprint arXiv:2407.18369*, 2024.

[147] X. Ma et al., *Safety at scale: A comprehensive survey of large model safety*, 2025. arXiv: 2502.05206 [cs.CR].

[148] X. Wu et al., *Usable xai: 10 strategies towards exploiting explainability in the llm era*, 2024. arXiv: 2403.08946 [cs.LG].

[149] L. Bereska and E. Gavves, "Mechanistic interpretability for ai safety–a review," *arXiv preprint arXiv:2404.14082*, 2024.

[150] S. Lee et al., "Llm attributor: Interactive visual attribution for llm generation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, 2025, pp. 29 655–29 657.

[151] R. Hazra, S. Layek, S. Banerjee, and S. Poria, "Safety arithmetic: A framework for test-time safety alignment of language models by steering parameters and activations," in *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, Y. Al-Onaizan, M. Bansal, and Y.-N. Chen, Eds., Miami, Florida, USA: Association for Computational Linguistics, Nov. 2024, pp. 21 759–21 776.

[152] W. Zhao et al., *Towards comprehensive post safety alignment of large language models via safety patching*, 2024. arXiv: 2405.13820 [cs.CL].

[153] A. Mishra et al., "Promptaid: Visual prompt exploration, perturbation, testing and iteration for large language models," *IEEE Transactions on Visualization and Computer Graphics*, pp. 1–14, 2025.

[154] J. Vig, "A multiscale visualization of attention in the transformer model," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, M. R. Costa-jussà and E. Alfonseca, Eds., Florence, Italy: Association for Computational Linguistics, Jul. 2019, pp. 37–42.

[155] Y. Wang, F. Weng, S. Yang, Z. Qin, M. Huang, and W. Wang, "Delman: Dynamic defense against large language model jailbreaking with model editing," *arXiv preprint arXiv:2502.11647*, 2025.

[156] D. Dale, E. Voita, L. Barrault, and M. R. Costa-jussà, "Detecting and mitigating hallucinations in machine translation: Model internal workings alone do well, sentence similarity Even better," in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, A. Rogers, J. Boyd-Graber, and N. Okazaki, Eds., Toronto, Canada: Association for Computational Linguistics, Jul. 2023, pp. 36–50.

[157] Y.-S. Chuang, L. Qiu, C.-Y. Hsieh, R. Krishna, Y. Kim, and J. R. Glass, "Lookback lens: Detecting and mitigating contextual hallucinations in large language models using only attention maps," in *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, Y. Al-Onaizan, M. Bansal, and Y.-N. Chen, Eds., Miami, Florida, USA: Association for Computational Linguistics, Nov. 2024, pp. 1419–1436.

[158] W. Pan, Z. Liu, Q. Chen, X. Zhou, H. Yu, and X. Jia, "The hidden dimensions of llm alignment: A multi-dimensional safety analysis," *arXiv preprint arXiv:2502.09674*, 2025.

[159] R. Li, R. Yang, W. Xiao, A. AbuRaed, G. Murray, and G. Carenini, *Visual analytics for generative transformer models*, 2023. arXiv: 2311.12418 [cs.CL].

[160] Q. Zhang et al., "Tell your model where to attend: Post-hoc attention steering for LLMs," in *The Twelfth International Conference on Learning Representations*, 2024.

[161] M. Zhu, L. Yang, Y. Wei, N. Zhang, and Y. Zhang, "Locking down the finetuned llms safety," *arXiv preprint arXiv:2410.10343*, 2024.

[162] H. Duan, Y. Yang, and K. Y. Tam, "Do llms know about hallucination? an empirical investigation of llm's hidden states," *arXiv preprint arXiv:2402.09733*, 2024.

[163] S. Ball, F. Kreuter, and N. Panickssery, "Understanding jailbreak success: A study of latent space dynamics in large language models," *arXiv preprint arXiv:2406.09289*, 2024.

[164] T. Li et al., "Revisiting jailbreaking for large language models: A representation engineering perspective," in *Proceedings of the 31st International Conference on Computational Linguistics*, O. Rambow, L. Wanner, M. Apidianaki, H. Al-Khalifa, B. D. Eugenio, and S. Schockaert, Eds., Abu Dhabi, UAE: Association for Computational Linguistics, Jan. 2025, pp. 3158–3178.

[165] H. Wang et al., "Model surgery: Modulating llm's behavior via simple parameter editing," *arXiv preprint arXiv:2407.08770*, 2024.

[166] J. Yang, D. Chen, Y. Sun, R. Li, Z. Feng, and W. Peng, "Enhancing semantic consistency of large language models through model editing: An interpretability-oriented approach," in *Findings of the Association for Computational Linguistics: ACL 2024*, L.-W. Ku, A. Martins, and V. Srikumar, Eds., Bangkok, Thailand: Association for Computational Linguistics, Aug. 2024, pp. 3343–3353.

[167] A. Bhattacharjee, S. Ghosh, T. Rebedea, and C. Parisien, "Towards inference-time category-wise safety steering for large language models," in *Neurips Safe Generative AI Workshop 2024*, 2024.

[168] Z. Chu, Y. Wang, L. Li, Z. Wang, Z. Qin, and K. Ren, "A causal explainable guardrails for large language models," in *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '24, Salt Lake City, UT, USA: Association for Computing Machinery, 2024, pp. 1136–1150, ISBN: 9798400706363.

[169] N. Rimsky, N. Gabrieli, J. Schulz, M. Tong, E. Hubinger, and A. Turner, "Steering llama 2 via contrastive activation addition," in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, L.-W. Ku, A. Martins, and V. Srikumar, Eds., Bangkok, Thailand: Association for Computational Linguistics, Aug. 2024, pp. 15 504–15 522.

[170] S. Singh, S. Ravfogel, J. Herzig, R. Aharoni, R. Cotterell, and P. Kumaraguru, "Representation surgery: Theory and practice of affine steering," in *Proceedings of the 41st International Conference on Machine Learning*, ser. ICML'24, Vienna, Austria: JMLR.org, 2024.

[171] K. Li, O. Patel, F. Viégas, H. Pfister, and M. Wattenberg, "Inference-time intervention: Eliciting truthful answers from a language model," *Advances in Neural Information Processing Systems*, vol. 36, pp. 41 451–41 530, 2023.

[172] A. M. Turner et al., "Steering language models with activation engineering," *arXiv preprint arXiv:2308.10248*, 2023.

[173] L. Gao, X. Zhang, P. Nakov, and X. Chen, "Shaping the safety boundaries: Understanding and defending against jailbreaks in large language models," *arXiv preprint arXiv:2412.17034*, 2024.

[174] G. Shen, D. Zhao, Y. Dong, X. He, and Y. Zeng, "Jailbreak antidote: Runtime safety-utility balance via sparse representation adjustment in large language models," in *The Thirteenth International Conference on Learning Representations*, 2025.

[175] P. Han, C. Qian, X. Chen, Y. Zhang, D. Zhang, and H. Ji, "Internal activation as the polar star for steering unsafe llm behavior," *arXiv preprint arXiv:2502.01042*, 2025.

[176] E. Hernandez, B. Z. Li, and J. Andreas, "Inspecting and editing knowledge representations in language models," in *First Conference on Language Modeling*, 2024.

[177] Y. Chen et al., *Designing a dashboard for transparency and control of conversational ai*, 2024. arXiv: 2406.07882 [cs.CL].

[178] M. Wang et al., "Detoxifying large language models via knowledge editing," in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, L.-W. Ku, A. Martins, and V. Srikumar,

Eds., Bangkok, Thailand: Association for Computational Linguistics, Aug. 2024, pp. 3093–3118.

[179] X. Li, Z. Li, Y. Kosuga, Y. Yoshida, and V. Bian, "Precision knowledge editing: Enhancing safety in large language models," *arXiv preprint arXiv:2410.03772*, 2024.

[180] Y. Chen et al., "Attributive reasoning for hallucination diagnosis of large language models," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, 2025, pp. 23 660–23 668.

[181] C. Zhao, Z. Dou, and K. Huang, "Eeg-defender: Defending against jailbreak through early exit generation of large language models," *arXiv preprint arXiv:2408.11308*, 2024.

[182] C. Burns, H. Ye, D. Klein, and J. Steinhardt, "Discovering latent knowledge in language models without supervision," *arXiv preprint arXiv:2212.03827*, 2022.

[183] T. Li, E. Dayanik, S. Tyagi, and A. Pierleoni, "Hallucana: Fixing llm hallucination with a canary lookahead," *arXiv preprint arXiv:2412.07965*, 2024.

[184] B. Deng, W. Wang, F. Zhu, Q. Wang, and F. Feng, "Cram: Credibility-aware attention modification in llms for combating misinformation in rag," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, 2025, pp. 23 760–23 768.

[185] Y. Liu et al., "The devil is in the neurons: Interpreting and mitigating social biases in language models," in *ICLR*, 2024.

[186] H. Li, H. Chi, M. Liu, and W. Yang, "Look within, why llms hallucinate: A causal perspective," *arXiv preprint arXiv:2407.10153*, 2024.

[187] W. Ma et al., "Deciphering stereotypes in pre-trained language models," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, H. Bouamor, J. Pino, and K. Bali, Eds., Singapore: Association for Computational Linguistics, Dec. 2023, pp. 11 328–11 345.

[188] S. Li, L. Yao, L. Zhang, and Y. Li, "Safety layers in aligned large language models: The key to LLM security," in *The Thirteenth International Conference on Learning Representations*, 2025.

[189] W. Zhao, Z. Li, Y. Li, Y. Zhang, and J. Sun, "Defending large language models against jailbreak attacks via layer-specific editing," in *Findings of the Association for Computational Linguistics: EMNLP 2024*, Y. Al-Onaizan, M. Bansal, and Y.-N. Chen, Eds., Miami, Florida, USA: Association for Computational Linguistics, Nov. 2024, pp. 5094–5109.

[190] E. Hernandez et al., "Linearity of relation decoding in transformer language models," in *The Twelfth International Conference on Learning Representations*, 2024.

[191] J. Lindsey et al., *On the biology of a large language model*, https://transformer-circuits.pub/2025/attribution-graphs/biology.html, Transformer Circuits Thread, Mar. 2025.

[192] E. Ameisen et al., *Circuit tracing: Revealing computational graphs in language models*, https://transformer-circuits.pub/2025/attribution-graphs/methods.html, Transformer Circuits Thread, Mar. 2025.

[193] D. Zhou, K. Patil, Y. Sun, K. lakshmanan, S. Rajamanoharan, and A. Conmy, "LLM neurosurgeon: Targeted knowledge removal in LLMs using sparse autoencoders," in *ICLR 2025 Workshop on Building Trust in Language Models and Applications*, 2025.

[194] A. Frikha, M. R. A. Razi, K. K. Nakka, R. Mendes, X. Jiang, and X. Zhou, "Privacyscalpel: Enhancing llm privacy via interpretable feature intervention with sparse autoencoders," *arXiv preprint arXiv:2503.11232*, 2025.

[195] P. Hegde, "Effectiveness of sparse autoencoder for understanding and removing gender bias in LLMs," in *NeurIPS 2024 Workshop on Scientific Methods for Understanding Deep Learning*, 2024.

[196] Z. He, Z. Wang, H. Xu, and K. Ren, "Towards llm guardrails via sparse representation steering," *arXiv preprint arXiv:2503.16851*, 2025.

[197] S. Abdaljalil, F. Pallucchini, A. Seveso, H. Kurban, F. Mercorio, and E. Serpedin, "SAFE: A sparse autoencoder-based framework for robust query enrichment and hallucination mitigation in llms," *arXiv preprint arXiv:2503.03032*, 2025.

[198] R. Bayat, A. Rahimi-Kalahroudi, M. Pezeshki, S. Chandar, and P. Vincent, "Steering large language model activations in sparse spaces," *arXiv preprint arXiv:2503.00177*, 2025.

[199] X. Wu, J. Yuan, W. Yao, X. Zhai, and N. Liu, "Interpreting and steering llms with mutual information-based explanations on sparse autoencoders," *arXiv preprint arXiv:2502.15576*, 2025.

[200] K. O'Brien et al., "Steering language model refusal with sparse autoencoders," *arXiv preprint arXiv:2411.11296*, 2024.

[201] M. Khoriaty, A. Shportko, G. Mercier, and Z. Wood-Doughty, "Don't forget it! conditional sparse autoencoder clamping works for unlearning," *arXiv preprint arXiv:2503.11127*, 2025.

[202] M. Geva, A. Caciularu, K. Wang, and Y. Goldberg, "Transformer feed-forward layers build predictions by promoting concepts in the vocabulary space," in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, Y. Goldberg, Z. Kozareva, and Y. Zhang, Eds., Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, Dec. 2022, pp. 30–45.

