

Interpreting and Aligning Large Language Models

Karel D'Oosterlinck

Doctoral dissertation submitted to obtain the academic degree of
Doctor of Computer Science Engineering

Supervisors

Prof. Chris Develder, PhD - Prof. Thomas Demeester, PhD
Department of Information Technology
Faculty of Engineering and Architecture, Ghent University

February 2025



ISBN null

NUR 984, 965

Wettelijk depot: null

Members of the Examination Board

Chair

Prof. Filip De Turck, PhD, Ghent University

Other members entitled to vote

Prof. Tijn De Bie, PhD, Ghent University

Prof. Tom Dhaene, PhD, Ghent University

Prof. A. Seza Dođruöz, PhD, Ghent University

Prof. Douwe Kiela, PhD, Stanford University, USA

Prof. Christopher Potts, PhD, Stanford University, USA

Supervisors

Prof. Chris Develder, PhD, Ghent University

Prof. Thomas Demeester, PhD, Ghent University

Acknowledgments

Embarking on this PhD journey has been an extraordinary experience, one that surpassed all my expectations. I would like to express my heartfelt gratitude to everyone who has supported and accompanied me on this journey. The interactions I've had and the relationships I've built have been deeply influential in ways I could not have imagined.

I would like to extend a special thanks to my supervisors at Ghent University, Professor Thomas Demeester and Professor Chris Develder. Throughout this journey, they have been reliable pillars, always ready to provide insightful feedback and guidance. From our very first meeting during my master's thesis, they have played an integral role in pushing the development of my ideas and guiding me to pursue my own path. I also want to thank them for cultivating a supportive atmosphere within the Text-to-Knowledge group, which made the PhD that much more enjoyable.

I would like to express my deepest gratitude to Professor Christopher Potts, one of my most invaluable collaborators throughout this entire PhD journey. Through his mentorship, I had the opportunity to conduct a research visit at Stanford University, an experience that proved to be a pivotal turning point in my academic development. This visit profoundly expanded my horizons, sparking numerous collaborations and lasting friendships that have shaped my personal and academic growth in ways I could not have anticipated. I am deeply grateful to Professor Potts for supporting me in this opportunity. Our collaboration has blossomed into a lasting partnership throughout this PhD that has not only enriched my research but also taught me many essential skills.

I would also like to express my sincere gratitude to Professor Douwe Kiela, whose support was instrumental in broadening my horizons beyond academia. By facilitating a productive research internship at Contextual AI and mentoring me throughout the process, he not only helped me gain clarity about my next steps after completing my PhD but also taught me a set of skills that will be incredibly valuable throughout the future of my career.

My gratitude extends to everyone I've had the privilege of collaborating with and to all those with whom I've engaged in extensive discussions on research topics. Your input and perspective have profoundly impacted my views on research and life in ways I cannot even begin to enumerate. While it would be impossible to list everyone here, I would like to acknowledge some key individuals. At Stanford University: Amir Zur, Aryaman Arora, Atticus Geiger, Brandon Papineau, Christopher Manning, Dilara Soylu, Elisa Kreiss, Jing Huang, Jon Saad-Falcon,

Julie Kallini, Omar Khattab, Omar Shaikh, Shikhar Murty, Tristan Thrush, and Zhengxuan Wu. At Ghent University: Amir Hadifar, Cédric Goemaere, François Remy, Félix Koulischer, Henri Arno, Jens-Joris Decorte, Johannes Deleu, Klim Zaporjets, Marija Pizurica, Paloma Rabaey, and Semere Kiros Bitew. At Contextual AI: Amanpreet Singh and Shikib Mehri. Thank you.

To my friends and family, your unwavering support and encouragement have meant the world to me. Thank you for believing in me and for providing the support that has continually kept me going. I would like to specifically thank my parents Marc and Lut, my brothers Willem and Hendrik, Deste, Aline, Gust, and Lara for their unconditional love and encouragement. I also want to thank some of my friends—Arent, Jens-Joris, Ines, Silas, Charles, Lars, and Andre—whose friendship has been an especially important source of joy and strength these past years.

I would like to sincerely thank the members of my doctoral committee, Professor Filip De Turck, Professor Tijn De Bie, Professor Tom Dhaene, Professor A. Seza Dođruöz, Professor Douwe Kiela, Professor Christopher Potts, Professor Thomas Demeester, and Professor Chris Develder, for their valuable feedback and insightful comments. Their expertise and thoughtful suggestions have greatly contributed to refining my work and enhancing the quality of my research. Additionally, I am grateful to the FWO for their financial support, which made parts of this research possible.

As I conclude this journey, I am reminded of the importance of paying it forward—of sharing knowledge, offering guidance, and helping others in the way so many have helped me. I hope to continue this cycle of support and collaboration as I move forward in my career and life.

Ghent, Fall 2024
Karel

Table of Contents

Acknowledgments	i
List of Figures	vii
List of Tables	xi
Samenvatting (Dutch Summary)	xv
Summary	xix
Publications and Contributions	xxiii
1 Introduction	1
1.1 Artificial Intelligence	1
1.1.1 Impact	2
1.1.2 Issues	2
1.1.3 Interpretability and Alignment	3
1.2 Deep Learning	4
1.2.1 Artificial Neural Networks	5
2 Interpretability	7
2.1 The Utility of Interpretability	8
2.1.1 Interpretability in Critical Scenarios	8
2.1.2 Counterarguments for Interpretability	8
2.1.3 Other Interpretability Benefits	9
2.2 What are Interpretations?	10
2.2.1 The Need for Abstractions	10
2.2.2 Conceptual Abstractions	10
2.3 Measuring Interpretations	11
2.3.1 Causal Estimates	11
2.3.2 Counterfactual data	11
2.4 Benchmarking Interpretability Methods	12
2.4.1 Formalizing Interpretability Methods	12
2.4.2 Standardizing Interpretability Methods	13
2.4.3 Faithfulness metrics	13
2.4.4 Counterfactual Dataset	13

2.4.5	Results	13
2.5	Learning to Interpret	14
2.5.1	Interpreting AI with AI	14
2.5.2	Training an AI Interpreter	14
2.5.3	Results	15
2.6	Conclusions	15
2.6.1	Future Work	15
2.7	The Paradox of Practical Interpretability?	16
3	Alignment	17
3.1	Impact of Alignment	17
3.2	Detecting Alignment Issues	18
3.3	Preventing Alignment Issues	19
3.3.1	Imitation Learning	19
3.3.2	Preference Pairs	19
3.3.3	Preference Learning	20
3.4	Precise Preferences	22
3.4.1	Minimally Contrastive Preferences	22
3.4.2	Revisions	23
3.5	Anchored Alignment	24
3.5.1	What You Know Impacts How You Learn	24
3.5.2	Anchored Preference Optimization	25
3.6	Experiments	26
3.7	Conclusions	27
3.7.1	Future Work	27
4	Conclusions	29
5	Technical: Measuring Interpretations	31
5.1	Abstract	32
5.2	Introduction	32
5.3	Previous Work	34
5.4	Estimating Concept Effects with CEBaB	35
5.5	Evaluated Explanation Methods	38
5.6	The CEBaB Dataset	40
5.7	Experiments and Results	42
5.8	Conclusion	44
5.9	Acknowledgement	45
5.10	Appendix	45
5.10.1	Causal Concept Effects and Metrics for Explanation Methods	45
5.10.1.1	Theoretical Quantities	46
5.10.1.2	Empirical Estimates	47
5.10.1.3	Explainer Errors	48
5.10.2	CEBaB	48
5.10.2.1	Restaurant-level metadata from OpenTable	49

5.10.2.2	Crowdworkers	49
5.10.2.3	Editing Phase	49
5.10.2.4	Validation Phase	49
5.10.2.5	Review-level Rating Phase	51
5.10.2.6	Randomly Selected Examples	51
5.10.2.7	Five-way Empirical ATE for CEBaB	51
5.10.2.8	Edit variability	51
5.10.3	CEBaB Modeling Experiments	54
5.10.3.1	Experiments Set-up	55
5.10.3.2	Models	55
5.10.3.3	Multi-class Sentiment Analysis Benchmark	56
5.10.3.4	Aspect-based Sentiment Analysis Benchmark	56
5.10.3.5	Author Identity Prediction	57
5.10.4	Additional Results	57
5.10.5	CausaLM	58
5.10.5.1	Our adaptation	58
5.10.5.2	Implementation details	60
5.10.6	INLP	61
5.10.7	Our adaptation	61
5.10.7.1	Implementation details	61
5.10.8	TCAV	62
5.10.8.1	Our adaptation	62
5.10.8.2	Implementation details	63
5.10.9	ConceptSHAP	63
5.10.9.1	Our adaptation	63
5.10.9.2	Completeness Scores of Treatment Concepts	64
5.10.9.3	Hyperparameters	64
6	Technical: Learning to Interpret	65
6.1	Abstract	66
6.2	Introduction	66
6.3	Related Work	67
6.4	Causal Proxy Model (CPM)	68
6.5	Experiment Setup	72
6.5.1	Causal Estimation-Based Benchmark (CEBaB)	72
6.5.2	Evaluation Metrics	72
6.5.3	Baseline Methods	73
6.5.4	Causal Proxy Models	74
6.6	Results	74
6.6.1	CEBaB Performance	75
6.6.2	Self-Explanation with CPM	76
6.6.3	Concept-Aware Feature Attribution with CPM _{HI}	76
6.7	Conclusion	78
6.8	Acknowledgement	78
6.9	Appendix	79

6.9.1	CEBaB Dataset Statistics	79
6.9.2	Types of Approximate Counterfactual Pairs	79
6.9.3	Training Regimes	80
6.9.4	Additional Baseline Results	81
6.9.5	Intervention Site Location and Size	83
6.9.6	Ablation Study of CPM _{HI}	83
6.9.7	GPT-3 Generation Process	85
6.9.8	Integrated Gradients	86
6.9.9	Model Debiasing	87
6.9.10	Learning Dynamics	88
7	Technical: Controlling Alignment	95
7.1	Abstract	96
7.2	Introduction	96
7.3	Underspecification in Alignment	98
7.4	Contrastive Learning from Revisions	99
7.5	Anchored Preference Optimization	101
7.6	Alignment Experiments	104
7.6.1	Evaluation Methodology	104
7.6.2	Training Specifications	105
7.6.3	Results	106
	7.6.3.1 Preference Data	106
	7.6.3.2 Alignment Objectives	107
7.6.4	Analysis	107
	7.6.4.1 Preference Data	108
	7.6.4.2 Alignment Objectives	108
7.7	Related Work	108
7.8	Future work	110
7.9	Conclusion	111
7.10	Acknowledgment	111
7.11	Appendix	111
7.11.1	Preference Dataset Creation	111
	7.11.1.1 Prompts	111
	7.11.1.2 Preference Pair Filtering	111
7.11.2	MixEval-Hard Performance Breakdown	113
7.11.3	Unpaired APO	114
7.11.4	How well does AlpacaEval control for lengths?	114
	References	117

List of Figures

5.1	A causal graph describing a data generating process with an exogenous variables U and V representing the state of the world, mediating concepts $C_1, C_2 \dots, C_k$, and data X that is featurized with ϕ . $\phi(X)$ is input to a classifier \mathcal{N} , which outputs a vector of scores over m output classes.	36
5.2	ICaCE-Error (Definition 3) for <code>bert-base-uncased</code> fine-tuned for five-way sentiment, averaged per aspect. We report values for cosine, L2, and normdiff. Lower is better . Stars mark the best result(s) per metric. Results averaged over 5 distinct seeds. †RandomExplainer takes the difference between two random probability vectors as the predicted effect.	43
5.3	Edit phase annotation interface where the task was to convey ‘Positive’ or ‘Negative’ for the target aspect.	50
5.4	Edit phase annotation interface where the task was to hide the sentiment of the target aspect.	50
5.5	Normalized edit distances between original texts and those created during the editing phase for CEBaB.	51
5.6	Validation phase annotation interface.	52
5.7	Review-level annotation interface.	52
5.8	Pairwise absolute difference in majority (a) and average (b) review score for all double edits. Figure (a) only considers the 132 pairs where both edits have an actual review majority. Figure (b) considers all 176 pairs. Averages of the distributions are shown with a dotted vertical line.	54
5.9	Pairwise review majority distribution for all double edits in 5-way (a), ternary (b), and binary (c) classification settings. Figures (a) and (b) consider only the 132 pairs where both edits have an actual review majority. Figure (c) considers the 76 pairs that have both a review majority and non-neutral labels.	55
5.10	ICaCE-Error for all experiments using the <i>cosine</i> distance metric. Lower is better . Results averaged over 5 distinct seeds. Error bars (in gray) display the standard deviation. Stars denote the best results for a given classification setting.	60