[203] M. Geva et al., "LM-debugger: An interactive tool for inspection and intervention in transformer-based language models," in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, W. Che and E. Shutova, Eds., Abu Dhabi, UAE: Association for Computational Linguistics, Dec. 2022, pp. 12–21.

[204] L. Yu, M. Cao, J. C. Cheung, and Y. Dong, "Mechanistic understanding and mitigation of language model non-factual hallucinations," in *Findings of the Association for Computational Linguistics: EMNLP 2024*, Y. Al-Onaizan, M. Bansal, and Y.-N. Chen, Eds., Miami, Florida, USA: Association for Computational Linguistics, Nov. 2024, pp. 7943–7956.

[205] S. Dhuliawala et al., "Chain-of-verification reduces hallucination in large language models," in *Findings of the Association for Computational Linguistics: ACL 2024*, L.-W. Ku, A. Martins, and V. Srikumar, Eds., Bangkok, Thailand: Association for Computational Linguistics, Aug. 2024, pp. 3563–3578.

[206] Y. Weng et al., "Large language models are better reasoners with self-verification," in *Findings of the Association for Computational Linguistics: EMNLP 2023*, H. Bouamor, J. Pino, and K. Bali, Eds., Singapore: Association for Computational Linguistics, Dec. 2023, pp. 2550–2575.

[207] X. Cheng, J. Li, W. X. Zhao, and J.-R. Wen, "Think more, hallucinate less: Mitigating hallucinations via dual process of fast and slow thinking," *arXiv preprint arXiv:2501.01306*, 2025.

[208] Y. Liu et al., "Guardreasoner: Towards reasoning-based llm safeguards," *arXiv preprint arXiv:2501.18492*, 2025.

[209] Y. Jiang et al., "Comt: Chain-of-medical-thought reduces hallucination in medical report generation," in *ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2025, pp. 1–5.

[210] B. Ji, H. Liu, M. Du, and S.-K. Ng, "Chain-of-thought improves text generation with citations in large language models," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, 2024, pp. 18 345–18 353.

[211] Y. Zhang, M. Li, W. Han, Y. Yao, Z. Cen, and D. Zhao, "Safety is not only about refusal: Reasoning-enhanced fine-tuning for interpretable llm safety," *arXiv preprint arXiv:2503.05021*, 2025.

[212] M. Kaneko, D. Bollegala, N. Okazaki, and T. Baldwin, "Evaluating gender bias in large language models via chain-of-thought prompting," *arXiv preprint arXiv:2401.15585*, 2024.

[213] L. Prahallad and R. Mamidi, "Significance of chain of thought in gender bias mitigation for english-dravidian machine translation," *arXiv preprint arXiv:2405.19701*, 2024.

[214] X. Li, Y. Zhang, R. Lou, C. Wu, and J. Wang, "Chain-of-scrutiny: Detecting backdoor attacks for large language models," *arXiv preprint arXiv:2406.05948*, 2024.

[215] Y. Cao, N. Gu, X. Shen, D. Yang, and X. Zhang, "Defending large language models against jailbreak attacks through chain of thought prompting," in *2024 International Conference on Networking and Network Applications (NaNA)*, 2024, pp. 125–130.

[216] M. K. Rad, H. Nghiem, A. Luo, S. Wadhwa, M. Sorower, and S. Rawls, "Refining input guardrails: Enhancing llm-as-a-judge efficiency through chain-of-thought fine-tuning and alignment," *arXiv preprint arXiv:2501.13080*, 2025.

[217] K. Moore, J. Roberts, T. Pham, and D. Fisher, "Reasoning beyond bias: A study on counterfactual prompting and chain of thought reasoning," *arXiv preprint arXiv:2408.08651*, 2024.

[218] A. Sicilia and M. Alikhani, "Eliciting uncertainty in chain-of-thought to mitigate bias against forecasting harmful user behaviors," in *Proceedings of the Third Workshop on NLP for Positive Impact*, D. Dementieva et al., Eds., Miami, Florida, USA: Association for Computational Linguistics, Nov. 2024, pp. 211–223.

[219] Y. Mou, Y. Luo, S. Zhang, and W. Ye, "Saro: Enhancing llm safety through reasoning-based alignment," *arXiv preprint arXiv:2504.09420*, 2025.

[220] B. C. Kwon and N. Mihindukulasooriya, "Finspector: A human-centered visual inspection tool for exploring and comparing biases among foundation models," in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, D. Bollegala, R. Huang, and A. Ritter, Eds., Toronto, Canada: Association for Computational Linguistics, Jul. 2023, pp. 42–50.

[221] C.-K. Yeh, J. Kim, I. E.-H. Yen, and P. K. Ravikumar, "Representer point selection for explaining deep neural networks," in *Advances in Neural Information Processing*

Systems, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds., vol. 31, Curran Associates, Inc., 2018.

[222] C.-P. Tsai, C.-K. Yeh, and P. Ravikumar, "Sample based explanations via generalized representers," in *Proceedings of the 37th International Conference on Neural Information Processing Systems*, ser. NIPS '23, New Orleans, LA, USA: Curran Associates Inc., 2023.

[223] Y. Su, J. J. Li, and M. Lease, "Wrapper boxes for faithful attribution of model predictions to training data," in *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, Y. Belinkov, N. Kim, J. Jumelet, H. Mohebbi, A. Mueller, and H. Chen, Eds., Miami, Florida, US: Association for Computational Linguistics, Nov. 2024, pp. 551–576.

[224] L. He, M. Xia, and P. Henderson, "What is in your safe data? identifying benign data that breaks safety," in *First Conference on Language Modeling*, 2024.

[225] D. Cheng, J. Bae, J. Bullock, and D. Kristofferson, "Training data attribution (tda): Examining its adoption & use cases," *arXiv preprint arXiv:2501.12642*, 2025.

[226] X. Han and Y. Tsvetkov, "Influence tuning: Demoting spurious correlations via instance attribution and instance-driven updates," in *Findings of the Association for Computational Linguistics: EMNLP 2021*, M.-F. Moens, X. Huang, L. Specia, and S. W.-t. Yih, Eds., Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 4398–4409.

[227] C.-K. Yeh, A. Taly, M. Sundararajan, F. Liu, and P. Ravikumar, "First is better than last for language data influence," *Advances in Neural Information Processing Systems*, vol. 35, pp. 32 285–32 298, 2022.

[228] Z. Wu, Y. Shu, and B. K. H. Low, "Davinz: Data valuation using deep neural networks at initialization," in *International Conference on Machine Learning*, PMLR, 2022, pp. 24 150–24 176.

[229] F. Ladhak, E. Durmus, and T. Hashimoto, "Contrastive error attribution for finetuned language models," in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, A. Rogers, J. Boyd-Graber, and N. Okazaki, Eds., Toronto, Canada: Association for Computational Linguistics, Jul. 2023, pp. 11 482–11 498.

[230] M. Xia, S. Malladi, S. Gururangan, S. Arora, and D. Chen, "LESS: Selecting influential data for targeted instruction tuning," in *International Conference on Machine Learning (ICML)*, 2024.

[231] Y. Pan, T. Shi, J. Zhao, and J. W. Ma, "Detecting and filtering unsafe training data via data attribution," *arXiv preprint arXiv:2502.11411*, 2025.

[232] Z. Hammoudeh and D. Lowd, "Training data influence analysis and estimation: A survey," *Mach. Learn.*, vol. 113, no. 5, pp. 2351–2403, Mar. 2024.

[233] F. R. Hampel, "The influence curve and its role in robust estimation," *Journal of the american statistical association*, vol. 69, no. 346, pp. 383–393, 1974.

[234] X. Han, B. C. Wallace, and Y. Tsvetkov, "Explaining black box predictions and unveiling data artifacts through influence functions," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, D. Jurafsky, J. Chai, N. Schluter, and J. Tetreault, Eds., Online: Association for Computational Linguistics, Jul. 2020, pp. 5553–5563.

[235] Z. Ren, R. Yeh, and A. Schwing, "Not all unlabeled data are equal: Learning to weight data in semi-supervised learning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 21 786–21 797, 2020.

[236] E. Barshan, M.-E. Brunet, and G. K. Dziugaite, "Relatif: Identifying explanatory training samples via relative influence," in *International Conference on Artificial Intelligence and Statistics*, PMLR, 2020, pp. 1899–1909.

[237] H. Guo, N. Rajani, P. Hase, M. Bansal, and C. Xiong, "FastIF: Scalable influence functions for efficient model interpretation and debugging," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, M.-F. Moens, X. Huang, L. Specia, and S. W.-t. Yih, Eds., Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 10 333–10 350.

[238] A. Schioppa, P. Zablotskaia, D. Vilar, and A. Sokolov, "Scaling up influence functions," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, 2022, pp. 8179–8186.

[239] Y. Kwon, E. Wu, K. Wu, and J. Zou, "Datainf: Efficiently estimating data influence in loRA-tuned LLMs and diffusion models," in *The Twelfth International Conference on Learning Representations*, 2024.

[240] S. K. Choe et al., "What is your data worth to gpt? llm-scale data valuation with influence functions," *arXiv preprint arXiv:2405.13954*, 2024.

[241] K. Wu, L. Pang, H. Shen, and X. Cheng, "Enhancing training data attribution for large language models with fitting error consideration," in *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, Y. Al-

Onaizan, M. Bansal, and Y.-N. Chen, Eds., Miami, Florida, USA: Association for Computational Linguistics, Nov. 2024, pp. 14 131–14 143.

[242] T. A. Chang, D. Rajagopal, T. Bolukbasi, L. Dixon, and I. Tenney, "Scalable influence and fact tracing for large language model pretraining," in *The Thirteenth International Conference on Learning Representations*, 2025.

[243] S. Basu, P. Pope, and S. Feizi, "Influence functions in deep learning are fragile," in *International Conference on Learning Representations*, 2021.

[244] E. Akyurek et al., "Towards tracing knowledge in language models back to the training data," in *Findings of the Association for Computational Linguistics: EMNLP 2022*, Y. Goldberg, Z. Kozareva, and Y. Zhang, Eds., Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, Dec. 2022, pp. 2429–2446.

[245] Z. Li, W. Zhao, Y. Li, and J. Sun, "Do influence functions work on large language models?" *arXiv preprint arXiv:2409.19998*, 2024.

[246] R. Jia et al., "Towards efficient data valuation based on the shapley value," in *The 22nd International Conference on Artificial Intelligence and Statistics*, PMLR, 2019, pp. 1167–1176.

[247] V. Feldman and C. Zhang, "What neural networks memorize and why: Discovering the long tail via influence estimation," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33, Curran Associates, Inc., 2020, pp. 2881–2891.

[248] J. Wang, X. Lin, R. Qiao, C.-S. Foo, and B. K. H. Low, "Helpful or harmful data? fine-tuning-free shapley attribution for explaining language model predictions," in *ICML*, 2024.

[249] J. T. Wang, P. Mittal, D. Song, and R. Jia, "Data shapley in one training run," in *The Thirteenth International Conference on Learning Representations*, 2025.

[250] R. Bommasani et al., "On the opportunities and risks of foundation models," *arXiv preprint arXiv:2108.07258*, 2021.

[251] J. Achiam et al., "Gpt-4 technical report," *arXiv preprint arXiv:2303.08774*, 2023.

[252] C. F. Park, M. Okawa, A. Lee, E. S. Lubana, and H. Tanaka, "Emergence of hidden capabilities: Exploring learning dynamics in concept space," *Advances in Neural Information Processing Systems*, vol. 37, pp. 84 698–84 729, 2024.

[253] N. Prakash, T. R. Shaham, T. Haklay, Y. Belinkov, and D. Bau, "Fine-tuning enhances existing mechanisms: A case study on entity tracking," in *ICLR*, 2024.

[254] J. Chen, X. Wang, Z. Yao, Y. Bai, L. Hou, and J. Li, *Finding safety neurons in large language models*, 2024. arXiv: 2406.14144 [cs.CL].

[255] L. Engstrom, A. Feldmann, and A. Madry, "Dsdm: Model-aware dataset selection with datamodels," in *Proceedings of the 41st International Conference on Machine Learning*, ser. ICML'24, Vienna, Austria: JMLR.org, 2024.

[256] X. Davies, M. Nadeau, N. Prakash, T. R. Shaham, and D. Bau, "Discovering variable binding circuitry with desiderata," *arXiv preprint arXiv:2307.03637*, 2023.

[257] N. Nanda, L. Chan, T. Lieberum, J. Smith, and J. Steinhardt, "Progress measures for grokking via mechanistic interpretability," in *The Eleventh International Conference on Learning Representations*, 2023.