5.11	ICaCE-Error for all experiments using the <i>normdiff</i> distance metric. Lower is better. Results averaged over 5 distinct seeds. Error bars (in gray) display the standard deviation. Stars denote the best results for a given classification setting.	61
5.12	ICaCE-Error for all experiments using the <i>L2</i> distance metric. Lower is better. Results averaged over 5 distinct seeds. Error bars (in gray) display the standard deviation. Stars denote the best results for a given classification setting.	62
6.1	Causal Proxy Model (CPM) summary. Every CPM for model \mathcal{N} is trained to mimic the factual behavior of \mathcal{N} ($\mathcal{L}_{\text{Mimic}}$). For CPM_{IN} , the counterfactual objective is \mathcal{L}_{IN} . For CPM_{HI} , the counterfactual objective is \mathcal{L}_{HI}	69
6.2	CEBaB scores for different intervention site locations and sizes for CPM_{HI} . The scores are measured in three different metrics on the test set for four different model architectures as a five-class sentiment classification task. Results averaged over three distinct seeds. Task performance as Macro-F1 score is reported when applicable. Shaded areas outline $\pm \text{SD}$	80
6.3	Example GPT-3 prompt (gray) and GPT-3 completion (bold). Note that all original examples convey the same sentiment towards service ($c = \text{negative}$) and same overall sentiment ($y = \text{neutral}$), and that the counterfactual examples are all edited such that the sentiment towards service is the same ($c' = \text{positive}$).	90
6.4	Example GPT-3 prompt (gray) and GPT-3 completion (bold). Note that all original examples convey the same sentiment towards service ($c = \text{unknown}$) and same overall sentiment ($y = \text{negative}$), and that the counterfactual examples are all metadata-sampled such that the sentiment towards service is the same ($c' = \text{positive}$).	91
6.5	Debiasing visualizations for different concepts of a CPM_{HI} with BERT model architecture. Individual plots are correlation plots between concept labels of a concept and the overall sentence sentiment label.	92
6.6	CEBaB scores measured in three different metrics on the dev and the test sets for a CPM_{HI} with the BERT architectures for different training epochs. Task performance as Macro-F1 score is reported.	93
7.1	Alignment is underspecified with regard to preferences and training objective. A: Preference pairs can vary along irrelevant aspects, Contrastive Learning from AI Revisions (CLAIR) creates a targeted preference signal instead. B: The quality of the model can impact alignment training, Anchored Preference Optimization (APO) explicitly accounts for this.	97

-
- 7.2 An **answer** produced by Llama-3-8B-Instruct for a **prompt**, and corresponding GPT4-turbo **revision** of this answer. The differences between answer and revision are highlighted. The revision generally follows the same outline as the answer but improves it where possible. For example, the revision correctly alters the count of Parisian restaurants from 2 to 3 in the second line of the answer. . . . 99
- 7.3 Comparison of gradients between DPO (equation **A**), APO-zero (equation **B**), and APO-down (equation **C**). Each gradient term is decomposed in a **direction** and **magnitude** factor. **Direction:** Either APO variant specifies explicitly if winning and losing likelihoods should increase or decrease during training. DPO only increases the likelihood difference, causing ambiguity with regard to the actual movement of these likelihoods during training. This explicit specification of direction is core to APO variants, and allows for a tighter fit between model and data during alignment. **Magnitude:** Each term in APO is scaled with a delta function. Here, $\delta(x) = \sigma(x)(1 - \sigma(x))$ is a function with a global maximum at $x = 0$ that tends to 0 for $x \rightarrow \pm\infty$. This causes APO gradients to saturate whenever the quantities being optimized have changed a lot compared to the beginning of training. Ethayarajh et al. [2024] theorize that such scaling leads to more robust optimization. . . . 102
- 7.4 **Log-likelihood** and **reward** on held-out winning and losing outputs for Llama-3-8B-Instruct trained on **CLAIR**, **on-policy judge**, **off-policy judge**, and **Stronger Preferred** preference datasets, using APO-down, APO-zero, or DPO alignment objectives. . . . 107

List of Tables

5.1	Toy examples illustrating the structure of CEBaB (actual corpus examples are longer and more complex; a sample is given in Appendix 5.10.2). Beginning from an OpenTable review, we give crowdworkers an actual restaurant review and they generate counterfactual restaurant reviews that would have been written if some aspect of the dining experience were changed and all else were held constant. Five different crowdworkers labeled each of the actual and counterfactual texts according to their aspect-level sentiment and overall sentiment. Aspect level sentiment labels are three way: ‘+’ (positive sentiment), ‘-’ (negative), and ‘unk’ (the aspect’s value is not expressed in the text). Overall sentiment labels are 1 (worst) to 5 (best). Edited aspect labels are shown in blue.	33
5.2	The evaluated explanation methods and their attributes. Explainer Method denotes the complexity of the models used by each explanation method. Access to Explained Model denotes the degree of access an explainer method needs to the explained model. Concept Labels Needed indicates whether a method estimating the effect for an input $x_u^{C=c}$ needs the actual input label c and/or the intervened value c' at test time. Models with a Counterfactual Representation approximate $\phi(x_u^{C=c'})$ to estimate the effect. Finally, only CausaLM and S-Learner have Confounder Control to minimize the impact of confounding concepts. *We predict these labels with a classifier.	38
5.3	Dataset statistics of CEBaB combining train/dev/test splits.	41
5.4	$\widehat{\text{CaCE}}$ (Definition 5.4) for <code>bert-base-uncased</code> fine-tuned as a 5-way sentiment classifier. Rows are concepts, columns are real-world concept interventions, and each entry indicates the average change in classifier output when the concept is intervened on with the given direction. ¹ Results are averaged over 5 distinct seeds with standard deviations. The $\widehat{\text{CaCE}}$ value of changing concept C from c to c' is the negative $\widehat{\text{CaCE}}$ value of changing concept C from c' to c	43
5.5	CEBaB metadata from OpenTable, tabulated at the level of individual original reviews. A total of 1,084 restaurants are represented in the data.	49

5.6	Randomly sampled edit pairs from CEBaB.	53
5.7	Empirical \widehat{ATE} for the binary sentiment labels in CEBaB. Reversing concept order results in the negation of the value given.	54
5.8	Model performance results for sequence classification as well as aspect-based sentiment analysis (ABSA) under 3 training conditions. Mean Macro-F1 scores across 5 runs with distinct random seeds are reported.	56
5.9	Model performance on top-k author identity prediction with number of train and dev examples.	57
5.10	ICaCE scores on the test set for the binary, ternary and 5-way classification settings. Lower is better . Results averaged over 5 distinct seeds; standard deviations in parentheses.	59
6.1	CEBaB scores measured in three different metrics on the test set for four different model architectures as a five-class sentiment classification task. Lower is better . Results averaged over three distinct seeds, standard deviations in parentheses. The metrics are described in Section 6.5. Best averaged result is bolded (including ties) per approximate counterfactual creation strategy.	75
6.2	Task performance measured as Macro-F1 score on the test set. Results averaged over three distinct seeds; standard deviations in parentheses.	77
6.3	Self-explanation CEBaB scores measured in three different metrics on the test set for four different model architectures as a five-class sentiment classification task. Lower is better . Results averaged over three distinct seeds, standard deviations in parentheses. . . .	77
6.4	Visualizations of word importance scores using Integrated Gradient (IG) by restricting gradient flow through the corresponding intervention site of the targeted concept. Our target class pools <i>positive</i> and <i>very positive</i> . Individual word importance is the sum of neuron-level importance scores for each input, normalized to $[-1, +1]$. -1 means the word contributes the most negatively to predicting the target class (red); +1 means the word contributes the most positively (green).	78
6.5	Dataset statistics.	79
6.6	CEBaB scores measured in three different metrics on the test set for four different model architectures as a five-class sentiment classification task. Results are adapted from Abraham et al. [2022]. Lower is better ; standard deviations over 5 distinct seeds in parentheses. Results are aggregated over all aspects and all directional concept label changes. Details about these evaluation metrics can be found in Section 6.5. Results are based on [†] Abraham et al. [2022], [‡] Künzel et al. [2019], and [§] Ravfogel et al. [2020].	82

6.7	CEBaB scores for additional baselines we considered. CEBaB scores are measured in three different metrics on the test set for four different model architectures as a five-class sentiment classification task. Lower is better . Results averaged over three distinct seeds, standard deviations in parentheses. Details about these evaluation metrics can be found in Section 6.5.	82
6.8	Ablation study of our CPM _{HI} method trained with <i>human</i> approximate counterfactual strategy. CEBaB scores measured in three different metrics on the test set for four different model architectures as a five-class sentiment classification task. Lower is better . Results averaged over three distinct seeds, standard deviations in parentheses.	84
6.9	Ablation study of our CPM _{HI} method for different <i>source</i> input <i>s</i> sampling strategies at inference time. CEBaB scores measured in three different metrics on the test set for four different model architectures as a five-class sentiment classification task. Lower is better . Results averaged over three distinct seeds, standard deviations in parentheses.	84
6.10	Additional visualizations of word importance scores using Integrated Gradient (IG) by restricting gradients flow through corresponding intervention site of the targeted concept. This table extends Table 6.4 in the main text.	87
6.11	Visualizations of word importance scores using Integrated Gradient (IG), using the same methods as in Table 6.4 and Table 6.10. . . .	88
7.1	Average token-level Jaccard similarity (intersection over union) and average character-level Levenshtein edit-distance between winning y_w and losing y_l answers for four comparable preference datasets built on top of Llama-3-8B-Instruct. The CLAIR dataset produces the best contrasts on both metrics.	101
7.2	Max and mean MixEval-Hard improvements for the 2024-06-01 and 2024-08-11 splits, aggregated over 18 epochs of aligning Llama-3-8B-Instruct. Best overall performance bold , best performance per dataset <u>underlined</u> , standard deviation in parentheses. While MixEval-Hard functions as our primary evaluation tool, we also report the average LC-AlpacaEval2.0 score increase over the two best MixEval-Hard checkpoints, and average length increase (in characters) of the responses. CLAIR leads to the greatest overall performance improvement on MixEval-Hard. APO methods achieve the best performance across both Judged and CLAIR datasets.	105

- 7.3 Prompt templates used for creating preference triples (x, y_l, y_w) with the *Reviser* and *Judge* function of Equation 7.1 and 7.2. The variables in the prompt template are **bolded** and bracketed. Both prompts target clear, correct, and engaging outputs. The *Reviser* prompt instructs that a losing output y_l should be minimally improved to create the winning output y_w . Instead, the *Judge* prompt picks the winning/losing output out of two candidates y_1 & y_2 . Both prompts also instruct a model to produce a reasoning before revising or judging. 112
- 7.4 Breakdown of `MixEval-Hard` performance (version 2024-06-01) in function of which dataset the queries originate from. Analysis given for `Llama-3-8B-Instruct` and our best models on the `CLAIR`, `Judge (on-policy)`, `Judge (off-policy)`, and `Stronger Preferred` datasets. While individual splits may not always indicate the best model (particularly when the amount of queries is low), the overall score correlates highly with human judgments about model performance (Chatbot Arena Elo; Chiang et al. 2024). `MixEval-Hard` uses a `GPT3.5-turbo` model to rate if a response to a query agrees with a known gold-truth response. 113
- 7.5 Max and mean `MixEval-Hard` improvements for the 2024-06-01 and 2024-08-11 splits, aggregated over 18 epochs of aligning `Llama-3-8B-Instruct`. Best overall performance **bold**, best performance per dataset underlined, standard deviation in parentheses. KTO is the best unpaired loss given the off-policy `Judge` and `CLAIR` datasets, while APO performs better when given the on-policy `Judge` and `Stronger Preferred` datasets. KTO can take 60% longer to train for the same configuration. †In this table, we use the APO-zero-unpaired loss for all APO experiments. 114

Samenvatting (Dutch Summary)

Moderne AI-systemen brengen aanzienlijke uitdagingen met zich mee om hun interne werking te begrijpen en hun uitkomsten en gedrag nauwkeurig te controleren. Dit leidt tot zowel ethische als praktische moeilijkheden bij hun toepassing. We hebben bijvoorbeeld slechts beperkte kennis van hoe een systeem zich zal gedragen in een nieuwe, ongeziene omgeving, terwijl dit cruciaal is om veilig gebruik te garanderen. Evenzo is ons vermogen om gerichte en efficiënte veranderingen in het gedrag van het systeem door te voeren, zonder ongewenste bijwerkingen te veroorzaken, beperkt. Gezien de potentiële bijdragen die AI kan leveren aan de mensheid, behoort het aanpakken van deze kwesties tot de belangrijkste taken waarmee we worden geconfronteerd.