[258] Y. Xu, Y. Wang, and H. Wang, "Tracking the feature dynamics in llm training: A mechanistic study," *arXiv preprint arXiv:2412.17626*, 2024.

[259] C. Ma et al., "Mechanistic insights: Circuit transformations across input and fine-tuning landscapes," *OpenReview*, 2024.

[260] T. Inaba, K. Inui, Y. Miyao, Y. Oseki, B. Heinzerling, and Y. Takagi, "How llms learn: Tracing internal representations with sparse autoencoders," *arXiv preprint arXiv:2503.06394*, 2025.

[261] A. Lee, X. Bai, I. Pres, M. Wattenberg, J. K. Kummerfeld, and R. Mihalcea, "A mechanistic understanding of alignment algorithms: A case study on dpo and toxicity," in *Proceedings of the 41st International Conference on Machine Learning*, ser. ICML'24, Vienna, Austria: JMLR.org, 2024.

[262] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," in *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Workshop Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2014.

[263] 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, e0130140, 2015.

[264] 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.

[265] 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.

[266] A. Shrikumar, P. Greenside, and A. Kundaje, "Learning important features through propagating activation differences," in *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ser. ICML'17, Sydney, NSW, Australia: JMLR.org, 2017, pp. 3145–3153.

[267] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic attribution for deep networks," in *International conference on machine learning*, PMLR, 2017, pp. 3319–3328.

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

[269] S. Wiegreffe and Y. Pinter, "Attention is not not explanation," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, K. Inui, J. Jiang, V. Ng, and X. Wan, Eds., Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 11–20.

[270] S. Abnar and W. Zuidema, "Quantifying attention flow in transformers," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, D. Jurafsky, J. Chai, N. Schluter, and J. Tetreault, Eds., Online: Association for Computational Linguistics, Jul. 2020, pp. 4190–4197.

[271] G. Kobayashi, T. Kuribayashi, S. Yokoi, and K. Inui, "Attention is not only a weight: Analyzing transformers with vector norms," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, B. Webber, T. Cohn, Y. He, and Y. Liu, Eds., Online: Association for Computational Linguistics, Nov. 2020, pp. 7057–7075.

[272] D. Halawi, J.-S. Denain, and J. Steinhardt, "Overthinking the truth: Understanding how language models process false demonstrations," in *The Twelfth International Conference on Learning Representations*, 2024.

[273] M. Yuksekgonul et al., "Attention satisfies: A constraint-satisfaction lens on factual errors of language models," in *The Twelfth International Conference on Learning Representations*, 2024.

[274] Y. Tu, J. Xu, and H.-W. Shen, "Keywordmap: Attention-based visual exploration for keyword analysis," in *2021 IEEE 14th Pacific Visualization Symposium (PacificVis)*, IEEE, 2021, pp. 206–215.

[275] G. Sarti, G. Chrupała, M. Nissim, and A. Bisazza, "Quantifying the plausibility of context reliance in neural machine translation," in *The Twelfth International Conference on Learning Representations*, 2024.

[276] G. Kobayashi, T. Kuribayashi, S. Yokoi, and K. Inui, "Incorporating Residual and Normalization Layers into Analysis of Masked Language Models," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, M.-F. Moens, X. Huang, L. Specia, and S. W.-t. Yih, Eds., Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 4547–4568.

[277] A. Modarressi, M. Fayyaz, Y. Yaghoobzadeh, and M. T. Pilehvar, "GlobEnc: Quantifying global token attribution by incorporating the whole encoder layer in transformers," in *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, M. Carpuat, M.-C. de Marneffe, and I. V. Meza Ruiz, Eds., Seattle, United States: Association for Computational Linguistics, Jul. 2022, pp. 258–271.

[278] J. Ferrando, G. I. Gállego, and M. R. Costa-jussà, "Measuring the mixing of contextual information in the transformer," in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, Y. Goldberg, Z. Kozareva, and Y. Zhang, Eds., Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, Dec. 2022, pp. 8698–8714.

[279] J. Ferrando, G. I. Gállego, I. Tsiamas, and M. R. Costa-jussà, "Explaining how transformers use context to build predictions," in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, A. Rogers, J. Boyd-Graber, and N. Okazaki, Eds., Toronto, Canada: Association for Computational Linguistics, Jul. 2023, pp. 5486–5513.

[280] A. Modarressi, M. Fayyaz, E. Aghazadeh, Y. Yaghoobzadeh, and M. T. Pilehvar, "DecompX: Explaining transformers decisions by propagating token decomposition," in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, A. Rogers, J. Boyd-Graber, and N. Okazaki, Eds., Toronto, Canada: Association for Computational Linguistics, Jul. 2023, pp. 2649–2664.

[281] S. Yang, S. Huang, W. Zou, J. Zhang, X. Dai, and J. Chen, "Local interpretation of transformer based on linear decomposition," in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, A. Rogers, J. Boyd-Graber, and N. Okazaki, Eds., Toronto, Canada: Association for Computational Linguistics, Jul. 2023, pp. 10 270–10 287.

[282] R. Achtibat et al., "Attnlrp: Attention-aware layer-wise relevance propagation for transformers," in *Proceedings of the 41st International Conference on Machine Learning*, ser. ICML'24, Vienna, Austria: JMLR.org, 2024.

[283] L. Song, Y. Cui, A. Luo, F. Lecue, and I. Li, "Better explain transformers by illuminating important information," in *Findings of the Association for Computational Linguistics: EACL 2024*, Y. Graham and M. Purver, Eds., St. Julian's, Malta: Association for Computational Linguistics, Mar. 2024, pp. 2048–2062.

[284] G. Kobayashi, T. Kuribayashi, S. Yokoi, and K. Inui, "Analyzing feed-forward blocks in transformers through the lens of attention maps," in *The Twelfth International Conference on Learning Representations*, 2024.

[285] L. Arras, B. Puri, P. Kahardipraja, S. Lapuschkin, and W. Samek, "A close look at decomposition-based xai-methods for transformer language models," *arXiv preprint arXiv:2502.15886*, 2025.

[286] J. Ferrando, G. I. Gállego, B. Alastruey, C. Escolano, and M. R. Costa-jussà, "Towards opening the black box of neural machine translation: Source and target interpretations of the transformer," in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, Y. Goldberg, Z. Kozareva, and Y. Zhang, Eds., Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, Dec. 2022, pp. 8756–8769.

[287] S. Abbasi, M. R. Modarres, and M. T. Pilehvar, "Normxlogit: The head-on-top never lies," *arXiv preprint arXiv:2411.16252*, 2024.

[288] B. Deiseroth, M. Deb, S. Weinbach, M. Brack, P. Schramowski, and K. Kersting, "Atman: Understanding transformer predictions through memory efficient attention manipulation," in *Proceedings of the 37th International Conference on Neural Information Processing Systems*, ser. NIPS '23, New Orleans, LA, USA: Curran Associates Inc., 2023.

[289] M. R. G. Madani, A. P. Gema, G. Sarti, Y. Zhao, P. Minervini, and A. Passerini, "Noiser: Bounded input perturbations for attributing large language models," *arXiv preprint arXiv:2504.02911*, 2025.

[290] A. Jacovi, S. Swayamdipta, S. Ravfogel, Y. Elazar, Y. Choi, and Y. Goldberg, "Contrastive explanations for model interpretability," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, M.-F. Moens, X. Huang, L. Specia, and S. W.-t. Yih, Eds., Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 1597–1611.

[291] K. Yin and G. Neubig, "Interpreting language models with contrastive explanations," in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language*

Processing, Y. Goldberg, Z. Kozareva, and Y. Zhang, Eds., Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, Dec. 2022, pp. 184–198.

[292] H. Mohebbi, W. Zuidema, G. Chrupała, and A. Alishahi, "Quantifying context mixing in transformers," in *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, A. Vlachos and I. Augenstein, Eds., Dubrovnik, Croatia: Association for Computational Linguistics, May 2023, pp. 3378–3400.

[293] B. Cohen-Wang, H. Shah, K. Georgiev, and A. Madry, "Contextcite: Attributing model generation to context," *Advances in Neural Information Processing Systems*, vol. 37, pp. 95 764–95 807, 2024.

[294] M. Finlayson, A. Mueller, S. Gehrmann, S. Shieber, T. Linzen, and Y. Belinkov, "Causal analysis of syntactic agreement mechanisms in neural language 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)*, C. Zong, F. Xia, W. Li, and R. Navigli, Eds., Online: Association for Computational Linguistics, Aug. 2021, pp. 1828–1843.

[295] F. Liu, P. Xu, Z. Li, Y. Feng, and H. Song, "Towards understanding in-context learning with contrastive demonstrations and saliency maps," *arXiv preprint arXiv:2307.05052*, 2023.

[296] N. G. Sadr, S. Madhusudan, and A. Emami, "Think or step-by-step? unzipping the black box in zero-shot prompts," *arXiv preprint arXiv:2502.03418*, 2025.

[297] A. Bhattacharjee, R. Moraffah, J. Garland, and H. Liu, "Towards llm-guided causal explainability for black-box text classifiers," *arXiv preprint arXiv:2309.13340*, 2023.

[298] G. Yona, O. Honovich, I. Laish, and R. Aharoni, "Surfacing biases in large language models using contrastive input decoding," *arXiv preprint arXiv:2305.07378*, 2023.

[299] Y. O. Gat, N. Calderon, A. Feder, A. Chapanin, A. Sharma, and R. Reichart, "Faithful explanations of black-box NLP models using LLM-generated counterfactuals," in *The Twelfth International Conference on Learning Representations*, 2024.

[300] I. Covert, S. Lundberg, and S.-I. Lee, "Explaining by removing: A unified framework for model explanation," *Journal of Machine Learning Research*, vol. 22, no. 209, pp. 1–90, 2021.

[301] M. Horovicz and R. Goldshmidt, "TokenSHAP: Interpreting large language models with Monte Carlo shapley value estimation," in *Proceedings of the 1st Workshop on NLP for Science (NLP4Science)*, L. Peled-Cohen, N. Calderon, S. Lissak, and

R. Reichart, Eds., Miami, FL, USA: Association for Computational Linguistics, Nov. 2024, pp. 1–8.

[302] B. Mohammadi, "Explaining large language models decisions using shapley values," *SSRN*, 2024.

[303] J. Enouen, H. Nakhost, S. Ebrahimi, S. Arik, Y. Liu, and T. Pfister, "TextGenSHAP: Scalable post-hoc explanations in text generation with long documents," in *Findings of the Association for Computational Linguistics: ACL 2024*, L.-W. Ku, A. Martins, and V. Srikumar, Eds., Bangkok, Thailand: Association for Computational Linguistics, Aug. 2024, pp. 13 984–14 011.

[304] J. Enguehard, "Sequential integrated gradients: A simple but effective method for explaining language models," in *Findings of the Association for Computational Linguistics: ACL 2023*, A. Rogers, J. Boyd-Graber, and N. Okazaki, Eds., Toronto, Canada: Association for Computational Linguistics, Jul. 2023, pp. 7555–7565.

[305] Y. Wang, T. Zhang, X. Guo, and Z. Shen, "Gradient based feature attribution in explainable ai: A technical review," *arXiv preprint arXiv:2403.10415*, 2024.

[306] O. Eberle, I. Chalkidis, L. Cabello, and S. Brandl, "Rather a nurse than a physician - contrastive explanations under investigation," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, H. Bouamor, J. Pino, and K. Bali, Eds., Singapore: Association for Computational Linguistics, Dec. 2023, pp. 6907–6920.

[307] O. Barkan, Y. Elisha, Y. Toib, J. Weill, and N. Koenigstein, "Improving LLM attributions with randomized path-integration," in *Findings of the Association for Computational Linguistics: EMNLP 2024*, Y. Al-Onaizan, M. Bansal, and Y.-N. Chen, Eds., Miami, Florida, USA: Association for Computational Linguistics, Nov. 2024, pp. 9430–9446.

[308] F. Rezaei Jafari, G. Montavon, K.-R. Müller, and O. Eberle, "Mambalrp: Explaining selective state space sequence models," *Advances in Neural Information Processing Systems*, vol. 37, pp. 118 540–118 570, 2024.

[309] J. Qi, G. Sarti, R. Fernández, and A. Bisazza, "Model internals-based answer attribution for trustworthy retrieval-augmented generation," in *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, Y. Al-Onaizan, M. Bansal, and Y.-N. Chen, Eds., Miami, Florida, USA: Association for Computational Linguistics, Nov. 2024, pp. 6037–6053.

[310] A. Bhattacharjee, R. Moraffah, J. Garland, and H. Liu, "Llms as counterfactual explanation modules: Can chatgpt explain blackbox text classifiers," *arXiv preprint arXiv:2309.13340*, vol. 4, 2023.

[311] S. Huang, S. Mamidanna, S. Jangam, Y. Zhou, and L. H. Gilpin, "Can large language models explain themselves? a study of llm-generated self-explanations," *arXiv preprint arXiv:2310.11207*, 2023.

[312] Y. Zhou and J. Shah, "The solvability of interpretability evaluation metrics," in *Findings of the Association for Computational Linguistics: EACL 2023*, A. Vlachos and I. Augenstein, Eds., Dubrovnik, Croatia: Association for Computational Linguistics, May 2023, pp. 2399–2415.

[313] O. Barkan, Y. Toib, Y. Elisha, J. Weill, and N. Koenigstein, "LLM explainability via attributive masking learning," in *Findings of the Association for Computational Linguistics: EMNLP 2024*, Y. Al-Onaizan, M. Bansal, and Y.-N. Chen, Eds., Miami, Florida, USA: Association for Computational Linguistics, Nov. 2024, pp. 9522–9537.

[314] O. Barkan, Y. Toib, Y. Elisha, and N. Koenigstein, "A learning-based approach for explaining language models," in *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, ser. CIKM '24, Boise, ID, USA: Association for Computing Machinery, 2024, pp. 98–108, ISBN: 9798400704369.

[315] A. Zou et al., "Representation engineering: A top-down approach to ai transparency," *arXiv preprint arXiv:2310.01405*, 2023.

[316] M. Geva, J. Bastings, K. Filippova, and A. Globerson, "Dissecting recall of factual associations in auto-regressive language models," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, H. Bouamor, J. Pino, and K. Bali, Eds., Singapore: Association for Computational Linguistics, Dec. 2023, pp. 12 216–12 235.

[317] M. Geva, R. Schuster, J. Berant, and O. Levy, "Transformer feed-forward layers are key-value memories," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, M.-F. Moens, X. Huang, L. Specia, and S. W.-t. Yih, Eds., Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 5484–5495.

[318] T. Mikolov, W.-t. Yih, and G. Zweig, "Linguistic regularities in continuous space word representations," in *Proceedings of the 2013 conference of the north american chapter of the association for computational linguistics: Human language technologies*, 2013, pp. 746–751.

[319] N. Elhage et al., "Toy models of superposition," *arXiv preprint arXiv:2209.10652*, 2022.

[320] K. Park, Y. J. Choe, and V. Veitch, "The linear representation hypothesis and the geometry of large language models," in *Proceedings of the 41st International Conference on Machine Learning*, ser. ICML'24, Vienna, Austria: JMLR.org, 2024.

[321] J. Liu, S. Chen, Y. Cheng, and J. He, "On the universal truthfulness hyperplane inside LLMs," in *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, Y. Al-Onaizan, M. Bansal, and Y.-N. Chen, Eds., Miami, Florida, USA: Association for Computational Linguistics, Nov. 2024, pp. 18 199–18 224.

[322] A. Arditi et al., "Refusal in language models is mediated by a single direction," *arXiv preprint arXiv:2406.11717*, 2024.

[323] Y. Lin et al., "Towards understanding jailbreak attacks in LLMs: A representation space analysis," in *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, Y. Al-Onaizan, M. Bansal, and Y.-N. Chen, Eds., Miami, Florida, USA: Association for Computational Linguistics, Nov. 2024, pp. 7067–7085.

[324] G. Alain and Y. Bengio, "Understanding intermediate layers using linear classifier probes," *arXiv preprint arXiv:1610.01644*, 2016.

[325] I. Tenney et al., "What do you learn from context? probing for sentence structure in contextualized word representations," *arXiv preprint arXiv:1905.06316*, 2019.

[326] F. Dalvi, N. Durrani, H. Sajjad, Y. Belinkov, A. Bau, and J. Glass, "What is one grain of sand in the desert? analyzing individual neurons in deep nlp models," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 6309–6317.

[327] S. Kadavath et al., "Language models (mostly) know what they know," *arXiv preprint arXiv:2207.05221*, 2022.

[328] W. Gurnee, N. Nanda, M. Pauly, K. Harvey, D. Troitskii, and D. Bertsimas, "Finding neurons in a haystack: Case studies with sparse probing," *arXiv preprint arXiv:2305.01610*, 2023.

[329] T. Ju, W. Sun, W. Du, X. Yuan, Z. Ren, and G. Liu, "How large language models encode context knowledge? a layer-wise probing study," *arXiv preprint arXiv:2402.16061*, 2024.

[330] A. Slobodkin, O. Goldman, A. Caciularu, I. Dagan, and S. Ravfogel, "The curious case of hallucinatory (un)answerability: Finding truths in the hidden states of over-confident large language models," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, H. Bouamor, J. Pino, and

K. Bali, Eds., Singapore: Association for Computational Linguistics, Dec. 2023, pp. 3607–3625.

[331] H. Orgad et al., "Llms know more than they show: On the intrinsic representation of llm hallucinations," *arXiv preprint arXiv:2410.02707*, 2024.

[332] D. Ashok and J. May, "Language models can predict their own behavior," *arXiv preprint arXiv:2502.13329*, 2025.

[333] Z. Zhou, H. Yu, X. Zhang, R. Xu, F. Huang, and Y. Li, "How alignment and jailbreak work: Explain LLM safety through intermediate hidden states," in *Findings of the Association for Computational Linguistics: EMNLP 2024*, Y. Al-Onaizan, M. Bansal, and Y.-N. Chen, Eds., Miami, Florida, USA: Association for Computational Linguistics, Nov. 2024, pp. 2461–2488.

[334] Z. Xu, R. HUANG, C. Chen, and X. Wang, "Uncovering safety risks of large language models through concept activation vector," in *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[335] S. Abdelnabi, A. Fay, G. Cherubin, A. Salem, M. Fritz, and A. Paverd, "Are you still on track!? catching llm task drift with activations," *arXiv preprint arXiv:2406.00799*, 2024.

[336] F. Hildebrandt, A. Maier, P. Krauss, and A. Schilling, "Refusal behavior in large language models: A nonlinear perspective," *arXiv preprint arXiv:2501.08145*, 2025.

[337] Z. Ji et al., "LLM internal states reveal hallucination risk faced with a query," in *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, Y. Belinkov, N. Kim, J. Jumelet, H. Mohebbi, A. Mueller, and H. Chen, Eds., Miami, Florida, US: Association for Computational Linguistics, Nov. 2024, pp. 88–104.

[338] F. Zhang, P. Yu, B. Yi, B. Zhang, T. Li, and Z. Liu, "Prompt-guided internal states for hallucination detection of large language models," *arXiv preprint arXiv:2411.04847*, 2024.

[339] L. Bürger, F. A. Hamprecht, and B. Nadler, "Truth is universal: Robust detection of lies in LLMs," in *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[340] Z. He et al., "Jailbreaklens: Interpreting jailbreak mechanism in the lens of representation and circuit," *arXiv preprint arXiv:2411.11114*, 2024.

[341] C. Li, B. Xing, D. Huo, Q. Zhou, Z. Xu, and Y. Wang, "Mixhd: A method for detecting hallucinations based on the internal state and output probability of large

language models," in *ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2025, pp. 1–5.

[342] X. Tan, H. Luan, M. Luo, X. Sun, P. Chen, and J. Dai, "Revprag: Revealing poisoning attacks in retrieval-augmented generation through llm activation analysis," *arXiv preprint arXiv:2411.18948*, 2025.

[343] J. He, Y. Gong, Z. Lin, C. Wei, Y. Zhao, and K. Chen, "LLM factoscope: Uncovering LLMs' factual discernment through measuring inner states," in *Findings of the Association for Computational Linguistics: ACL 2024*, L.-W. Ku, A. Martins, and V. Srikumar, Eds., Bangkok, Thailand: Association for Computational Linguistics, Aug. 2024, pp. 10 218–10 230.

[344] M. Beigi et al., "Internalinspector I^2: Robust confidence estimation in llms through internal states," *arXiv preprint arXiv:2406.12053*, 2024.

[345] Y. Belinkov, "Probing classifiers: Promises, shortcomings, and advances," *Computational Linguistics*, vol. 48, no. 1, pp. 207–219, 2022.

[346] S. CH-Wang, B. Van Durme, J. Eisner, and C. Kedzie, "Do androids know they're only dreaming of electric sheep?" In *Findings of the Association for Computational Linguistics: ACL 2024*, L.-W. Ku, A. Martins, and V. Srikumar, Eds., Bangkok, Thailand: Association for Computational Linguistics, Aug. 2024, pp. 4401–4420.

[347] B. A. Levinstein and D. A. Herrmann, "Still no lie detector for language models: Probing empirical and conceptual roadblocks," *Philosophical Studies*, pp. 1–27, 2024.

[348] Z. Jin et al., "Cutting off the head ends the conflict: A mechanism for interpreting and mitigating knowledge conflicts in language models," in *Findings of the Association for Computational Linguistics: ACL 2024*, L.-W. Ku, A. Martins, and V. Srikumar, Eds., Bangkok, Thailand: Association for Computational Linguistics, Aug. 2024, pp. 1193–1215.

[349] 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.

[350] C. Olsson et al., *In-context learning and induction heads*, https://transformer-circuits.pub/2022/in-context-learning-and-induction-heads/index.html, Mar. 2022.

[351] B. Wei et al., "Assessing the brittleness of safety alignment via pruning and low-rank modifications," in *Proceedings of the 41st International Conference on Machine Learning*, ser. ICML'24, Vienna, Austria: JMLR.org, 2024.

[352] Y. Yang, H. Duan, A. Abbasi, J. P. Lalor, and K. Y. Tam, "Bias a-head? analyzing bias in transformer-based language model attention heads," *arXiv preprint arXiv:2311.10395*, 2023.

[353] N. Luick, "Universal response and emergence of induction in llms," *arXiv preprint arXiv:2411.07071*, 2024.

[354] Z. Zhou et al., "On the role of attention heads in large language model safety," in *The Thirteenth International Conference on Learning Representations*, 2025.

[355] S. Y. Peng, P.-Y. Chen, M. Hull, and D. H. Chau, "Navigating the safety landscape: Measuring risks in finetuning large language models," *Advances in Neural Information Processing Systems*, vol. 37, pp. 95 692–95 715, 2024.

[356] T. Huang, S. Hu, F. Ilhan, S. F. Tekin, and L. Liu, "Harmful fine-tuning attacks and defenses for large language models: A survey," *arXiv preprint arXiv:2409.18169*, 2024.

[357] C. T. Leong, Y. Cheng, K. Xu, J. Wang, H. Wang, and W. Li, "No two devils alike: Unveiling distinct mechanisms of fine-tuning attacks," *arXiv preprint arXiv:2405.16229*, 2024.

[358] J. Pearl, "Direct and indirect effects," in *Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence*, 2001, pp. 411–420.

[359] J. Vig et al., "Investigating gender bias in language models using causal mediation analysis," *Advances in neural information processing systems*, vol. 33, pp. 12 388–12 401, 2020.

[360] G. Monea et al., "A glitch in the matrix? locating and detecting language model grounding with fakepedia," in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, L.-W. Ku, A. Martins, and V. Srikumar, Eds., Bangkok, Thailand: Association for Computational Linguistics, Aug. 2024, pp. 6828–6844.

[361] A. Geiger, H. Lu, T. Icard, and C. Potts, "Causal abstractions of neural networks," *Advances in Neural Information Processing Systems*, vol. 34, pp. 9574–9586, 2021.

[362] A. Stolfo, Y. Belinkov, and M. Sachan, "A mechanistic interpretation of arithmetic reasoning in language models using causal mediation analysis," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, H. Bouamor, J. Pino, and K. Bali, Eds., Singapore: Association for Computational Linguistics, Dec. 2023, pp. 7035–7052.

[363] V. Cabannes, C. Arnal, W. Bouaziz, X. Yang, F. Charton, and J. Kempe, "Iteration head: A mechanistic study of chain-of-thought," *Advances in Neural Information Processing Systems*, vol. 37, pp. 109 101–109 122, 2024.

[364] K. Meng, D. Bau, A. J. Andonian, and Y. Belinkov, "Locating and editing factual associations in GPT," in *Advances in Neural Information Processing Systems*, A. H. Oh, A. Agarwal, D. Belgrave, and K. Cho, Eds., 2022.

[365] N. Nanda, S. Rajamanoharan, J. Kramár, and R. Shah, *Fact finding: Attempting to reverse-engineer factual recall on the neuron level (post 1)*, https://www.lesswrong.com/posts/iGuwZTHWb6DFY3sKB/fact-finding-attempting-to-reverse-engineer-factual-recall, LessWrong, Dec. 2023.