Op het eerste gezicht lijken deze uitdagingen op het gebied van interpretatie en controle paradoxaal. Hoe kan de interne werking van een AI-systeem een mysterie voor ons blijven, terwijl wij degenen zijn die deze systemen ontwerpen en op computers uitvoeren? Evenzo: hoe is het mogelijk om buitengewoon performante AI-systemen te bouwen, maar toch moeite te hebben om nauwkeurig te sturen wat ze leren en of hun gedrag overeenkomt met de bedoelingen van de ontwikkelaar?

In dit proefschrift behandelen we verschillende fundamentele vragen met betrekking tot AI-interpretatie en AI-alignment, waarbij we ons uitsluitend baseren op basisveronderstellingen over AI-systemen. Op basis van dit conceptuele fundament introduceren we onze bijdragen, die conceptueel eenvoudig zijn maar diepgaande technische implicaties hebben voor zowel interpretatie als alignment. Deze bijdragen werpen niet alleen licht op belangrijke uitdagingen, maar bieden ook nieuwe perspectieven en methoden om deze aan te pakken. Naarmate we vorderen, verbinden we deze kwesties met bredere gebieden binnen AI-onderzoek en reflecteren we op cruciale momenten in de recente evolutie van het veld.

In dit werk geven we een hoog-niveau uitleg van Deep Learning-systemen om een gemeenschappelijk begrip te creëren, terwijl we waar nodig gedetailleerde commentaren bieden. Doorheen de discussie benadrukken we belangrijke aspecten van deze systemen, zoals de structuur van neurale netwerken, de rol van trainingsdata en de optimalisatieprocessen die leren mogelijk maken. Door conceptuele overzichten in evenwicht te brengen met gerichte details waar relevant, streven we ernaar een solide basis te leggen voor onze verkenning van interpretatie en alignment.

Deep Learning-systemen zijn berucht moeilijk te interpreteren. We kunnen *elke* berekening binnen het systeem *observeren*, maar we worstelen om echt te *begrijpen* hoe deze berekeningen samenkomen tot betekenisvol gedrag. Deze systemen zijn in staat om indrukwekkende taken uit te voeren, zoals het besturen van een auto, het

schrijven van poëzie of het uiteenzetten van wiskundige redeneringen, maar we begrijpen niet volledig hoe deze vaardigheden conceptueel worden geïmplementeerd. Elke individuele berekening in een Deep Learning-systeem is een eenvoudige vorm van numerieke rekenkunde—maar waar, en hoe, combineren deze berekeningen zich om concepten als “*Ik schrijf nu een haiku*” of “*Rem voor deze voetganger*” te representeren?

Wat betekent het eigenlijk om een AI-systeem te begrijpen en te interpreteren? In ons werk nemen we een causale benadering van interpretatie: een interpretatie moet ons informatie geven over hoe een bepaalde verandering in een invoer het uitvoergedrag van het AI-systeem zal beïnvloeden.

Onze eerste technische bijdrage op het gebied van AI-interpretatie maakt gebruik van deze causale benadering om bestaande interpretatiemethoden op een gelijke basis te vergelijken en te benchmarken. Voor ons werk drukten bestaande methoden die AI-systemen wilden interpreteren deze interpretaties allemaal anders uit, waardoor het onmogelijk was hun effectiviteit te vergelijken.

In onze zoektocht naar benchmarks voor interpretatie hebben we nieuwe wiskundige formaliseringen en een raamwerk voor AI-interpretatie bijgedragen, evenals een nieuwe dataset waarmee de praktische prestaties van interpretatiemethoden kunnen worden gemeten. Onze resultaten gaven aan dat geen enkele bestaande interpretatiemethode bijzonder effectief was, wat het belang van onze bijdrage onderstreept: we moeten de effectiviteit van interpretaties rigoureus meten en benchmarken.

Gemotiveerd door dit resultaat hebben we geprobeerd de prestaties van interpretatiemethoden verder te verbeteren. Onze tweede bijdrage in dit veld interpreteert een AI-systeem met behulp van een secundair AI-systeem. Ons werk toont aan dat dit zeer effectief kan zijn, wat consistent leidt tot de beste interpretaties in onze experimenten.

Effectief AI-systemen interpreteren is slechts een gedeeltelijke vereiste voor effectieve en ethische AI. We moeten er ook voor zorgen dat elk systeem dat we bouwen in lijn is met de intenties van de ontwikkelaar. Onze bijdrage op het gebied van AI-alignment bestaat uit een kritische evaluatie van bestaande trainings- en alignmentmethoden voor AI. Tot onze verrassing ontdekten we dat state-of-the-art methoden op sommige belangrijke dimensies ernstig ondergespecificeerd waren. Experimenteel ontdekten we dat dit een aanzienlijke invloed kan hebben op hoe het systeem zich uiteindelijk gedraagt. We introduceren nieuwe methoden die hier rekening mee houden, zowel door betere trainingsdata te produceren als door preciezere trainingsalgoritmen te ontwikkelen, en we vinden dat dit het resultaat en de precisie van AI-training aanzienlijk kan verbeteren.

Zowel ons werk op het gebied van AI-interpretatie als AI-alignment is gedeeltelijk gebaseerd op het concept van *counterfactual data*. Counterfactual data vertegenwoordigt een “wat als”-scenario; het belichaamt *wat er zou zijn gebeurd als dit of dat anders was geweest* [Pearl, 2009b]. Maar er is een addertje onder het gras: counterfactual data bestaat niet echt. Die kans is verkeken—we kunnen niet terug in de tijd om dingen te veranderen. Toch echoën de “wat als”-scenario’s soms luider dan de schreeuw van “wat was”, en deze echo’s hebben ons werk op

AI-interpretatie en alignment diepgaand beïnvloed.

We beginnen in Hoofdstuk 1 met het introduceren van de fundamentele concepten die nodig zijn om een intuïtief begrip van de huidige generatie AI op te bouwen, met de nadruk op Large Language Models en hun transformerende impact. We houden de conceptuele details in deze secties tot een minimum, en reserveren diepere uitleg voor latere secties waar ze direct relevant zijn. Vervolgens onderzoekt Hoofdstuk 2 de cruciale vraag van AI-interpretatie, waarbij we verkennen hoe we complexe modellen transparanter en begrijpelijker kunnen maken, en welke voordelen interpreteerbare AI ons daadwerkelijk biedt. In Hoofdstuk 3 richten we ons op het even cruciale probleem van AI-alignment, waarbij we ingaan op hoe we kunnen garanderen dat AI-systemen handelen in overeenstemming met de bedoelingen van de ontwikkelaar. Op basis van deze discussies reflecteert Hoofdstuk 4 op de overkoepelende bijdragen van dit werk en verbindt de inzichten en vooruitgangen die in de voorgaande hoofdstukken zijn gepresenteerd.

Voor lezers die op zoek zijn naar diepere technische details, bieden Hoofdstukken 5 and 6 een gedetailleerd verslag van onze methodologische innovaties op het gebied van AI-interpretatie, terwijl Hoofdstuk 7 onze technische bijdragen aan het veld van AI-alignment presenteert.

Summary

Modern AI systems pose significant challenges in understanding their inner workings and precisely controlling their outcomes and behaviors, leading to substantial ethical and practical difficulties in their deployment. For instance, we only have limited understanding of how a system will behave in a new, unseen environment, yet this is crucial for ensuring safe use. Likewise, our ability to introduce targeted and efficient changes to the system's behavior, without causing undesirable side effects, is limited. Given the potential contributions AI stands to make to humanity, addressing these issues is among the most crucial tasks we face.

At first glance, these challenges of interpretability and control seem paradoxical. How can the inner workings of an AI system remain a mystery to us, when we are the ones designing and running these systems on computers? Similarly, how is it possible to build extraordinarily performant AI systems, yet struggle to control precisely what they learn and if their behavior is aligned with the developer's intentions?

In this dissertation, we will address several fundamental questions related to AI Interpretability and AI alignment, relying solely on basic assumptions about AI systems. Building on this conceptual foundation, we will introduce our contributions, which are conceptually simple yet carry profound technical implications for both interpretability and alignment. These contributions not only shed light on key challenges but also offer new perspectives and methods for addressing them. As we progress, we connect these issues to broader areas within AI research and reflect on pivotal moments in the field's recent evolution.

In this work, we provide a high-level explanation of Deep Learning systems to establish a common understanding, while offering more detailed commentary when relevant. Throughout the discussion, we highlight key aspects of these systems, such as the structure of neural networks, the role of training data, and the optimization processes that enable learning. By balancing conceptual overviews with focused details where necessary, we aim to build a solid foundation for our exploration of interpretability and alignment.

Deep Learning systems are notoriously difficult to interpret. We can *observe* every computation within the system, yet we struggle to truly *understand* how these computations come together to form meaningful behavior. These systems are capable of performing impressive tasks, such as steering a car, writing poetry, or outlining mathematical reasoning, yet we don't fully grasp how these abilities are conceptually implemented. Every individual computation in a Deep Learning system is a simple piece of numerical arithmetic—but where, and how, do these

computations combine to represent the concepts “*I’m writing a haiku now*” or “*Brake for this pedestrian*”?

What does it actually mean to understand and interpret an AI system? In our work, we take a causal view of interpretability: an interpretation should give us some information about how a certain change in an input will cause the AI system to change its output behavior.

Our first technical contribution in the field of AI interpretability used this causal view of interpretability to benchmark and compare existing interpretability methods on an equal footing. Prior to our work, existing methods which aimed to interpret AI systems all expressed these interpretations differently, making it impossible to compare their effectiveness.

In our quest to benchmark interpretability, we contributed new mathematical formalisms and a framework for AI interpretability, as well as a new dataset which allows for the practical measurement of interpretability performance. Our results indicated that no existing interpretability methods were particularly effective, underscoring the importance of our contribution: we need to rigorously measure and benchmark the effectiveness of interpretations.

Motivated by this result, we sought to further improve the performance of interpretability methods. Our second contribution in this field interprets an AI system using a secondary AI system. Our work shows that this can be very effective, consistently leading to the best interpretations in our experiments.

Effectively interpreting AI systems is only a partial requirement to effective and ethical AI. We also need to make sure that whatever system we build is aligned with the intentions of the developer. Our contribution in the field of AI alignment consists of critically assessing existing training and alignment methods for AI. To our surprise, we discovered that state-of-the-art methods were grossly underspecified along some key dimensions. Experimentally, we discovered that this can severely impact how the system ends up behaving. We introduce new methods which account for this, both by producing better training data and more precise training algorithms, and we find this can greatly improve the outcome and precision of AI system training.

Both our work on AI interpretability and AI alignment is in part predicated on the concept of *counterfactual data*. Counterfactual data represents a “what if” scenario, it embodies *what would’ve happened if this or that were different* [Pearl, 2009b]. But there is a catch, counterfactual data does not really exist. That ship has sailed – we can’t go back in time and change things. Yet the echoes of “what if” can linger louder than the cries of “what was”, and the lingering of “what if” has deeply impacted our work on AI interpretability and alignment.

We begin in Chapter 1 by introducing the foundational concepts necessary to build an intuitive understanding of the current generation of AI, with an emphasis on Large Language Models and their transformative impact. We keep conceptual details to a minimum in these sections, reserving deeper explanations for later sections where they are directly relevant. Following this, Chapter 2 investigates the crucial question of AI interpretability, exploring how we can make complex models more transparent and comprehensible, and what we actually stand to benefit from

interpretable AI . In Chapter 3, we turn our attention to the equally critical issue of AI alignment, addressing how to ensure that AI systems act in accordance with the developer’s intention. Building on these discussions, Chapter 4 reflects on the overarching contributions of this work, tying together the insights and advancements presented throughout.

For readers seeking deeper technical details, Chapters 5 and 6 provide a detailed account of our methodological innovations in AI interpretability, while Chapter 7 presents our technical contributions to the field of AI alignment.

Publications and Contributions

The research output obtained during this PhD has been published in a scientific journal and multiple international conferences and workshops, and disseminated online through preprints, blog posts, or talks. Not all of these works are directly related to the main thesis of this dissertation and, as such, are not discussed throughout the rest of this dissertation.

Publications in International Journals

1. **Karel D’Oosterlinck**, Winnie Xu, Chris Develder, Thomas Demeester, Amanpreet Singh, Christopher Potts, Douwe Kiela, and Shikib Mehri (2024). “*Anchored Preference Optimization and Contrastive Revisions: Addressing Underspecification in Alignment*”. Accepted for Publication in: *Transactions of the Association for Computational Linguistics* (TACL 2025). Also presented at the Safe Generative AI workshop (part of NeurIPS 2024). <https://www.arxiv.org/abs/2408.06266>

Publications in International Conferences

* denotes equal contribution

1. **Karel D’Oosterlinck**, François Remy, Johannes Deleu, Thomas Demeester, Chris Develder, Klim Zaporozets, Aneiss Ghodsi, Simon Ellershaw, Jack Collins, and Christopher Potts (2023). “*BioDEX: Large-Scale Biomedical Adverse Drug Event Extraction for Real-World Pharmacovigilance*”. In: *Findings of the Association for Computational Linguistics: EMNLP 2023*. <https://arxiv.org/abs/2305.13395>
2. **Karel D’Oosterlinck**, Semere Kiros Bitew, Brandon Papineau, Christopher Potts, Thomas Demeester, and Chris Develder (2023). “*CAW-coref: Conjunction-Aware Word-level Coreference Resolution*”. In: *Proceedings of the Sixth Workshop on Computational Models of Reference, Anaphora and Coreference*. <https://arxiv.org/abs/2310.06165>
3. Eldar David Abraham*, **Karel D’Oosterlinck***, Amir Feder*, Yair Ori Gat*, Atticus Geiger*, Christopher Potts*, Roi Reichart*, and Zhengxuan Wu* (2022). “*CEBaB: Estimating the Causal Effects of Real-World Concepts*”

- on NLP Model Behavior*". In: *Advances in Neural Information Processing Systems* 35. <https://arxiv.org/abs/2205.14140>
4. Zhengxuan Wu*, **Karel D'Oosterlinck***, Atticus Geiger*, Amir Zur, and Christopher Potts (2023). "*Causal Proxy Models for Concept-Based Model Explanations*". In: *Proceedings of Machine Learning Research*. Also presented as oral at the Counterfactuals in Minds and Machines workshop at ICML 2023. <https://arxiv.org/abs/2209.14279>
 5. Jing Huang, Atticus Geiger, **Karel D'Oosterlinck**, Zhengxuan Wu, and Christopher Potts (2023). "*Rigorously Assessing Natural Language Explanations of Neurons*". In: *Proceedings of the 6th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*. **Best Paper Award**. <https://arxiv.org/abs/2309.10312>
 6. Mathieu De Coster, **Karel D'Oosterlinck**, Marija Pizurica, Paloma Rabaey, Mieke Van Herreweghe, Joni Dambre, and Severine Verlinden (2021). "*Frozen Pretrained Transformers for Neural Sign Language Translation*". In: *18th Biennial Machine Translation Summit*, pp. 88–97. <https://aclanthology.org/2021.mtsummit-at4ssl.10>
 7. Houjun Liu, John Bauer, **Karel D'Oosterlinck**, Christopher Potts, and Christopher D. Manning (2024). "*MSCAW-coref: Multilingual, Singleton and Conjunction-Aware Word-Level Coreference Resolution*". In: *Proceedings of The Seventh Workshop on Computational Models of Reference, Anaphora and Coreference*. <https://aclanthology.org/2024.crac-1.4.pdf>

Preprints and Blogposts

1. **Karel D'Oosterlinck**, Omar Khattab, François Remy, Thomas Demeester, Chris Develder, and Christopher Potts (2024). "*In-context Learning for Extreme Multi-Label Classification*". In: *arXiv preprint arXiv:2401.12178*. <https://arxiv.org/abs/2401.12178>
2. **Karel D'Oosterlinck**, Thomas Demeester, Chris Develder, and Christopher Potts (2023). "*Flexible Model Interpretability through Natural Language Model Editing*". Presented at the Black Box NLP Workshop (EMNLP 2023). In: *arXiv preprint arXiv:2311.10905* <https://arxiv.org/abs/2311.10905>
3. Amir Zur, Elisa Kreiss, **Karel D'Oosterlinck**, Christopher Potts, and Atticus Geiger (2024). "*Updating CLIP to Prefer Descriptions Over Captions*". In: *arXiv preprint arXiv:2406.09458*. <https://arxiv.org/abs/2406.09458>

-
4. Atticus Geiger, Zhengxuan Wu, **Karel D’Oosterlinck**, Elisa Kreiss, Noah D. Goodman, Thomas Icard, and Christopher Potts (2022). “*Faithful, Interpretable Model Explanations via Causal Abstraction*”. Stanford AI Lab Blog. <https://ai.stanford.edu/blog/causal-abstraction/>

Talks and Lectures

1. *Are we still aligned? Tackling some fundamental issues in preference learning for AI* (2024). Talk at the 9th Generative AI Meetup, Ghent, Belgium.
2. *Anchored Preference Optimization and Contrastive Revisions: Addressing Underspecification in Alignment* (2024). Talk at the Stanford NLP Lunch sessions, Stanford, U.S.A.
3. *Building Efficient In-Context Learning Pipelines for Extreme Classification* (2024). Talk at Techtrust.ai, online.
4. *Building Efficient In-Context Learning Pipelines for Extreme Classification* (2024). Talk at BitsInBio Meetup, Ghent, Belgium.
5. *Improving Drug Safety Monitoring with NLP* (2024). Guest Lecture at the Catholic University of Leuven, 3rd year Applied Computer Science program, online.
6. *Conjunction-Aware Word-Level Coreference Resolution* (2023). Oral presentation at the Sixth Workshop on Computational Models of Reference, Anaphora, and Coreference, EMNLP 2023, Singapore.
7. *Causal Proxy Models for Concept-Based Model Explanations* (2023). Oral presentation at the Counterfactuals in Minds and Machines Workshop, ICML 2023, Honolulu, U.S.A.
8. *Improving Drug Safety Monitoring with NLP* (2023). Talk at the 19th Belgium NLP Meetup, Leuven, Belgium.
9. *Improving Drug Safety Monitoring with NLP* (2023). Talk at IDLab Day, Ghent, Belgium.
10. *Large Language Models and Explainability* (2023). Talk at OMP Data Science Tech Sessions, virtual.
11. *GPT-4 and Beyond: Key Concepts for Large Language Models and Impacts on the Banking Industry* (2023). Talk at Santander Nordics Data Conference, Oslo, Norway.
12. *CEBaB: Estimating the Causal Effect of Real-World Concepts on NLP Model Behavior* (2022). Talk at the Stanford NLP Lunch sessions, Stanford, U.S.A.

Overview of Contributions

This thesis expands upon three of the previously outlined papers, chosen for the significance of their contributions and their coherent narrative. These papers represent key advances in AI interpretability and preference optimization, and together, they form a comprehensive investigation into understanding and improving machine learning model behavior.

The three papers discussed in this thesis are:

- *CEBaB: Estimating the Causal Effects of Real-World Concepts on NLP Model Behavior*,
- *Causal Proxy Models for Concept-Based Model Explanations*, and
- *Anchored Preference Optimization and Contrastive Revisions: Addressing Underspecification in Alignment*.

Below, the contributions made in each paper are summarized, and a personal contribution statement is provided for each publication.

CEBaB: Estimating the Causal Effects of Real-World Concepts on NLP Model Behavior

This paper introduces a causal inference framework to evaluate the faithfulness of explanations of Deep Neural Networks. The key contributions include:

- Motivating the use of causal inference theory to assess explanation faithfulness in deep learning models.
- Developing a formalism that unifies various existing interpretability methods within a causal framework.
- Proposing a suite of metrics to quantitatively evaluate the faithfulness of interpretability methods.
- Collecting a real-world dataset through crowdsourcing to empirically assess interpretability methods within Natural Language Processing (NLP) models.
- Publicly releasing all code and data to facilitate further research in this area.

As co-first author, I was heavily involved throughout almost every aspect of this work. Specifically, I intensely collaborated with the other co-first authors on the creation of the formalisms, the standardization of interpretability methods, coding and running experiments, co-authoring the paper, analyzing the results, open-sourcing the data, and presenting the work at the conference where it was published. However, I was not involved in setting up the crowdworker annotation tasks that produced the dataset.

Causal Proxy Models for Concept-Based Model Explanations

Building upon the causal formalisms introduced in the first paper, this work develops novel training methods for improving AI interpretability techniques. The main contributions are:

- Designing two new training methods specifically tailored for AI interpretability.
- Establishing a comprehensive suite of comparable baselines to rigorously evaluate these methods.
- Validating the effectiveness of these novel training methods and baselines on the same NLP use case introduced in the previous work.
- Ensuring reproducibility by publicly releasing all associated code.

As co-first author, I was heavily involved throughout every aspect of this work and all the contributions listed above, in intense collaboration with the other co-first authors. I was solely responsible for presenting this work at the conference in which it was published.

Anchored Preference Optimization and Contrastive Revisions: Addressing Underspecification in Alignment

This paper addresses the problem of underspecification in preference optimization by introducing new techniques for preference data generation and optimization. The core contributions include:

- Introducing a novel data-generation method for preference learning.
- Proposing a new family of optimization techniques for direct preference learning.
- Providing motivation for these methods and demonstrating their advantages.
- Conducting extensive experiments comparing these new techniques against state-of-the-art models.
- Publicly releasing the newly generated datasets and all related code.

As sole first author, I was responsible for every aspect of this work and all the contributions listed above, with the other authors providing valuable guidance and advisory support throughout.

1

Introduction

1.1 Artificial Intelligence

Humans build and use tools. We've developed hammers, sewing machines, and airplanes to manipulate our physical world. Likewise, we've developed calculators, books, and the internet to manipulate our cognitive world. This tool-use has been profoundly useful and new tools have routinely increased human productivity and shaped our understanding of the world.