[366] A. Ghandeharioun, A. Caciularu, A. Pearce, L. Dixon, and M. Geva, "Patchscopes: A unifying framework for inspecting hidden representations of language models," in *Proceedings of the 41st International Conference on Machine Learning*, ser. ICML'24, Vienna, Austria: JMLR.org, 2024.

[367] N. Elhage et al., "A mathematical framework for transformer circuits," *Transformer Circuits Thread*, vol. 1, no. 1, p. 12, 2021.

[368] K. R. Wang, A. Variengien, A. Conmy, B. Shlegeris, and J. Steinhardt, "Interpretability in the wild: A circuit for indirect object identification in GPT-2 small," in *The Eleventh International Conference on Learning Representations*, 2023.

[369] N. Goldowsky-Dill, C. MacLeod, L. Sato, and A. Arora, "Localizing model behavior with path patching," *arXiv preprint arXiv:2304.05969*, 2023.

[370] M. Hanna, O. Liu, and A. Variengien, "How does gpt-2 compute greater-than? interpreting mathematical abilities in a pre-trained language model," in *Proceedings of the 37th International Conference on Neural Information Processing Systems*, ser. NIPS '23, New Orleans, LA, USA: Curran Associates Inc., 2023.

[371] A. Conmy, A. Mavor-Parker, A. Lynch, S. Heimersheim, and A. Garriga-Alonso, "Towards automated circuit discovery for mechanistic interpretability," in *Advances in Neural Information Processing Systems*, A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, Eds., vol. 36, Curran Associates, Inc., 2023, pp. 16 318–16 352.

[372] J. Ferrando and E. Voita, "Information flow routes: Automatically interpreting language models at scale," in *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, Y. Al-Onaizan, M. Bansal, and Y.-N. Chen, Eds., Miami, Florida, USA: Association for Computational Linguistics, Nov. 2024, pp. 17 432–17 445.

[373] A. Bhaskar, A. Wettig, D. Friedman, and D. Chen, "Finding transformer circuits with edge pruning," *Advances in Neural Information Processing Systems*, vol. 37, pp. 18 506–18 534, 2024.

[374] N. Nanda, *Attribution patching: Activation patching at industrial scale*, https://www.neelnanda.io/mechanistic-interpretability/attribution-patching, Neel Nanda, Feb. 2024.

[375] A. Syed, C. Rager, and A. Conmy, "Attribution patching outperforms automated circuit discovery," in *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, Y. Belinkov, N. Kim, J. Jumelet, H. Mohebbi, A. Mueller, and H. Chen, Eds., Miami, Florida, US: Association for Computational Linguistics, Nov. 2024, pp. 407–416.

[376] J. Kramár, T. Lieberum, R. Shah, and N. Nanda, "Atp*: An efficient and scalable method for localizing llm behaviour to components," *arXiv preprint arXiv:2403.00745*, 2024.

[377] M. Hanna, S. Pezzelle, and Y. Belinkov, "Have faith in faithfulness: Going beyond circuit overlap when finding model mechanisms," *arXiv preprint arXiv:2403.17806*, 2024.

[378] A. Foote, N. Nanda, E. Kran, I. Konstas, S. Cohen, and F. Barez, *Neuron to graph: Interpreting language model neurons at scale*, 2023. arXiv: 2305.19911 [cs.LG].

[379] S. Arora, Y. Li, Y. Liang, T. Ma, and A. Risteski, "Linear algebraic structure of word senses, with applications to polysemy," *Transactions of the Association for Computational Linguistics*, vol. 6, L. Lee, M. Johnson, K. Toutanova, and B. Roark, Eds., pp. 483–495, 2018.

[380] T. Bricken et al., *Towards monosemanticity: Decomposing language models with dictionary learning*, 2023.

[381] A. Templeton et al., *Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet*, https://transformer-circuits.pub/2024/scaling-monosemanticity/index.html, Transformer Circuits Thread, May 2024.

[382] J. Mairal, J. Ponce, G. Sapiro, A. Zisserman, and F. Bach, "Supervised dictionary learning," *Advances in neural information processing systems*, vol. 21, 2008.

[383] A. Makhzani and B. Frey, "K-sparse autoencoders," *arXiv preprint arXiv:1312.5663*, 2013.

[384] L. Sharkey, D. Braun, and B. Millidge, "Taking features out of superposition with sparse autoencoders," in *AI Alignment Forum*, vol. 8, 2022, pp. 15–16.

[385] R. Huben, H. Cunningham, L. R. Smith, A. Ewart, and L. Sharkey, "Sparse autoencoders find highly interpretable features in language models," in *The Twelfth International Conference on Learning Representations*, 2024.

[386] T. Lieberum et al., "Gemma scope: Open sparse autoencoders everywhere all at once on gemma 2," *arXiv preprint arXiv:2408.05147*, 2024.

[387] G. Paulo, A. Mallen, C. Juang, and N. Belrose, "Automatically interpreting millions of features in large language models," *arXiv preprint arXiv:2410.13928*, 2024.

[388] S. Rajamanoharan et al., "Improving sparse decomposition of language model activations with gated sparse autoencoders," in *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[389] J. Dunefsky, P. Chlenski, and N. Nanda, *Transcoders enable fine-grained interpretable circuit analysis for language models*, https://www.lesswrong.com/posts/YmkjnWtZGLbHRbzrP/transcoders-enable-fine-grained-interpretable-circuit, LessWrong, Apr. 2024.

[390] A. Mudide, J. Engels, E. J. Michaud, M. Tegmark, and C. S. de Witt, "Efficient dictionary learning with switch sparse autoencoders," in *The Thirteenth International Conference on Learning Representations*, 2025.

[391] S. Rajamanoharan et al., "Jumping ahead: Improving reconstruction fidelity with jumprelu sparse autoencoders," *arXiv preprint arXiv:2407.14435*, 2024.

[392] C. Kissane, R. Krzyzanowski, J. I. Bloom, A. Conmy, and N. Nanda, "Interpreting attention layer outputs with sparse autoencoders," in *ICML 2024 Workshop on Mechanistic Interpretability*, 2024.

[393] D. Ghilardi, F. Belotti, and M. Molinari, "Efficient training of sparse autoencoders for large language models via layer groups," *arXiv preprint arXiv:2410.21508*, 2024.

[394] D. Braun, J. Taylor, N. Goldowsky-Dill, and L. Sharkey, "Identifying functionally important features with end-to-end sparse dictionary learning," *Advances in Neural Information Processing Systems*, vol. 37, pp. 107 286–107 325, 2024.

[395] W. Shi et al., "Route sparse autoencoder to interpret large language models," *arXiv preprint arXiv:2503.08200*, 2025.

[396] L. Farnik, T. Lawson, C. Houghton, and L. Aitchison, "Jacobian sparse autoencoders: Sparsify computations, not just activations," *arXiv preprint arXiv:2502.18147*, 2025.

[397] C. O'Neill, C. Ye, K. Iyer, and J. F. Wu, "Disentangling dense embeddings with sparse autoencoders," *arXiv preprint arXiv:2408.00657*, 2024.

[398] Z. He et al., "Llama scope: Extracting millions of features from llama-3.1-8b with sparse autoencoders," *arXiv preprint arXiv:2410.20526*, 2024.

[399] J. Ferrando, O. B. Obeso, S. Rajamanoharan, and N. Nanda, "Do i know this entity? knowledge awareness and hallucinations in language models," in *The Thirteenth International Conference on Learning Representations*, 2025.

[400] J. Theodorus et al., "Finding sparse autoencoder representations of errors in cot prompting," in *ICLR 2025 Workshop on Building Trust in Language Models and Applications*, 2025.

[401] R. Härle, F. Friedrich, M. Brack, B. Deiseroth, P. Schramowski, and K. Kersting, *Scar: Sparse conditioned autoencoders for concept detection and steering in llms*, 2024. arXiv: 2411.07122.

[402] A. Muhamed, M. T. Diab, and V. Smith, "Decoding dark matter: Specialized sparse autoencoders for interpreting rare concepts in foundation models," in *Findings of the Association for Computational Linguistics: NAACL 2025*, L. Chiruzzo, A. Ritter, and L. Wang, Eds., Albuquerque, New Mexico: Association for Computational Linguistics, Apr. 2025, pp. 1604–1635, ISBN: 979-8-89176-195-7.

[403] J. Gallifant, S. Chen, K. Sasse, H. Aerts, T. Hartvigsen, and D. S. Bitterman, "Sparse autoencoder features for classifications and transferability," *arXiv preprint arXiv:2502.11367*, 2025.

[404] Z. He, X. Ge, Q. Tang, T. Sun, Q. Cheng, and X. Qiu, "Dictionary learning improves patch-free circuit discovery in mechanistic interpretability: A case study on othello-gpt," *arXiv preprint arXiv:2402.12201*, 2024.

[405] J. Dunefsky, P. Chlenski, S. Rajamanoharan, and N. Nanda, *Case studies in reverse-engineering sparse autoencoder features by using mlp linearization*, https://www.lesswrong.com/posts/93nKtsDL6YY5fRbQv/case-studies-in-reverse-engineering-sparse-autoencoder, LessWrong, Jan. 2024.

[406] S. Marks, C. Rager, E. J. Michaud, Y. Belinkov, D. Bau, and A. Mueller, "Sparse feature circuits: Discovering and editing interpretable causal graphs in language models," in *The Thirteenth International Conference on Learning Representations*, 2025.

[407] N. Balagansky, I. Maksimov, and D. Gavrilov, "Mechanistic permutability: Match features across layers," in *The Thirteenth International Conference on Learning Representations*, 2025.

[408] J. Lindsey, A. Templeton, J. Marcus, T. Conerly, J. Batson, and C. Olah, *Sparse crosscoders for cross-layer features and model diffing*, https://transformer-circuits.pub/2024/crosscoders/index.html, Transformer Circuits Thread, Oct. 2024.

[409] J. Dunefsky, P. Chlenski, and N. Nanda, "Transcoders find interpretable LLM feature circuits," in *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[410] nostalgebraist, *Interpreting gpt: The logit lens*, https://www.lesswrong.com/posts/AcKRB8wDpdaN6v6ru/interpreting-gpt-the-logit-lens, LessWrong, Aug. 2020.

[411] G. Dar, M. Geva, A. Gupta, and J. Berant, "Analyzing transformers in embedding space," in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, A. Rogers, J. Boyd-Graber, and N. Okazaki, Eds., Toronto, Canada: Association for Computational Linguistics, Jul. 2023, pp. 16 124–16 170.

[412] N. Belrose et al., "Eliciting latent predictions from transformers with the tuned lens," *arXiv preprint arXiv:2303.08112*, 2023.

[413] A. Y. Din, T. Karidi, L. Choshen, and M. Geva, "Jump to conclusions: Short-cutting transformers with linear transformations," *arXiv preprint arXiv:2303.09435*, 2023.

[414] S. Katz, Y. Belinkov, M. Geva, and L. Wolf, "Backward lens: Projecting language model gradients into the vocabulary space," in *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, Y. Al-Onaizan, M. Bansal, and Y.-N. Chen, Eds., Miami, Florida, USA: Association for Computational Linguistics, Nov. 2024, pp. 2390–2422.

[415] A. Haviv, I. Cohen, J. Gidron, R. Schuster, Y. Goldberg, and M. Geva, "Understanding transformer memorization recall through idioms," in *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, A. Vlachos and I. Augenstein, Eds., Dubrovnik, Croatia: Association for Computational Linguistics, May 2023, pp. 248–264.

[416] Q. Yu, J. Merullo, and E. Pavlick, "Characterizing mechanisms for factual recall in language models," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, H. Bouamor, J. Pino, and K. Bali, Eds., Singapore: Association for Computational Linguistics, Dec. 2023, pp. 9924–9959.

[417] C. Jiang et al., "On large language models' hallucination with regard to known facts," in *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, K. Duh, H. Gomez, and S. Bethard, Eds., Mexico City, Mexico: Association for Computational Linguistics, Jun. 2024, pp. 1041–1053.

[418] Z. Feng, H. Zhou, Z. ZHU, J. Qian, and K. Mao, "Unveiling and manipulating prompt influence in large language models," in *The Twelfth International Conference on Learning Representations*, 2024.

[419] J. Huang and K. C.-C. Chang, "Towards reasoning in large language models: A survey," in *Findings of the Association for Computational Linguistics: ACL 2023*, A. Rogers, J. Boyd-Graber, and N. Okazaki, Eds., Toronto, Canada: Association for Computational Linguistics, Jul. 2023, pp. 1049–1065.