Artificial Intelligence (AI) is the new frontier for tool-building. Certain problems proved extremely challenging to tackle with conventional tools. Consider the invention of the programmable calculator, with which the tedious execution of arithmetic could be automated at unforeseen scales and cost-efficiencies. Its widespread adoption freed humans from countless hours of boring arithmetic work, and the new capabilities it unlocked sparked breakthroughs in virtually every scientific field. While arithmetic became the domain of machines, the design and use of these machines remained a uniquely human endeavor. A calculator does not solve problems in and of itself; some *intelligence*¹ is required to translate a problem to programming logic, decide on approximations and initial conditions, use the calculator, interpret the result, and iterate on the solution if necessary. Intelligence is a bottleneck, and it is this bottleneck that the field of Artificial Intelligence seeks to alleviate—producing tools that are more sophisticated and efficient than ever before.

¹The Merriam-Webster Dictionary defines intelligence as "the ability to apply knowledge to manipulate one's environment or to think abstractly as measured by objective criteria". The Oxford Advanced American Dictionary defines intelligence as "the ability to learn, understand and think in a logical way about things; the ability to do this well".

1.1.1 Impact

The potential impact of Artificial Intelligence is hard to overstate. AI tools are already making tremendous impacts in high-value task such as drug discovery [Jumper et al., 2021], computer chip design [Mirhoseini et al., 2021], and education [Wang et al., 2024a]. In November 2021, when I embarked on this PhD journey, being able to write good computer code was an essential requirement for any successful PhD in this field. Just three years later at the end of this PhD, Artificial Intelligence has matched or even eclipsed the code writing ability of the average PhD student [Li et al., 2022]. Today, I spend almost no time writing code anymore, and I spend more time thinking about what my code should look like, how it should handle edge-cases, and communicating these thoughts to an AI system. As a result, producing software has never been more accessible across both academia and industry, significantly lowering the barrier to innovation over the course of just a few years. It is hard to be pessimistic about the prospects this presents.

These paradigm shifts extend beyond the fields of engineering, math, and science. Before the advent of the most recent AI tools, virtually every tool used by a writer, painter, or cinematographer had already undergone some significant technological innovation. Paint became cheaper and digital word editors streamlined the writing process. Yet ultimately, the pen (or brush) was still put to paper by a human; AI is leading us to rethink this requirement [Brown et al., 2020, Ramesh et al., 2021, Singer et al.]. AI tools are significantly making many of these disciplines more accessible, allowing individuals with less conventional training or resources to still render their artistic vision to text, image, or video. It is very hard to not be excited about a world in which language barriers are trivially transcended and many more students have access to personalized tutoring.

1.1.2 Issues

A technological shift of this magnitude is sure to cause at least a few hiccups. One of the most pressing practical issues in AI right now is arguably the crisis in AI evaluation methodology [Kiela et al., 2021]. The current wave of more-generalist AI displays a wide range of competent behaviors across various open-ended domains. As a consequence, chatbots (such as ChatGPT or Claude), image-generator (such as StableDiffusion and Dall-E), and various related systems are incredibly hard to evaluate: we need to devise adequate tests across a plethora of domains and we need to increasingly rely on the most expert humans to come up with tests which match the capabilities of these systems – this is neither simple nor cheap. Consequences of ill measurements may be disastrous as accurately measuring the strengths and weaknesses of AI systems is vital to inform AI policy, ensure safe deployment, and motivate designs for the next generation of systems. The benchmarks that we do have are often quickly saturated and can become obsolete in a matter of months, forcing us to continually invest in more pertinent tests. This investment has generally not kept pace with the capabilities of these systems.

Powerful technologies have profound ethical implications. It is not clear to what

extent these tools will be used for misinformation and other nefarious purposes. It is not clear how AI developers can best guard against misuse, and who gets to decide (or regulate) what constitutes as a misuse. It is not clear where the data used to develop proprietary AI models originated from [Groeneveld et al., 2024], and who actually holds the rights to this data [Doe v. GitHub, 2022]. It is not clear how the labor markets will react to widespread adoption [Eloundou et al., 2023]. It is not clear in which contexts we can safely deploy such systems and how we should best go about testing this [Ganguli et al., 2022], or how we should attribute blame when something goes wrong.

1.1.3 Interpretability and Alignment

Many of these questions are deeply entangled with our ability to interpret the inner workings of AI systems. Understanding what is going on internally can teach us more about what a system learned, if it can be safely adopted, and how we may want to further engineer it to our liking [Lipton, 2018]. Yet, at first, such a question may feel paradoxical. If we built the system, and we know how to run it on a computer, how can its inner workings remain a mystery to us? In the first part of this dissertation, we will tackle the AI interpretability question and explore its technical and more philosophical underpinnings. Along this journey, we will ponder the utility of interpretability, and point towards future research directions.

Our ability to precisely control AI systems carries profound ethical implications as well, and has been extensively explored in both technical research and philosophical inquiry [Gabriel, 2020]. Ensuring the alignment of an AI system with the intentions of the developer is challenging for some of the same reasons AI interpretability is vexing. In the second part of this dissertation, we will study the AI alignment problem, again combining both technical and more philosophical concepts. As a result of our study, we uncovered intuitive (yet previously overlooked) aspects of the alignment problem which have deep technical implications. As a result, our contributed methods greatly outperform previous state-of-the-art methods.

Both our work on AI interpretability and AI alignment is in part predicated on the concept of *counterfactual data*. Counterfactual data represents a "what if" scenario, it embodies *what would've happened if this or that were different* [Pearl, 2009b]. But there is a catch, counterfactual data does not really exist. That ship has sailed – we can't go back in time and change things. Yet the echoes of "what if" can linger louder than the cries of "what was", and the lingering of "what if" has deeply impacted our work on AI interpretability and alignment.

We begin in Section 1.2 by introducing the foundational concepts necessary to build an intuitive understanding of the current generation of AI, with an emphasis on Large Language Models and their transformative impact. We keep technical details to a minimum in these sections, reserving deeper explanations for later sections where they are directly relevant. Following this, Chapter 2 investigates the crucial question of AI interpretability, exploring how we can make complex models more transparent and comprehensible, and what we actually stand to benefit from

interpretable AI. In Chapter 3, we turn our attention to the equally critical issue of AI alignment, addressing how to ensure that AI systems act in accordance with the developer’s intention. Building on these discussions, Chapter 4 reflects on the overarching contributions of this work, tying together the insights and advancements presented throughout.

For readers seeking deeper technical details, Chapters 5 and 6 provide a detailed account of our methodological innovations in AI interpretability, while Chapter 7 presents our technical contributions to the field of AI alignment.

1.2 Deep Learning

Deep Learning (DL), a subset of Artificial Intelligence, focuses on developing computer programs that can learn to perform specific tasks without being explicitly programmed for each scenario [LeCun et al., 2015]. At its core, deep learning involves developing a model, which is a highly flexible computer program with the *potential* to perform useful tasks. Initially, this model starts as an unrefined system incapable of any meaningful behavior. To unlock this potential, practitioners collect and curate a dataset that exemplifies the desired input-output relationships the model should learn to replicate.

The process of developing a Deep Learning system begins by shaping the training data in order to reflect the specific behavior or task the model is intended to perform. Next, a learning algorithm is employed to train the model by fitting it to the provided data [Rumelhart et al., 1986]. This training process is iterative: the model’s internal computations are gradually adjusted to identify and internalize patterns within the data. With each iteration, the model improves its ability to map inputs to outputs based on the examples in the training dataset.

When the learning process is successful, the trained model has effectively internalized the input-output mapping present in the training data. Hopefully, this means the model can generalize its learned behavior to handle unseen inputs, applying the knowledge it gained during training to new and unfamiliar scenarios. In this way, deep learning enables the creation of systems capable of solving complex problems, from recognizing images and translating languages to predicting outcomes and driving autonomous vehicles, with minimal human intervention once the training is complete.

For many tasks, it has proven far more efficient to allow a model to learn a solution using representative data, rather than having a developer attempting to manually implement an approximate solution. In this sense, the role of a Deep Learning engineer is less about explicitly programming a solution and more about cultivating one—providing the right conditions, data, and structure for the model to learn autonomously. This shift toward learning-based approaches offers remarkable flexibility and scalability. Innovations in model architecture or learning algorithms can often be seamlessly applied across diverse tasks and domains, enabling breakthroughs to propagate with minimal task-specific adjustments. As a result, Deep Learning has driven a wave of consolidation in AI techniques, where many tasks

that once required highly specialized, hand-crafted solutions are now efficiently addressed using the same or closely related Deep Learning methods [Chung et al., 2024]. This universality has positioned Deep Learning as a cornerstone of modern AI development, which is why Deep Learning techniques are the ones primarily studied in this dissertation.

There is much to learn about Deep Learning, such as the nature of the training data, the specifics of the optimization procedure, intuitions behind how a system learns, and how Deep Learning can be applied to generate new data like text and images. However, we will defer detailed explanations of these concepts for now, as we will address them in Chapters 2 and 3 when they become relevant. To start, we will briefly outline what a Deep Learning model looks like in Section 1.2.1, providing a foundational starting point for our work.

1.2.1 Artificial Neural Networks

ANNs typically consist of three types of layers: an input layer, hidden layers, and an output layer [Rosenblatt, 1958, Yegnanarayana, 2009]. The input layer takes in raw data, such as a numerical representation of an images or text. Hidden layers perform a series of computations to transform the data and extract underlying features. The output layer then produces the final result, which can have various shapes. For example, in a classification task, the output might be a label which identifies whether an image contains a cat or a dog. In regression tasks, the output could be a continuous value, such as the probability of rain tomorrow. Neural networks can also be generative [Goodfellow et al., 2014, Sutskever et al., 2014], producing entirely new content; for instance, they can generate the next pixel in an image or the next word in a sentence.

In deep neural networks, billions of interconnected computations work together to transform one single input into one single output. Each computation in the network feeds into others, forming a large network of transformations that extracts increasingly abstract features from the raw input data. This interconnected system enables deep networks to handle highly complex tasks, from translating languages to generating lifelike images. All of these computations are learned through an optimization procedure and training data, none of them have been explicitly programmed.

2

Interpretability

Deep Learning systems are notoriously difficult to interpret. We can *observe* every computation within the system, yet we struggle to truly *understand* how these computations come together to form meaningful behavior. These systems are capable of performing impressive tasks, such as steering a car, writing poetry, or outlining mathematical reasoning, yet we don't fully grasp how these abilities are conceptually implemented. Every individual computation in a Deep Learning system is a simple piece of numerical arithmetic—but where, and how, do these computations combine to represent the concepts “*I'm writing a haiku now*” or “*Brake for this pedestrian*”?

It's actually not even clear if such concepts exist in these systems. Do Deep Learning models ever develop something akin to these human-interpretable concepts in their internal computations, or are they solving problems in entirely different ways? Do these systems learn deep, meaningful truths about the world, or are they simply performing superficial pattern recognition and exploiting hacks to complete tasks? To what point is this even a real distinction?

Even more puzzling is the question of influence: which specific training examples shaped the generation of a particular poem? And in what ways? Which parts of the internal computations are responsible for a certain error? Despite having complete access to every computation within these deep system, our understanding remains astonishingly shallow. This paradox of total observability yet profound mystery is both frustrating and captivating.

The sheer scale of Deep Learning only adds to the challenge. Making sense of millions of computations underlying a simple behavior is difficult enough—but trying to understand billions of computations driving a more complex, general-purpose chatbot trained on trillions of datapoints seems exponentially harder.

In this chapter, we expand on the critical topic of interpretability in Deep Learning systems. In Section 2.1, we outline, and critically assess, the motivations behind AI interpretability. In Section 2.2, we clarify and operationalize the broad concept of interpretability, providing a framework for thinking about what it means to interpret a system, and which trade-offs are inherent to this understanding. Building on this, Sections 2.3 and 2.4 introduce our key contributions: rigorous methods for assessing if an interpretation of a system was actually correct. In Section 2.5, we present our radical approach for improving interpretability: leveraging AI itself as a tool to analyze and interpret other AI systems. Finally, in Section 2.6, we summarize our findings, and in Section 2.7, we reflect on some of the paradoxes inherent to interpretability research.

2.1 The Utility of Interpretability

The lack of interpretability in Deep Learning systems presents significant challenges to their adoption in real-world applications, and we stand to gain much from interpretable AI [Lipton, 2018].

2.1.1 Interpretability in Critical Scenarios

In critical use-cases, it is essential to ensure (to the best of our abilities) that a system behaves as intended across a wide range of unseen tasks or domains. Since it is impossible to test for every possible scenario, we may want to rely on a deeper understanding of the inner workings of a Deep Learning system to guarantee that its behavior aligns with expectations. The ability to audit a system and identify what went wrong when mistakes occur is equally important.

For critical use-cases, it is often argued that trust in AI systems requires both a deep insight into the internal operations of such a system, and robust capacity to audit these [Bereska and Gavves, 2024]. This is how critical software is traditionally developed today—designed to be understandable by humans, debuggable, and often even mathematically verified to behave as intended across a range of conditions. The adoption of AI in critical use-cases marks a significant departure from these expectations, which understandably raises concerns and skepticism.

2.1.2 Counterarguments for Interpretability

While the potential benefits of interpretability are clear, it's important to contextualize our desire for AI interpretability. In many cases, a less stringent approach may be justified. If a system passes rigorous behavioral tests, why bother understanding the complexities of its internal workings? The Wright brothers' first powered flight in 1903 was made possible through careful observation, trial and error, and empirical validation—not by a complete understanding of the theory of aerodynamics, which was not even fully developed at the time. Through careful engineering, complex

systems can be made fool-proof *enough* without a full grasp of every detail of their operation.

Humans are also not really interpretable. We like to think we are, but we really are not. When your friend is telling you a story, you have no way of knowing exactly what is going on at a conceptual level in their brain [Descartes, 1984]. Yet, this does not stop us from cooperating with friends and colleagues or trusting them in critical situations. Through repeated behavioral observations, we come to trust that their internal processes are broadly aligned with our own, and that they will handle unforeseen situations as best as they can. This trust is not perfect, but we build up a tolerance for human error. In extremely critical contexts, we strive to build a fail-safe system around human intelligence instead, making the overall system safer and more performant.

These stories offer some push-back against the strong case for AI interpretability, which is often assumed in interpretability research. We can build and eventually come to rely on AI systems without fully understanding their inner workings. Rigorous testing across various conditions will form the foundation of this trust, and we will come to accept that, like humans, these systems are not guaranteed to be perfect. Following this line of thought, it seems that interpretability is not a required condition for deploying useful and trustworthy AI systems in most domains.

This is not to say that interpretability is not valuable, we should prioritize making AI systems more interpretable whenever possible, but it does not always have to be a stringent requirement.

2.1.3 Other Interpretability Benefits

AI interpretability can offer benefits beyond the practical goals for safety or auditability of systems. From a purely scientific perspective, these highly complex Deep Learning systems are fascinating objects of study in their own right. The field of interpretability can drive the development of mathematical and statistical tools necessary to state and validate claims about the inner workings of these systems, without a direct link to any practical use-cases.

Furthermore, there is potential for cross-pollination between the study of natural intelligence and artificial intelligence. Studying natural intelligence is challenging due to biological and ethical constraints. For example, it's extremely difficult to observe the human brain in action, and systematically manipulating its environment is even harder. Once your friend has told you a story, you can't rewind their internal state and place them in slightly different conditions to study how their cognitive process changes. Moreover, many interventions that could provide deeper insight into someone's internal workings would be ethically problematic. In contrast, AI systems can be endlessly probed, perturbed, and manipulated, making the systematic study of their capabilities far more accessible. This might allow researchers to explore hypotheses about how natural intelligence works, although there is no guarantee to which degree artificial and natural intelligence correspond.

Finally, the study of AI interpretability can protect us from false or misleading claims about these systems. If someone asserts that "*this AI will do X in situation*

Y” or that “*computation X represents concept Y in the AI’s reasoning,*” we need the tools to critically evaluate such statements and determine whether they are valid—or even possible to verify. Without this ability, we risk being misled by a false sense of understanding, which could lead to reckless decisions and potentially harmful outcomes. The study of AI Interpretability can thus help ensure that our trust in AI systems is grounded in reality, not misplaced assumptions.

2.2 What are Interpretations?

What does it actually mean to understand and interpret an AI system? We navigated the previous section without a rigorous definition, so clearly we already rely on some intuitive notion of what understanding entails. However, rigorously defining what it means to understand a system is far from straightforward: as outline above, AI interpretability encompasses many distinct goals.

2.2.1 The Need for Abstractions

One could summarize all the billions of computations which make up an AI system with a concise statement such as “*This system processes pictures and labels all images containing a dog*” or “*This chatbot provides helpful answers to finance-related questions.*” These descriptions are undoubtedly useful, as they reveal a high-level behavior that would be difficult to deduce from the raw computations alone. In that sense, such a statement serves as an interpretation of the system, making its behavior more comprehensible. However, these statements are also highly abstracted and provide no actionable insight into the system’s internal mechanisms or how it achieves these outcomes. Additionally, these statements aren’t always faithful—a chatbot can often give unhelpful answers as well.

In contrast, an exhaustive list of all raw computations provides a perfect-fidelity representation of a system’s internal workings—but not at a meaningful or useful level of abstraction. Therefore, it is clear that a good interpretation needs to strike a balance: it should abstract away enough details to be practical while still offering non-trivial and faithful insights into the system’s general behavior [Geiger et al., 2021]. It is important to realize that any abstraction inevitably sacrifices some degree of fidelity to the underlying system. As a result, all interpretations are wrong, but some are useful nevertheless.

2.2.2 Conceptual Abstractions

The degree to which an interpretation is useful also greatly depends on who is the interpreter. A hypothetical super-intelligent entity might fully comprehend all the raw computations of a system without the need for any additional abstractions. For human interpreters, however, an intermediate level of abstraction tends to be optimal [Kim et al., 2018].

When auditing an AI system in a self-driving car, we might ask why the system ran a red light in a specific scenario. Explanations framed in terms of concepts like "*the speed of the car behind*" or "*whether there was a pedestrian in the crossing ahead*" feel intuitive and relatable. Ideally, such concept-level explanations provide meaningful insights into the system's decision-making process without sacrificing too much fidelity to the underlying mechanisms. A truly useful interpretation should empower us to apply our own reasoning to evaluate whether the system's behavior was appropriate in a given context and, if not, to identify what needs to change at a conceptual level.

2.3 Measuring Interpretations

We already understand that all interpretations are flawed to some extent, but some are more flawed than others. Interpretations need to be as faithful as possible given a certain level of abstraction. Knowing this, how would one go about testing the accuracy of a given interpretation?

2.3.1 Causal Estimates

If someone claims that an image classifier predicted "*dog*" for a picture because it "*detected a tail and paws*," what does this statement really mean? Essentially, they are asserting that "*the detection of a tail and paws caused the AI system to predict a dog*."

This causal framing provides a pathway to test the validity of the interpretation [Pearl, 2009a]. Hypothetically, you could revisit the image in question and modify it systematically to evaluate the role of the concepts in question. Experimentally, you might edit the image to include a cat's tail to see if it distracts the classifier, or crop out the paws entirely to assess their importance. After running such experiments, you might find that tails and paws were irrelevant, and the system's predictions actually depended on detecting features like ears or a snout. Alternatively, you might discover that no clear and satisfying interpretation exists that explains the system's behavior in terms of these concepts.

Regardless of the outcome, by framing an interpretation as a *causal* statement, we establish a method for verifying whether it truly reflects the underlying system behavior. This approach moves beyond speculation and provides a more rigorous framework for evaluating explanations.

2.3.2 Counterfactual data

More specifically, a causal interpretation can be verified by using *counterfactual* data [Pearl, 2009a]. Counterfactual data represents "*what if*" scenarios, such as "*what if the paws weren't visible*" or "*what if the pedestrian ahead of us was not yet crossing the street*." These hypothetical scenarios can verify if the interpretation faithfully represented the system. However, counterfactual data doesn't truly exist,

any data point we observe is (by definition) factual. If a pedestrian was already crossing the street, we cannot go back in time and observe what would have happened if they had not. The real counterfactual is forever out of reach [Holland, 1986].

Luckily, counterfactual data can be approximated [Imbens, 2015]. This is what we did in the earlier example by systematically altering the representation of certain concepts, like removing the paws or substituting them with other features. Using these approximations, we can empirically test the validity of causal interpretations and gain a better understanding of the system's behavior.

2.4 Benchmarking Interpretability Methods

What can we do with our newfound insights into AI interpretability? When we first engaged with this question, the field of AI interpretability was highly fragmented. Numerous desiderata were being actively debated in the literature, and a wide array of practical methods, all chasing a different flavor of interpretability, were competing for attention. There was little clarity on how to consolidate these methods or establish a grounded framework for assessing how well they interpreted a system.

To address this gap, we set out to create a more unified understanding, leveraging our insights into measuring causal interpretability through counterfactual data [Abraham et al., 2022]. In the sections that follow, we outline this work conceptually, leaving the full technical details for Chapter 5.

2.4.1 Formalizing Interpretability Methods

Since there was no universally agreed-upon mathematical definition for an interpretability method [Goyal et al., 2020, Künzel et al., 2019, Kim et al., 2018, Yeh et al., 2020, Ravfogel et al., 2020, Feder et al., 2021b], our first step was to construct our own formalism. We formalized concept-level interpretability methods through the lens of causal estimators [Pearl, 2009a], incorporating many of our insights described above. Conceptually, these causal estimators measure how much an AI system's output would change in response to a concept-level modification of its input [Abraham et al., 2022].

As illustrated earlier, such a method could estimate how much less likely an AI system is to predict "*dog*" if the paws in an otherwise typical picture of a dog are obstructed. In this way, a concept-level causal estimator quantifies the causal effect of the concept (e.g., paws) on the system's output.

Formally, we defined a concept-level interpretability method as a function that takes three inputs: (i) an AI system, (ii) a specific input to this system, and (iii) a description of a concept-level modification to that input [Abraham et al., 2022]. The method then outputs a predicted change in the system's behavior, effectively answering a "*what if*" question regarding the relationship between the concept and the system's output.

2.4.2 Standardizing Interpretability Methods

To study popular interpretability methods within our framework, we first had to retrofit these methods to correspond with our definition of a concept-level causal estimator. For instance, one widely used interpretability method sought to explain a system’s behavior by approximating it with a simpler, more interpretable statistical model [Künzel et al., 2019, Goyal et al., 2020]. While this method does not directly answer counterfactual questions, we can extract such predictions by manipulating the variables in the statistical model that correspond to the concepts of interest. The reasoning is straightforward: if the statistical model is a good approximation of the system, then concept-level changes in the statistical model should approximate the corresponding concept-level changes in the system itself.

2.4.3 Faithfulness metrics

To evaluate whether a predicted causal effect is faithful to a system, we need a metric to quantify how much the prediction corresponds with a real causal effect.

Classifier AI systems typically output a probability for each class. A causal effect for a classifier system will thus consist of the *difference in probabilities* for each class, representing the change in behavior due to the counterfactual scenario. Specifically, this difference in probabilities tells us how much more or less likely a particular class will become as a result of the change [Goyal et al., 2020].

We devised several new metrics to assess exactly how well a predicted change aligns with an actual change. These metrics measure different attributes of success, such as how well an interpretability method can predict *which classes* are likely to increase or decrease in likelihood, or *how much likelihood is displaced* across classes.

2.4.4 Counterfactual Dataset

To use these metrics, we needed to be able to actually measure a real change in the system behavior due to a concept-level change in the input. To do this, we created a large real-world dataset with many approximated counterfactuals [Abraham et al., 2022]. Specifically, we took written restaurant reviews and used human annotators to rewrite several parts of these reviews to alter the information they expressed about specific aspects of the dining experience. This process resulted in a dense clustering of factual restaurant reviews and counterfactual manipulations.