[420] F. Yu, H. Zhang, P. Tiwari, and B. Wang, "Natural language reasoning, a survey," *ACM Comput. Surv.*, vol. 56, no. 12, Oct. 2024.

[421] O.-M. Camburu, T. Rocktäschel, T. Lukasiewicz, and P. Blunsom, "E-snli: Natural language inference with natural language explanations," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[422] N. F. Rajani, B. McCann, C. Xiong, and R. Socher, "Explain yourself! leveraging language models for commonsense reasoning," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, A. Korhonen, D. Traum, and L. Màrquez, Eds., Florence, Italy: Association for Computational Linguistics, Jul. 2019, pp. 4932–4942.

[423] A. Marasovic, I. Beltagy, D. Downey, and M. Peters, "Few-shot self-rationalization with natural language prompts," in *Findings of the Association for Computational Linguistics: NAACL 2022*, M. Carpuat, M.-C. de Marneffe, and I. V. Meza Ruiz, Eds., Seattle, United States: Association for Computational Linguistics, Jul. 2022, pp. 410–424.

[424] A. Chaudhry, S. Thiagarajan, and D. Gorur, "Finetuning language models to emit linguistic expressions of uncertainty," *arXiv preprint arXiv:2409.12180*, 2024.

[425] A. Amayuelas, K. Wong, L. Pan, W. Chen, and W. Y. Wang, "Knowledge of knowledge: Exploring known-unknowns uncertainty with large language models," in *Findings of the Association for Computational Linguistics: ACL 2024*, L.-W. Ku, A. Martins, and V. Srikumar, Eds., Bangkok, Thailand: Association for Computational Linguistics, Aug. 2024, pp. 6416–6432.

[426] T. Xu et al., "SaySelf: Teaching LLMs to express confidence with self-reflective rationales," in *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, Y. Al-Onaizan, M. Bansal, and Y.-N. Chen, Eds., Miami, Florida, USA: Association for Computational Linguistics, Nov. 2024, pp. 5985–5998.

[427] J. Wei et al., "Chain-of-thought prompting elicits reasoning in large language models," in *Proceedings of the 36th International Conference on Neural Information*

Processing Systems, ser. NIPS '22, New Orleans, LA, USA: Curran Associates Inc., 2022, ISBN: 9781713871088.

[428] J. Zhao, Z. Yao, zhichao Yang, and hong yu, "SELF-EXPLAIN: Teaching large language models to reason complex questions by themselves," in *R0-FoMo:Robustness of Few-shot and Zero-shot Learning in Large Foundation Models*, 2023.

[429] X. Chu, Z. Tan, H. Xue, G. Wang, T. Mo, and W. Li, "Domaino1s: Guiding llm reasoning for explainable answers in high-stakes domains," *arXiv preprint arXiv:2501.14431*, 2025.

[430] V. Cahlik, R. Alves, and P. Kordik, "Reasoning-grounded natural language explanations for language models," *arXiv preprint arXiv:2503.11248*, 2025.

[431] S. Yao et al., "Tree of thoughts: Deliberate problem solving with large language models," in *Proceedings of the 37th International Conference on Neural Information Processing Systems*, ser. NIPS '23, New Orleans, LA, USA: Curran Associates Inc., 2023.

[432] M. Besta et al., "Graph of thoughts: Solving elaborate problems with large language models," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, 2024, pp. 17 682–17 690.

[433] H. Qu, Y. Cao, J. Gao, L. Ding, and R. Xu, "Interpretable proof generation via iterative backward reasoning," in *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, M. Carpuat, M.-C. de Marneffe, and I. V. Meza Ruiz, Eds., Seattle, United States: Association for Computational Linguistics, Jul. 2022, pp. 2968–2981.

[434] Q. Lyu et al., "Faithful chain-of-thought reasoning," in *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, J. C. Park et al., Eds., Nusa Dua, Bali: Association for Computational Linguistics, Nov. 2023, pp. 305–329.

[435] O. Tafjord, B. Dalvi Mishra, and P. Clark, "Entailer: Answering questions with faithful and truthful chains of reasoning," in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, Y. Goldberg, Z. Kozareva, and Y. Zhang, Eds., Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, Dec. 2022, pp. 2078–2093.

[436] A. Creswell and M. Shanahan, "Faithful reasoning using large language models," *arXiv preprint arXiv:2208.14271*, 2022.

[437] A. Creswell, M. Shanahan, and I. Higgins, "Selection-inference: Exploiting large language models for interpretable logical reasoning," in *The Eleventh International Conference on Learning Representations*, 2023.

[438] J. Gao, X. Ding, B. Qin, and T. Liu, "Is ChatGPT a good causal reasoner? a comprehensive evaluation," in *Findings of the Association for Computational Linguistics: EMNLP 2023*, H. Bouamor, J. Pino, and K. Bali, Eds., Singapore: Association for Computational Linguistics, Dec. 2023, pp. 11 111–11 126.

[439] X. Ye and G. Durrett, "The unreliability of explanations in few-shot prompting for textual reasoning," in *Proceedings of the 36th International Conference on Neural Information Processing Systems*, ser. NIPS '22, New Orleans, LA, USA: Curran Associates Inc., 2022, ISBN: 9781713871088.

[440] R. Araya, "Do chains-of-thoughts of large language models suffer from hallucinations, cognitive biases, or phobias in bayesian reasoning?" *arXiv preprint arXiv:2503.15268*, 2025.

[441] M. Turpin, J. Michael, E. Perez, and S. R. Bowman, "Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting," in *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[442] N. Miao, Y. W. Teh, and T. Rainforth, "Selfcheck: Using LLMs to zero-shot check their own step-by-step reasoning," in *The Twelfth International Conference on Learning Representations*, 2024.

[443] W. Jiang et al., "Forward-backward reasoning in large language models for mathematical verification," in *Findings of the Association for Computational Linguistics: ACL 2024*, L.-W. Ku, A. Martins, and V. Srikumar, Eds., Bangkok, Thailand: Association for Computational Linguistics, Aug. 2024, pp. 6647–6661.

[444] F. J. Binder et al., "Looking inward: Language models can learn about themselves by introspection," in *The Thirteenth International Conference on Learning Representations*, 2025.

[445] J.-J. Li et al., "Safetyanalyst: Interpretable, transparent, and steerable llm safety moderation," *arXiv preprint arXiv:2410.16665*, 2024.

[446] J. Betley, X. Bao, M. Soto, A. Sztyber-Betley, J. Chua, and O. Evans, "Tell me about yourself: Llms are aware of their learned behaviors," *arXiv preprint arXiv:2501.11120*, 2025.

[447] S. A. Akbar et al., "HalluMeasure: Fine-grained hallucination measurement using chain-of-thought reasoning," in *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, Y. Al-Onaizan, M. Bansal, and Y.-N.

Chen, Eds., Miami, Florida, USA: Association for Computational Linguistics, Nov. 2024, pp. 15 020–15 037.

[448] S. Lee, H. Lee, S. Heo, and W. Choi, "Hudex: Integrating hallucination detection and explainability for enhancing the reliability of llm responses," *arXiv preprint arXiv:2502.08109*, 2025.

[449] Z. Liu, R. A. Amjad, R. Adkathimar, T. Wei, and H. Tong, "Selfelicit: Your language model secretly knows where is the relevant evidence," *arXiv preprint arXiv:2502.08767*, 2025.

[450] S. Krishna, J. Ma, D. Slack, A. Ghandeharioun, S. Singh, and H. Lakkaraju, "Post hoc explanations of language models can improve language models," in *Proceedings of the 37th International Conference on Neural Information Processing Systems*, ser. NIPS '23, New Orleans, LA, USA: Curran Associates Inc., 2023.

[451] S. Soo, W. Teng, C. Balaganesh, T. Guoxian, and M. YAN, "Interpretable steering of large language models with feature guided activation additions," in *ICLR 2025 Workshop on Building Trust in Language Models and Applications*, 2025.

[452] R. Härle, F. Friedrich, M. Brack, B. Deiseroth, P. Schramowski, and K. Kersting, "Scar: Sparse conditioned autoencoders for concept detection and steering in llms," *arXiv preprint arXiv:2411.07122*, 2024.

[453] R. Zhao, X. Li, S. Joty, C. Qin, and L. Bing, "Verify-and-edit: A knowledge-enhanced chain-of-thought framework," in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, A. Rogers, J. Boyd-Graber, and N. Okazaki, Eds., Toronto, Canada: Association for Computational Linguistics, Jul. 2023, pp. 5823–5840.

[454] H. Kaur, H. Nori, S. Jenkins, R. Caruana, H. Wallach, and J. Wortman Vaughan, "Interpreting interpretability: Understanding data scientists' use of interpretability tools for machine learning," in *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, ser. CHI '20, Honolulu, HI, USA: Association for Computing Machinery, 2020, pp. 1–14, ISBN: 9781450367080.

[455] H. Lakkaraju, D. Slack, Y. Chen, C. Tan, and S. Singh, *Rethinking explainability as a dialogue: A practitioner's perspective*, 2022. arXiv: 2202.01875 [cs.LG].

[456] L. Sharkey et al., *Open problems in mechanistic interpretability*, 2025. arXiv: 2501.16496 [cs.LG].

[457] N. Kokhlikyan et al., "Captum: A unified and generic model interpretability library for pytorch," *arXiv preprint arXiv:2009.07896*, 2020.

[458] S. Hao et al., "LLM reasoners: New evaluation, library, and analysis of step-by-step reasoning with large language models," in *First Conference on Language Modeling*, 2024.

[459] F. Hohman, M. Kahng, R. Pienta, and D. H. Chau, "Visual analytics in deep learning: An interrogative survey for the next frontiers," *IEEE Transactions on Visualization and Computer Graphics*, vol. 25, no. 8, pp. 2674–2693, Aug. 2019.

[460] E. Beauxis-Aussalet et al., "The role of interactive visualization in fostering trust in ai," *IEEE Computer Graphics and Applications*, vol. 41, no. 6, pp. 7–12, 2021.

[461] B. La Rosa et al., "State of the art of visual analytics for explainable deep learning," in *Computer Graphics Forum*, Wiley Online Library, vol. 42, 2023, pp. 319–355.

[462] Q. V. Liao and J. Wortman Vaughan, "AI Transparency in the Age of LLMs: A Human-Centered Research Roadmap," *Harvard Data Science Review*, no. Special Issue 5, May 2024, https://hdsr.mitpress.mit.edu/pub/aelql9qy.

[463] Y. Wang, G. Sun, Z. Fu, and R. Liang, *Human-computer interaction and visualization in natural language generation models: Applications, challenges, and opportunities*, 2025. arXiv: 2410.08723 [cs.HC].

[464] C. Park et al., "Sanvis: Visual analytics for understanding self-attention networks," in *2019 IEEE Visualization Conference (VIS)*, 2019, pp. 146–150.

[465] Z. J. Wang, R. Turko, and D. H. Chau, "Dodrio: Exploring transformer models with interactive visualization," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, H. Ji, J. C. Park, and R. Xia, Eds., Online: Association for Computational Linguistics, Aug. 2021, pp. 132–141.

[466] A. Coscia, L. Holmes, W. Morris, J. S. Choi, S. Crossley, and A. Endert, "Iscore: Visual analytics for interpreting how language models automatically score summaries," in *Proceedings of the 29th International Conference on Intelligent User Interfaces*, ser. IUI '24, Greenville, SC, USA: Association for Computing Machinery, 2024, pp. 787–802, ISBN: 9798400705083.

[467] C. Yeh, Y. Chen, A. Wu, C. Chen, F. Viégas, and M. Wattenberg, "Attentionviz: A global view of transformer attention," *IEEE Transactions on Visualization and Computer Graphics*, vol. 30, no. 1, pp. 262–272, 2024.

[468] H. Strobelt, S. Gehrmann, M. Behrisch, A. Perer, H. Pfister, and A. M. Rush, "Seq2seq-vis: A visual debugging tool for sequence-to-sequence models," *IEEE Transactions on Visualization and Computer Graphics*, vol. 25, no. 1, pp. 353–363, 2019.

[469] S. Katz and Y. Belinkov, "VISIT: Visualizing and interpreting the semantic information flow of transformers," in *The 2023 Conference on Empirical Methods in Natural Language Processing*, 2023.

[470] K. Pal, J. Sun, A. Yuan, B. Wallace, and D. Bau, "Future lens: Anticipating subsequent tokens from a single hidden state," in *Proceedings of the 27th Conference on Computational Natural Language Learning (CoNLL)*, J. Jiang, D. Reitter, and S. Deng, Eds., Singapore: Association for Computational Linguistics, Dec. 2023, pp. 548–560.

[471] N. Nanda, *Neuroscope: A website for mechanistic interpretability of language models*, https://neuroscope.io, 2022.