Each counterfactual review represented an answer to a specific *"what if"* question, such as, *"What would the original reviewer have written if everything in the restaurant was the same, but the service was worse?"*

2.4.5 Results

We trained several Language Models [Devlin et al., 2019, Liu et al., 2019, Hochreiter and Schmidhuber, 1997] to predict the 5-star rating associated with each restaurant

review. The task of an interpretability method was to estimate how the predicted rating would change due to a changing aspect in the input. We used the human-created counterfactuals to verify how the prediction actually changed.

We found that, across six popular interpretability methods [Goyal et al., 2020, Künzel et al., 2019, Kim et al., 2018, Yeh et al., 2020, Ravfogel et al., 2020, Feder et al., 2021b] and one simple baseline [Abraham et al., 2022], the baseline actually produced the most accurate interpretations. This was a striking and unexpected result, but it validated our initial instinct to consolidate and systematically compare different interpretability methods. Without a rigorous mathematical definition and an experimental framework to evaluate the effectiveness of these methods, there is little hope of developing adequate tools for interpreting AI systems. We saw our work as a rallying cry to approach AI interpretability with greater precision, rigor, and a commitment to validating interpretability methods before adopting them.

2.5 Learning to Interpret

Given the underwhelming performance of off-the-shelf interpretability methods, a natural question arises: can we do better? After all, a key motivation behind our experimental framework and data was to actually advance the field of AI interpretability.

2.5.1 Interpreting AI with AI

One unconventional approach is to use a second, auxiliary AI system as the interpretability method [Wu et al., 2023]. At first glance, this idea might seem counterintuitive or even controversial—it appears to shift the problem of interpretability to yet another AI system, which itself may be just as opaque as the original. However, the primary goal of an interpretability method should be to produce accurate estimates, regardless of how these estimates are created [Abraham et al., 2022]. Thus, it is perfectly valid to use a secondary AI system to estimate changes in a primary system [Wu et al., 2023].

2.5.2 Training an AI Interpreter

We implemented this approach by using a secondary AI system that takes the same factual input as the original system, along with a description of a "*what if*" scenario. This secondary system is trained to predict how the original system would respond under the hypothetical condition [Wu et al., 2023].

To train such a system, we used some of our counterfactual restaurant reviews as training examples [Abraham et al., 2022]. This way, we were able to calibrate the secondary AI system on some real examples of how the primary system responded to changes. We also calibrated existing interpretability methods with this counterfactual training data [Künzel et al., 2019], to make the comparison with our AI interpreter fair.

2.5.3 Results

Our AI-as-interpreter system produced much more faithful interpretations, better estimating how the original AI system would respond to a hypothetical situation compared to conventional interpretability methods. Our work pioneered the use of Deep Learning to interpret Deep Learning, and we rigorously validated this is effective. Additionally, we motivated the use of counterfactual data as an explicit training signal for interpretability methods.

2.6 Conclusions

We formalized off-the-shelf interpretability methods as causal estimators and leveraged approximate counterfactual data to evaluate the faithfulness of their interpretations, developing an experimental framework to support this evaluation. Our findings revealed that conventional interpretability methods performed worse than a simple baseline, underscoring the need for rigorous evaluation in the field of AI interpretability. To address this, we demonstrated how performance could be significantly improved by training a secondary AI system specifically to produce faithful interpretations.

The idea of causally evaluating interpretability methods is gaining traction within the interpretable AI community. For example, in May 2023, OpenAI released a paper in which a GPT-4 model was used to automate the interpretation of components of GPT-2’s computations [Bills et al., 2023]. In a subsequent project, we applied the same notions of causal interpretability to test these GPT-4 generated interpretations [Huang et al., 2023] (this project was not included in this dissertation, as I was not a lead author). Strikingly, our findings revealed that almost none of the interpretations held up under rigorous validation. This result underscores the critical importance of accurately measuring the validity of interpretability methods and serves as a cautionary tale: even in large, well-funded labs, it is alarmingly easy to mislead oneself into believing that interpretations are meaningful when they are not.

2.6.1 Future Work

The idea of training AI systems to interpret other AI systems is gaining significant traction in the field. An interesting approach involves the use of Sparse Autoencoders (SAEs) [Ng et al., 2011] to interpret neural networks. These SAEs are designed to take internal representations formed by an AI system and learn to disentangle the information encoded within these representations [Bricken et al., 2023]. This process offers a novel perspective on how such representations can be understood and analyzed. Notably, one of the first major efforts in this direction was spearheaded by Anthropic.

However, a critical challenge remains: the causal efficacy of these interpretations is not always rigorously evaluated. Simply producing a new perspective on an AI system’s internal representation does not necessarily mean that the perspective

is meaningful or actionable. For example, even if an SAE is able to disentangle a concept like "cat" from a particular internal representation, it is essential to assess whether this disentanglement is causally faithful—that is, whether the interpretation accurately reflects the underlying mechanisms of the system. As discussed in Section 2.3, such causal faithfulness is a key metric for evaluating interpretability methods.

Beyond merely measuring the effectiveness of SAEs, a more ambitious goal would be to train them to generate even more insightful or actionable interpretations. This could potentially be achieved by incorporating counterfactual data, as suggested in Section 2.5. By exposing SAEs to counterfactual scenarios, we could refine their ability to produce interpretations that not only align with the underlying system but also provide practical utility in understanding and controlling AI behavior.

2.7 The Paradox of Practical Interpretability?

We began our study of interpretability by outlining several motivations for interpretable AI. Some of these motivations are practically oriented, such as understanding potential failure modes in a system, assessing its safety for deployment, or allowing stakeholders to build trust in the system's behavior.

Through our work, we have come to understand that interpretations are inherently abstractions of the underlying system. To ensure that an abstraction is meaningful, we must rigorously validate its efficacy. This implies that whenever we aim to use interpretability for a practical purpose, we must dedicate additional effort to evaluate whether the interpretations are accurate themselves. Ironically, the effort required to validate practically-motivated interpretability could be redirected toward direct behavioral testing, potentially achieving the same practical goals without relying on interpretability.

This tension resembles somewhat of a paradox: it is not clear whether interpretability offers unique value for practical applications that could not be achieved through more direct methods. Lipton (2017) discusses several facets of AI interpretability, highlighting interpretability's primary motivation as the ability to capture aspects of a system's behavior that are otherwise difficult to measure. Yet our findings underscore the necessity of validating interpretations, which in and of itself requires measurement.

This is not to say that interpretability has no value; rather, it is important to contextualize its role as a solution to practical problems. We discussed its practical applications, as well as other, less practical desiderata, such as the purely scientific value of interpretability as a field of study. Exploring interpretability for its potential to deepen our understanding of AI systems and their internal mechanisms remains a compelling and worthwhile direction, even when immediate practical utility could be limited.

3

Alignment

The exact behavior of a Deep Learning system emerges as the outcome of an optimization process. This process is directly guided by the Deep Learning engineer, who determines the objective of the optimization, curates the data it operates on, and designs the system's architecture. Through these choices, the engineer attempts to encode their intended goals for the system into the optimization process, aiming to produce desirable characteristics and behaviors in the resulting model. However, this process only *indirectly* shapes the system's actual behavior. If the engineer does not have direct control over the emergent behavior, how can we ensure that it is *aligned* with their intended goals [Gabriel, 2020]? Studying this question is an important focal point for both engineering and philosophy, given the broad potential impact of AI and the cost of misalignment.

Section 3.1 examines the impact of AI alignment, while Section 3.2 discusses methods for detecting alignment issues. Section 3.3 explores some potential causes of alignment issues, and introduces the concepts of imitation learning and preference learning. In Section 3.4, we motivate our new approach to training data for improved alignment, while Section 3.5 introduces our novel optimization procedures to better control alignment outcomes. In Section 3.6, we outline our experimental contributions and summarize our findings in Section 3.7.

3.1 Impact of Alignment

Every instance of imperfect behavior in a Deep Learning system reflects some degree of misalignment, since no engineer sets out to create a system that fails to perform as intended. While this broad perspective may lack precision, it underscores

how pervasive and impactful the alignment question is.

To illustrate, consider a system designed to optimize for mathematical problem-solving performance. There are numerous reasons why such a system might fail to meet expectations, including gaps in the training data, architectural limitations, or the exploitation of unintended shortcuts during optimization. Investigating these misalignments in depth is key to achieving better performance in the future.

In a more popular context however, the alignment problem refers to extreme (often fictional) scenarios in which powerful AI systems cause catastrophic outcomes due to unforeseen discrepancies between their intended and actual behavior. One illustrative example (which has become something of a cliché at this point) imagines a highly capable AI tasked with maximizing the output of a paperclip factory [Bostrom, 2014]. In its pursuit of this objective, the AI could theoretically decide to enslave or eradicate humanity, interpreting these drastic measures as the most effective way to achieve its goal. A slightly more elaborate exploration of such dilemmas can be found in Isaac Asimov’s science fiction classic *I, Robot*, where advanced AI systems with seemingly straightforward objectives consistently produce unintended, undesirable outcomes [Asimov, 2004]. These cautionary tales highlight the critical importance of ensuring AI systems remain aligned with human values and intentions, even as their power and autonomy grow. Currently, the more doomsday-oriented aspects of alignment firmly remain in the realm of science fiction. Yet, with the rapid advancements in general-purpose Artificial Intelligence, considerable resources have been devoted to studying these scenarios both practically and philosophically [Center for AI Safety, 2024].

Regardless of such long-term concerns, there are immediate and tangible harms associated with the deployment of these technologies today. AI systems can already be used to amplify misinformation, enable hacking, and facilitate plagiarism [Bommasani et al., 2021]. As a result, there is significant value in addressing alignment challenges now—both to enhance the long-term development of these systems and to mitigate the specific harms they can pose in their current forms today.

3.2 Detecting Alignment Issues

First, the question arises of how and when we can detect alignment issues to guide system deployment and implement safeguards against worst-case outcomes. This concern closely parallels the question of interpretability’s utility, as discussed in Chapter 2. Once again, there appears to be no silver bullet for understanding precisely what a Deep Learning system is doing and ensuring its harmlessness in unforeseen environments—it’s very hard to account for unknown unknowns. As a result, the most pragmatic approach seems to be extensive behavioral testing [Ganguli et al., 2022]. Through targeted testing, we can develop sufficiently deep insights into the system’s behavior across a wide range of representative environments. While some level of uncertainty will always remain, this approach allows us to make informed decisions, trusting that the benefits of deploying the system will outweigh the potential risks.

3.3 Preventing Alignment Issues

While behavioral testing seems the most fruitful strategy for detecting and mitigating alignment risks, the natural next question is how—and to what extent—these scenarios can be prevented in the first place. Whether we are addressing immediate harms or preparing for long-term risks, the technical challenge remains fundamentally the same: the designer intended the system to be safe and effective, yet it falls short of this ideal. Something in the process—whether in the design, training, or deployment—must have gone amiss [Amodei et al., 2016]. Identifying and addressing these points of failure is crucial to ensure that AI systems behave as intended. In order to understand how these points of failure arise and how they can be addressed, we need to first delve into imitation learning and preference learning.

3.3.1 Imitation Learning

Imitation learning has typically been the dominant optimization paradigm for Deep Learning systems [Hussein et al., 2017]. As the name suggests, imitation learning adjusts the system to mimic the patterns in the training data as closely as possible.

Unfortunately, data is never perfect. Even in high-quality datasets, undesirable or irrelevant patterns often persist, often deeply intertwined with useful ones, making them hard to eliminate entirely. This challenge becomes especially pronounced in open-ended tasks with modern AI systems, such as authoring books or composing music, where no single demonstration is ever truly perfect—each contains both desirable and undesirable patterns. Imitation learning inherently forces us to accept both the good and the bad, potentially causing alignment problems or otherwise suboptimal training outcomes.