[472] A. Garde, E. Kran, and F. Barez, *Deepdecipher: Accessing and investigating neuron activation in large language models*, 2023. arXiv: 2310.01870 [cs.LG].

[473] S. Bills et al., *Language models can explain neurons in language models*, https://openaipublic.blob.core.windows.net/neuron-explainer/paper/index.html, OpenAI, May 2023.

[474] J. Lin, *Neuronpedia: Interactive reference and tooling for analyzing neural networks*, Software available from neuronpedia.org, 2023.

[475] S. Chalnev, M. Siu, and A. Conmy, *Improving steering vectors by targeting sparse autoencoder features*, 2024. arXiv: 2411.02193 [cs.LG].

[476] Y. Li, Z. Zhang, K. Wang, L. Shi, and H. Wang, *Model-editing-based jailbreak against safety-aligned large language models*, 2024. arXiv: 2412.08201 [cs.CR].

[477] J. Su, "Enhancing adversarial attacks through chain of thought," *arXiv preprint arXiv:2410.21791*, 2024.

[478] T. Winninger, B. Addad, and K. Kapusta, *Using mechanistic interpretability to craft adversarial attacks against large language models*, 2025. arXiv: 2503.06269 [cs.LG].

[479] C. Shi et al., "Hypothesis testing the circuit hypothesis in llms," *Advances in Neural Information Processing Systems*, vol. 37, pp. 94 539–94 567, 2024.

[480] N. Alangari, M. El Bachir Menai, H. Mathkour, and I. Almosallam, "Exploring evaluation methods for interpretable machine learning: A survey," *Information*, vol. 14, no. 8, p. 469, 2023.

[481] S. Kong, Y. Shen, and L. Huang, "Resolving training biases via influence-based data relabeling," in *International Conference on Learning Representations*, 2022.

[482] M. Mozes, T. Bolukbasi, A. Yuan, F. Liu, N. Thain, and L. Dixon, *Gradient-based automated iterative recovery for parameter-efficient tuning*, 2023. arXiv: 2302.06598 [cs.CL].

[483] D. Slack, S. Krishna, H. Lakkaraju, and S. Singh, "Explaining machine learning models with interactive natural language conversations using talktomodel," *Nature Machine Intelligence*, vol. 5, no. 8, pp. 873–883, 2023.

[484] Q. Wang, T. Anikina, N. Feldhus, J. Genabith, L. Hennig, and S. Möller, "LLM-Checkup: Conversational examination of large language models via interpretability tools and self-explanations," in *Proceedings of the Third Workshop on Bridging Human–Computer Interaction and Natural Language Processing*, S. L. Blodgett et al., Eds., Mexico City, Mexico: Association for Computational Linguistics, Jun. 2024, pp. 89–104.

[485] C. Siska and A. Sankaran, "Attentiondefense: Leveraging system prompt attention for explainable defense against novel jailbreaks," *arXiv preprint arXiv:2504.12321*, 2025.

[486] Z. Yang, Z. Sun, T. Z. Yue, P. Devanbu, and D. Lo, "Robustness, security, privacy, explainability, efficiency, and usability of large language models for code," *arXiv preprint arXiv:2403.07506*, 2024.

[487] C. Xiong, X. Qi, P.-Y. Chen, and T.-Y. Ho, "Defensive prompt patch: A robust and interpretable defense of llms against jailbreak attacks," *arXiv preprint arXiv:2405.20099*, 2024.

[488] I. H. Sarker, "Llm potentiality and awareness: A position paper from the perspective of trustworthy and responsible ai modeling," *Discover Artificial Intelligence*, vol. 4, no. 1, p. 40, 2024.

[489] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, et al., "Llama 2: Open foundation and fine-tuned chat models," *arXiv:2307.09288*, 2023.

[490] Y. Zhang et al., "Siren's song in the ai ocean: A survey on hallucination in large language models," *arXiv preprint arXiv:2309.01219*, 2023.

[491] Y. Pan, L. Pan, W. Chen, P. Nakov, M.-Y. Kan, and W. Y. Wang, "On the risk of misinformation pollution with large language models," *arXiv preprint arXiv:2305.13661*, 2023.

[492] J. Zhou, Y. Zhang, Q. Luo, A. G. Parker, and M. De Choudhury, "Synthetic lies: Understanding ai-generated misinformation and evaluating algorithmic and human solutions," in *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, 2023, pp. 1–20.

[493] H. Kotek, R. Dockum, and D. Sun, "Gender bias and stereotypes in large language models," in *Proceedings of The ACM Collective Intelligence Conference*, 2023, pp. 12–24.

[494] R. Strom, "Fake chatgpt cases cost lawyers $5,000 plus embarrassment," *Bloomberg Law*, Jun. 22, 2023.

[495] G. Zuccon, B. Koopman, and R. Shaik, "Chatgpt hallucinates when attributing answers," in *Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region*, 2023, pp. 46–51.

[496] M. Kahng, I. Tenney, M. Pushkarna, M. X. Liu, J. Wexler, et al., "Llm comparator: Visual analytics for side-by-side evaluation of large language models," *arXiv:2402.10524*, 2024.

[497] K. Sparck Jones, "A statistical interpretation of term specificity and its application in retrieval," *Journal of documentation*, vol. 28, no. 1, pp. 11–21, 1972.

[498] S. Lee et al., "Diffusion explainer: Visual explanation for text-to-image stable diffusion," in *2024 IEEE Visualization and Visual Analytics (VIS)*, IEEE, 2024, pp. 96–100.

[499] 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.

[500] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution Image Synthesis with Latent Diffusion Models," in *CVPR*, 2022, pp. 10 684–10 695.

[501] Stability AI, *Stable Diffusion Public Release*, https://stability.ai/blog/stable-diffusion-public-release, Accessed on: 2022-08-22, 2022.

[502] OpenAI, *DALL-E 2*, https://openai.com/product/dall-e-2, Accessed on: 2022-09-28, 2022.

[503] M. Sung, "Lensa, the AI portrait app, has soared in popularity. But many artists question the ethics of AI art.," *NBC News*, 2022, Accessed on: 2023-04-30.

[504] J. Brusseau, "Acceleration AI Ethics, the Debate between Innovation and Safety, and Stability AI's Diffusion versus OpenAI's Dall-E," *arXiv preprint arXiv:2212.01834*, 2022.

[505] L. Choudhary, *Stable Diffusion is Now Accused of 'Stealing' Artwork*, https://analyticsindiamag.com/stable-diffusion-is-now-accused-of-stealing-artwork/, Accessed on: 2023-04-30, 2022.

[506] P. Dixit, *Meet The Three Artists Behind A Landmark Lawsuit Against AI Art Generators*, https://www.buzzfeednews.com/article/pranavdixit/ai-art-generators-lawsuit-stable-diffusion-midjourney, Accessed on: 2023-04-30, 2023.

[507] A. G. Eshoo, *Eshoo Urges NSA & OSTP to Address Unsafe AI Practices*, https://eshoo.house.gov/media/press-releases/eshoo-urges-nsa-ostp-address-unsafe-ai-practices, Accessed on: 2023-04-30, 2022.

[508] A. Engler, "Early thoughts on regulating generative AI like ChatGPT," *Brookings Institution*, Mar. 27, 2023, Accessed on: 2023-04-30.

[509] T. Ryan-Mosley, "An early guide to policymaking on generative AI," *MIT Technology Review*, Mar. 27, 2023, Accessed on: 2023-04-30.

[510] *Copyright Office Launches New Artificial Intelligence Initiative*, https://www.copyright.gov/newsnet/2023/1004.html, Accessed on: 2023-04-30, Mar. 16, 2023.

[511] J. Hendrix, "Generative AI, Section 230 and Liability: Assessing the Questions," *Tech Policy Press*, Mar. 23, 2023, Accessed on: 2023-04-30.

[512] A. Radford et al., "Learning Transferable Visual Models from Natural Language Supervision," in *International conference on machine learning*, PMLR, 2021, pp. 8748–8763.

[513] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional Networks for Biomedical Image Segmentation," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, Springer, 2015, pp. 234–241.

[514] D. P. Kingma and M. Welling, "Auto-encoding Variational Bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[515] P. von Platen, *Testing Stable Diffusion is hard*, https://github.com/huggingface/diffusers/issues/937, Accessed on: 2023-04-30, 2022.

[516] J. Alammar, *The illustrated Stable Diffusion*, https://jalammar.github.io/illustrated-stable-diffusion/, Accessed on: 2023-04-30, 2022.

[517] J. Howard, *From Deep Learning Foundations to Stable Diffusion*, https://www.fast.ai/posts/part2-2023.html, Accessed on: 2023-04-30, 2023.

[518] J. Alammar, *AI Art Explained: How AI Generates Images (Stable Diffusion, Midjourney, and DALLE)*, https://youtu.be/MXmacOUJUaw, Accessed on: 2023-04-30, 2023.

[519] M. Kahng, P. Y. Andrews, A. Kalro, and D. H. Chau, "Activis: Visual exploration of industry-scale deep neural network models," *IEEE TVCG*, vol. 24, no. 1, pp. 88–97, 2017.

[520] V. Liu and L. B. Chilton, "Design Guidelines for Prompt Engineering Text-to-image Generative Models," in *CHI*, 2022, pp. 1–23.

[521] J. Oppenlaender, "A Taxonomy of Prompt Modifiers for Text-to-Image Generation," *arXiv preprint arXiv:2204.13988*, 2022.

[522] Y. Feng et al., "Promptmagician: Interactive prompt engineering for text-to-image creation," *IEEE Transactions on Visualization and Computer Graphics*, 2023.

[523] J. B. Kai Ninomiya Brandon Jones, *Webgpu*, https://www.w3.org/TR/webgpu/, 2024.

[524] Machine Learning Compilation, *Web Stable Diffusion*, https://mlc.ai/web-stable-diffusion, Accessed on: 2024-04-28, 2024.

[525] E. Smith, *A Traveler's Guide to the Latent Space*, Accessed on: 2023-04-29, 2022.

[526] Andrew, *Stable Diffusion prompt: a definitive guide*, https://stable-diffusion-art.com/prompt-guide/, Accessed on: 2023-04-29, 2023.

[527] N. Pavlichenko and D. Ustalov, "Best Prompts for Text-to-Image Models and How to Find Them," *arXiv preprint arXiv:2209.11711*, 2022.

[528] M. Bostock, V. Ogievetsky, and J. Heer, "D^3 data-driven documents," *IEEE transactions on visualization and computer graphics*, vol. 17, no. 12, pp. 2301–2309, 2011.

[529] K. Turner, *Decoding latents to RGB without upscaling*, https://discuss.huggingface.co/t/decoding-latents-to-rgb-without-upscaling/23204/2, Accessed on: 2023-04-30, 2022.

[530] L. McInnes, J. Healy, and J. Melville, "Umap: Uniform Manifold Approximation and Projection for Dimension Reduction," *arXiv preprint arXiv:1802.03426*, 2018.

[531] H. J. Seltman, *Experimental design and analysis*, 2012.

[532] D. N. Parmar and B. B. Mehta, "Face recognition methods & applications," *CoRR*, vol. abs/1403.0485, 2014. arXiv: 1403.0485.

[533] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "Deepface: Closing the gap to human-level performance in face verification," in *2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014*, IEEE Computer Society, 2014, pp. 1701–1708.

[534] M. Wang and W. Deng, "Deep face recognition: A survey," *Neurocomputing*, vol. 429, pp. 215–244, 2021.

[535] G. Prabhakar, B. Kailath, S. Natarajan, and R. Kumar, "Obstacle detection and classification using deep learning for tracking in high-speed autonomous driving," in *2017 IEEE Region 10 Symposium (TENSYMP)*, 2017, pp. 1–6.

[536] H. Fujiyoshi, T. Hirakawa, and T. Yamashita, "Deep learning-based image recognition for autonomous driving," *IATSS Research*, vol. 43, no. 4, pp. 244–252, 2019.

[537] E. J. Topol, "High-performance medicine: The convergence of human and artificial intelligence," *Nature Medicine*, vol. 25, pp. 44–56, 2019.

[538] P. Shah et al., "Artificial intelligence and machine learning in clinical development: A translational perspective," *npj Digital Medicine*, vol. 2, no. 69, 2019.

[539] Y. Zoabi, S. Deri-Rozov, and N. Shomron, "Machine learning-based prediction of covid-19 diagnosis based on symptoms," *npj Digital Medicine*, vol. 4, no. 3, 2021.

[540] A. Buetti-Dinh et al., "Deep neural networks outperform human expert's capacity in characterizing bioleaching bacterial biofilm composition," *Biotechnology Reports*, vol. 22, e00321, 2019.

[541] K. S. Krishnapriya, V. Albiero, K. Vangara, M. C. King, and K. W. Bowyer, "Issues related to face recognition accuracy varying based on race and skin tone," *IEEE Transactions on Technology and Society*, vol. 1, no. 1, pp. 8–20, 2020.

[542] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan, "A survey on bias and fairness in machine learning," *ACM Comput. Surv.*, vol. 54, no. 6, 115:1–115:35, 2021.

[543] C. Garvie, A. Bedoya, and J. Frankle, *The perceptual line-up: Unregulated police face recognition in america*, 2016.

[544] O. A. Osoba and W. Welser, "An intelligence in our image: The risks of bias and errors in artificial intelligence," RAND Corporation, Santa Monica, Calif., 2017.

[545] K. A. Zweig, G. Wenzelburger, and T. D. Krafft, "On chances and risks of security related algorithmic decision making systems," *European Journal for Security Research*, vol. 3, pp. 181–203, 2018.

[546] A. L. Washington, "How to argue with an algorithm: Lessons from the compas propublica debate," *The Colorado Technology Law Journal*, vol. 17, 2019.

[547] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States*, 2012, pp. 1106–1114.

[548] C. Szegedy et al., "Going deeper with convolutions," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, IEEE Computer Society, 2015, pp. 1–9.

[549] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, IEEE Computer Society, 2016, pp. 770–778.

[550] N. Sharma, V. Jain, and A. Mishra, "An analysis of convolutional neural networks for image classification," *Procedia Computer Science*, vol. 132, pp. 377–384, 2018.

[551] F. Sultana, A. Sufian, and P. Dutta, "Advancements in image classification using convolutional neural network," *CoRR*, vol. abs/1905.03288, 2019. arXiv: 1905.03288.

[552] P. Drozdowski, C. Rathgeb, A. Dantcheva, N. Damer, and C. Busch, "Demographic bias in biometrics: A survey on an emerging challenge," *IEEE Transactions on Technology and Society*, 2020.

[553] T. Wang, J. Zhao, M. Yatskar, K. Chang, and V. Ordonez, "Balanced datasets are not enough: Estimating and mitigating gender bias in deep image representations," in *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, IEEE, 2019, pp. 5309–5318.

[554] M. Wang and W. Deng, "Mitigating bias in face recognition using skewness-aware reinforcement learning," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, Computer Vision Foundation / IEEE, 2020, pp. 9319–9328.

[555] P. Dhar, J. Gleason, H. Souri, C. D. Castillo, and R. Chellappa, "Towards gender-neutral face descriptors for mitigating bias in face recognition," *arXiv preprint arXiv:2006.07845*, 2020.

[556] S. Gong, X. Liu, and A. K. Jain, "Mitigating face recognition bias via group adaptive classifier," in *IEEE Conference on Computer Vision and Pattern Recognition,*

CVPR 2021, virtual, June 19-25, 2021, Computer Vision Foundation / IEEE, 2021, pp. 3414–3424.

[557] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, IEEE Computer Society, 2015, pp. 3730–3738.

[558] M. Bostock, V. Ogievetsky, and J. Heer, "D³ data-driven documents," *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 12, pp. 2301–2309, 2011.

[559] A. Paszke et al., "Pytorch: An imperative style, high-performance deep learning library," *Advances in neural information processing systems*, vol. 32, 2019.

[560] M. Jaderberg, A. Vedaldi, and A. Zisserman, "Speeding up convolutional neural networks with low rank expansions," in *British Machine Vision Conference, BMVC 2014, Nottingham, UK, September 1-5, 2014*, BMVA Press, 2014.

[561] W. Wen, C. Wu, Y. Wang, Y. Chen, and H. Li, "Learning structured sparsity in deep neural networks," *Advances in neural information processing systems*, vol. 29, 2016.

[562] Y. He, X. Zhang, and J. Sun, "Channel pruning for accelerating very deep neural networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 1389–1397.

[563] R. Duggal, C. Xiao, R. Vuduc, D. H. Chau, and J. Sun, "Cup: Cluster pruning for compressing deep neural networks," in *2021 IEEE International Conference on Big Data (Big Data)*, IEEE, 2021, pp. 5102–5106.

[564] S. Yeom, I. Giacomelli, M. Fredrikson, and S. Jha, "Privacy risk in machine learning: Analyzing the connection to overfitting," in *2018 IEEE 31st computer security foundations symposium (CSF)*, IEEE, 2018, pp. 268–282.

[565] Z. Li, K. Kamnitsas, and B. Glocker, "Overfitting of neural nets under class imbalance: Analysis and improvements for segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, 2019, pp. 402–410.

[566] T. Farrand, F. Mireshghallah, S. Singh, and A. Trask, "Neither private nor fair: Impact of data imbalance on utility and fairness in differential privacy," in *Proceedings of the 2020 Workshop on Privacy-Preserving Machine Learning in Practice*, 2020, pp. 15–19.

[567] W. Zheng and M. Jin, "The effects of class imbalance and training data size on classifier learning: An empirical study," *SN Computer Science*, vol. 1, no. 2, pp. 1–13, 2020.

[568] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1026–1034.

[569] 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.

[570] 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.

[571] L. Oakden-Rayner, J. Dunnmon, G. Carneiro, and C. Ré, "Hidden stratification causes clinically meaningful failures in machine learning for medical imaging," in *Proceedings of the ACM conference on health, inference, and learning*, 2020, pp. 151–159.

[572] K. T. Rodolfa, H. Lamba, and R. Ghani, "Empirical observation of negligible fairness–accuracy trade-offs in machine learning for public policy," *Nature Machine Intelligence*, vol. 3, no. 10, pp. 896–904, 2021.

[573] Z. Wang et al., "Towards fairness in visual recognition: Effective strategies for bias mitigation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 8919–8928.

[574] R. Srinivasan and A. Chander, "Biases in ai systems," *Communications of the ACM*, vol. 64, no. 8, pp. 44–49, 2021.

[575] S. Seo, J.-Y. Lee, and B. Han, "Information-theoretic bias reduction via causal view of spurious correlation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, 2022, pp. 2180–2188.

[576] Z. Wang et al., "Fairness-aware adversarial perturbation towards bias mitigation for deployed deep models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 10 379–10 388.

[577] I. Sarridis, C. Koutlis, S. Papadopoulos, and C. Diou, "Flac: Fairness-aware representation learning by suppressing attribute-class associations," *arXiv preprint arXiv:2304.14252*, 2023.

[578] H. Wang et al., "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.

[579] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[580] F. Hohman, H. Park, C. Robinson, and D. H. Chau, "Summit: Scaling deep learning interpretability by visualizing activation and attribution summarizations," *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 2020.

[581] 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)*, 2021.

[582] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*, Ieee, 2009, pp. 248–255.

[583] 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.

[584] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[585] 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.

[586] P. Welinder and P. Perona, "Online crowdsourcing: Rating annotators and obtaining cost-effective labels," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Workshops*, IEEE, 2010, pp. 25–32.

[587] C. Rashtchian, P. Young, M. Hodosh, and J. Hockenmaier, "Collecting image annotations using amazon's mechanical turk," in *Proceedings of the NAACL HLT 2010 workshop on creating speech and language data with Amazon's Mechanical Turk*, 2010, pp. 139–147.

[588] S.-i. Amari, "Backpropagation and stochastic gradient descent method," *Neurocomputing*, vol. 5, no. 4-5, pp. 185–196, 1993.

[589] Y. Hong and E. Yang, "Unbiased classification through bias-contrastive and bias-balanced learning," *Advances in Neural Information Processing Systems*, vol. 34, pp. 26 449–26 461, 2021.

[590] B. Simpson, F. Dutil, Y. Bengio, and J. P. Cohen, "Gradmask: Reduce overfitting by regularizing saliency," *arXiv preprint arXiv:1904.07478*, 2019.

[591] J. D. Viviano, B. Simpson, F. Dutil, Y. Bengio, and J. P. Cohen, "Saliency is a possible red herring when diagnosing poor generalization," *arXiv preprint arXiv:1910.00199*, 2019.

[592] 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)*, Dec. 2015.

[593] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A convnet for the 2020s," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 11 976–11 986.

[594] S. Wu et al., "Bloomberggpt: A large language model for finance," *arXiv preprint arXiv:2303.17564*, 2023.

[595] A. J. Thirunavukarasu, D. S. J. Ting, K. Elangovan, L. Gutierrez, T. F. Tan, and D. S. W. Ting, "Large language models in medicine," *Nature medicine*, vol. 29, no. 8, pp. 1930–1940, 2023.

[596] M. Bohannon, "Lawyer Used ChatGPT In Court—And Cited Fake Cases. A Judge Is Considering Sanctions," *Forbes*, 2023.

[597] Z. Ji et al., "Survey of hallucination in natural language generation," *ACM Computing Surveys*, vol. 55, no. 12, pp. 1–38, 2023.

[598] Z. Xu, S. Jain, and M. Kankanhalli, "Hallucination is inevitable: An innate limitation of large language models," *arXiv preprint arXiv:2401.11817*, 2024.

[599] J. Chen and J. Mueller, "Quantifying uncertainty in answers from any language model and enhancing their trustworthiness," in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2024, pp. 5186–5200.

[600] J. Kossen, J. Han, M. Razzak, L. Schut, S. Malik, and Y. Gal, "Semantic Entropy Probes: Robust and Cheap Hallucination Detection in LLMs," *arXiv preprint arXiv:2406.15927*, 2024.

[601] L. Huang et al., "A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions," *arXiv preprint arXiv:2311.05232*, 2023.

[602] J. Ferrando, O. Obeso, S. Rajamanoharan, and N. Nanda, "Do i know this entity? knowledge awareness and hallucinations in language models," *arXiv preprint arXiv:2411.14257*, 2024.

[603] A. Martino, M. Iannelli, and C. Truong, "Knowledge injection to counter large language model (llm) hallucination," in *European Semantic Web Conference*, Springer, 2023, pp. 182–185.

[604] K. Sun, Y. E. Xu, H. Zha, Y. Liu, and X. L. Dong, "Head-to-tail: How knowledgeable are large language models (llm)? aka will llms replace knowledge graphs?" *arXiv preprint arXiv:2308.10168*, 2023.

[605] B. Huang, C. Chen, X. Xu, A. Payani, and K. Shu, "Can knowledge editing really correct hallucinations?" *arXiv preprint arXiv:2410.16251*, 2024.

[606] G. Liu, X. Wang, L. Yuan, Y. Chen, and H. Peng, *Examining llms' uncertainty expression towards questions outside parametric knowledge*, 2024. arXiv: 2311.09731 [cs.CL].

[607] M. Honnibal and I. Montani, "spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing," 2017.

[608] A. Mallen, A. Asai, V. Zhong, R. Das, D. Khashabi, and H. Hajishirzi, "When not to trust language models: Investigating effectiveness of parametric and non-parametric memories," in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, A. Rogers, J. Boyd-Graber, and N. Okazaki, Eds., Toronto, Canada: Association for Computational Linguistics, Jul. 2023, pp. 9802–9822.

[609] N. Kandpal, H. Deng, A. Roberts, E. Wallace, and C. Raffel, "Large language models struggle to learn long-tail knowledge," in *International Conference on Machine Learning*, PMLR, 2023, pp. 15 696–15 707.

[610] Z. Ji, T. Yu, Y. Xu, N. Lee, E. Ishii, and P. Fung, "Towards mitigating hallucination in large language models via self-reflection," *arXiv preprint arXiv:2310.06271*, 2023.

[611] V. Hartmann, A. Suri, V. Bindschaedler, D. Evans, S. Tople, and R. West, "Sok: Memorization in general-purpose large language models," *arXiv preprint arXiv:2310.18362*, 2023.

[612] F. J. Massey Jr, "The Kolmogorov-Smirnov test for goodness of fit," *Journal of the American statistical Association*, vol. 46, no. 253, pp. 68–78, 1951.

[613] A. Dubey et al., "The llama 3 herd of models," *arXiv preprint arXiv:2407.21783*, 2024.

[614] A. Q. Jiang et al., "Mistral 7b," *arXiv preprint arXiv:2310.06825*, 2023.

[615] F. Tonolini, N. Aletras, J. Massiah, and G. Kazai, "Bayesian prompt ensembles: Model uncertainty estimation for black-box large language models," in *Findings of the Association for Computational Linguistics ACL 2024*, 2024, pp. 12 229–12 272.

[616] Q. Team, *Qwen2.5: A party of foundation models*, Sep. 2024.

[617] M. Joshi, E. Choi, D. S. Weld, and L. Zettlemoyer, "Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension," *arXiv preprint arXiv:1705.03551*, 2017.