3.3.2 Preference Pairs

Data is never perfect, but some datapoints are better than others—every demonstration has some less desirable patterns, but some demonstrations are more desirable than others. By comparing two data points, we can isolate specific patterns from the whole. Instead of learning from a single output per input, as is done in imitation learning, we can focus on the *difference between two valid outputs*. Rather than learning from one good example, we can learn why one example is *preferred or better than* another [Christiano et al., 2017].

In this comparison-based approach, any pattern that does not meaningfully differ between the paired examples did not contribute to our preference. This gives us an improved ability to point to specific patterns within data points and express if we think they are desirable for a system to internalize. Such preference-pairs contain far more nuanced information than unpaired examples, enabling us to be much more specific and precise with our optimization.

In some domains, it is cumbersome for a human to demonstrate desirable behavior directly, but much easier for them to select the better of two AI-generated demonstrations. For instance, the AI might attempt a certain video game scenario

multiple times, and the human can select the most successful trial. The AI is then optimized to perform more like the selected demonstration. In this context, we say the AI is *reinforced* based on the reward signal provided by the human [Sutton and Barto, 1999].

The application of *Reinforcement Learning from Human Feedback* (RLHF) [Christiano et al., 2017] to Large Language Models has been a key factor in the success of systems like ChatGPT [Ouyang et al., 2022]. Given the complexity of the behaviors we want these systems to exhibit, sourcing human demonstrations can be challenging and resource-intensive. Instead, having human experts compare multiple AI-generated attempts and select the best one can be a more scalable and effective approach. RLHF has also played a critical role in training AI assistants to be safe, harmless, correct, and helpful [Bai et al., 2022]. These attributes are often difficult to demonstrate through singular examples—and may even conflict with each other—but paired comparisons can effectively capture subtle distinctions, such as one response being more helpful or accurate than another.

The success of RLHF in ChatGPT has established preference-learning as a cornerstone technique, alongside imitation learning, for nearly all state-of-the-art general-purpose AI systems [Bai et al., 2022, Ouyang et al., 2022, Dubey et al., 2024]. The field has quickly iterated on many aspects of RLHF and explored diverse ways to source and learn from preferences. Notably, these preferences do not always need to be created by humans. Instead, a more advanced AI system can be employed to express preferences, a method known as Reinforcement Learning from AI Feedback (RLAIF) [Bai et al., 2022]. This approach works particularly well when the AI system making the judgments has an advantage over the system generating the demonstrations, for instance, by being more powerful or using additional resources such as human-designed rubrics to guide its decisions [Gu et al., 2024].

3.3.3 Preference Learning

Learning from these preference-pairs, however, necessitates a fundamentally different optimization strategy than the standard approach used in imitation learning. Generative Deep Learning systems, such as Language Models, are traditionally optimized to produce their output step by step [Radford and Narasimhan, 2018]. Specifically, Language Models are trained to predict the next part of a word, known as a token, given all the tokens that came before it. This token-level optimization reduces the challenge of imitating an entire example, such as a sentence or paragraph, into a series of smaller, manageable tasks: learning to predict and imitate one token at a time [Sutskever et al., 2014]. Once trained, the model generates sentences by iteratively predicting the next token based on the sequence so far, continuing this process until a stopping condition, like the end of a sentence or paragraph, is reached.

Unfortunately, token-level imitation learning falls short when working with paired data [Ouyang et al., 2022]. Consider two poems that share the same theme but are expressed with completely different phrasings. Even if one poem is preferred

over the other, it would be nonsensical to claim that the fifth word in the first poem is inherently better than the corresponding word in the second. Similarly, two mathematical proofs may differ only slightly in their phrasing, yet some of these differences might be inconsequential, while others could fundamentally alter the proof's validity. In general, preference-pairs cannot be easily reduced to token-level preferences, unlike imitation learning examples, which can be broken down into token-by-token tasks. Since Language Models inherently generate output token by token, we must ultimately find a way to bridge the gap and adapt preference learning into a form that corresponds with this token-level generation process.

One empirically successful method for learning from preferences involves the use of a secondary AI system, referred to as a Reward Model, to extract insights from a large set of paired comparisons [Christiano et al., 2017, Ouyang et al., 2022]. Ideally, this Reward Model captures the underlying qualities that make one example better than another, effectively internalizing the concept of "goodness." Once trained, the Reward Model can estimate the utility, or "goodness," of partially completed outputs. This effectively translates the example-level preferences into token-level utilities. This estimated reward can then be used to guide the initial AI system, steering it towards generating the next token that is more likely to result in higher overall utility [Schulman et al., 2017]. This was the methodology used in the initial application of Reinforcement Learning from Human Feedback to ChatGPT.

The additional effort of training a secondary AI system makes this method cumbersome. In response, *Direct Preference Optimization* (DPO) was developed [Rafailov et al., 2024b]. At a high level, the DPO objective states that the optimization procedure should increase the likelihood of generating the better example more than the worse example in the pair. While the likelihood of an entire example is derived from the product of likelihoods for each next-token prediction, the DPO objective does not dictate how the likelihoods of individual tokens should shift [D'Oosterlinck et al., 2024]. Instead, it only requires that the overall likelihood of the preferred example increases more during training than that of the less-preferred example, leaving token-level adjustments flexible.

Through the lens of imitation learning, DPO essentially encodes that the better example should be imitated more than the worse example during training. Empirically, this approach has proven to be highly successful [Dubey et al., 2024], offering a more practical alternative to the more cumbersome method of using a Reward Model. Many derivatives of this objective have been explored and have found widespread success in both industry and academia [Wang et al., 2024b].

Creating preference-pairs and effectively learning from them unlocks more precise Deep Learning training, making it an invaluable piece of the alignment puzzle. In the field of Large Language Models, these techniques have become almost synonymous with the more abstract goal of alignment. The success of alignment techniques has been extensively verified empirically across a variety of tasks [Dubey et al., 2024]. Virtually every chatbot you've interacted with has undergone such an alignment step, in addition to the more conventional pre-training and fine-tuning processes [Dubey et al., 2024, Bai et al., 2022, Ouyang et al., 2022].

3.4 Precise Preferences

When we began studying the alignment problem, the field was eager to get their hands on all kinds of preference-pairs. However, upon inspecting some of these preferences [Cui et al., 2024], we quickly realized that not all preferences are created equally. *The ability of pairs to convey more nuanced signals does not necessarily mean they always do.* In preference-learning, the *underlying comparisons* need to be of high quality for the preference to make sense [D’Oosterlinck et al., 2024]. In preference-learning, you can have a low-quality comparison between two high-quality examples, or a high-quality comparison between to low-quality examples. This is in stark contrast to the imitation learning paradigm, where the unpaired datapoints themselves needs to be of high quality [Hussein et al., 2017]. While some comparisons can provide valuable insights, others can be entirely uninformative or even misleading.

3.4.1 Minimally Contrastive Preferences

We asked ourselves the question: "What makes a comparison useful?" If I tell you that I much prefer to eat pizza to cobblestone, you’ve gained very little insight into my dietary preferences. However, knowing that I prefer my pizza wood-fired with a thin crust provides much more meaningful information. In both cases, the preference is valid – I like one thing more than the other – but the nature of the underlying comparison differs greatly [D’Oosterlinck et al., 2024]. When the items being compared are drastically different, we might be comparing apples to oranges (metaphorically speaking; literally comparing apples to oranges would actually be valid in the context of our dietary example). Ideally, a preference-pair should highlight a clear distinction along a few dimensions while keeping all other factors constant. We call such a comparison *minimally contrastive*. This insight turned out to be a key contribution of our work on alignment.

Our concept of minimal contrasts is closely related to the notion of counterfactual data [Pearl, 2009a], which we explored in Section 2.3 to study and improve interpretability methods for Deep Learning systems. Counterfactual data seeks to answer "*what if*" questions by keeping all but one concept fixed and observing what changes. Certain what-ifs simply cannot be answered with factual data alone, as the context needed to answer them is purely hypothetical and not necessarily represented in factual data [Holland, 1986]. The same principle applies to preference-pairs: if the two examples differ too widely, it becomes impossible to determine *which specific difference* led to one being preferred. Even worse, preference-pairs can contain spurious differences that undermine learning, much like spurious patterns in unpaired data can disrupt imitation learning [Geirhos et al., 2020]. If one example in the preference-pair would be a counterfactual of the other pair, their comparison would be incredibly nuanced and the amount of spurious differences would be minimized.

This important dimension of contrastiveness was not taken into account in previous work, and as a result, existing preference datasets can vary widely in

their contrastiveness. For example, people can make preference judgments by comparing two answers coming from either the same AI system or from different AI systems. Statistically, we can expect both answers to be more comparable when they originate from the same model. Additionally, people are collecting preference data from the internet using various assumptions. For instance, the preferences mined from targeted Wikipedia edits will differ widely from preferences created by comparing the number of upvotes on internet comments responding to the same post [Ethayarajh et al., 2022]. We contributed the idea that this minimal contrastiveness is an important part of the alignment story.

3.4.2 Revisions

Now that we understand the preference comparison needs to be relevant, how can we go about obtaining such high-quality preferences? One approach is to ensure that the initial demonstrations, which will later be judged to select the preferred one (either by a human or an AI system), are minimally contrastive from the outset [D’Oosterlinck et al., 2024]. This could involve designing a filtering procedure to identify and select demonstration pairs that embody clear, meaningful differences while minimizing irrelevant variability.

Alternatively, we could actively construct minimally contrastive preferences ourselves. Instead of presenting two demonstrations to a judge for comparison, we can create a preference-pair by *minimally revising* one demonstration to improve it. The original demonstration and its revision form a preference pair where the revision is the better demonstration. These revisions could be performed by a human annotator or another AI system, just like judgments can be performed by both humans or stronger AI systems. In our work [D’Oosterlinck et al., 2024], we decided to use an AI reviser to create minimally contrastive preference pairs through targeted revisions, introducing the method of *Contrastive Learning from AI Revisions* (CLAIR). Using an AI reviser offers greater control over the data creation process and is far more cost-effective than relying on human annotators for revisions.

Empirically, we validate that the preferences produced by CLAIR indeed form better contrasts compared to a conventional judgment-based method. Conceptually, there is another reason why CLAIR preferences may be more desirable compared to judgment-based preferences. If an AI system initially performs really poorly on a task, almost all of its demonstrations will be incorrect. It may take many iterations of judgments and training before the system becomes slightly better. In contrast, revisions allow a flawed example to be significantly improved in a single step. This means the resulting preference pairs generated through revisions are likely to contain higher-quality examples on average, potentially accelerating the learning process.

Any additional effort we invest in obtaining better preference data must justify the incremental gains it brings. This is well summarized by *"the bitter lesson"* of Machine Learning: simple methods that scale well with data and compute tend to outperform more advanced methods that do not [Sutton, 2019]. Fortunately, CLAIR

is not significantly more expensive than previously used alternatives for preference data. Producing revisions via an AI system or human annotator can even be more cost-effective: sometimes it is easier to fix one mathematical reasoning instead of comparing two distinct reasonings.

Starting from a critical analysis into the information conveyed by a preference-pair, we have advanced the field’s general understanding of preference learning by emphasizing the importance of meaningful contrasts in preference data. Our method, Contrastive Learning from AI Revisions (CLAIR), produces higher-quality preference data at comparable costs to existing approaches.

3.5 Anchored Alignment

While engaging with this work on alignment, we discovered that this important dimension of contrastiveness was not the only thing missing in alignment work. There is more we can learn from investigating simple better-than preferences.

3.5.1 What You Know Impacts How You Learn

High-quality contrasts are necessary, but the relationship between the preferences and the system needs to be accounted for as well [D’Oosterlinck et al., 2024]. If you already understand that humans typically don’t enjoy eating rocks, my preference for pizza over cobblestone is redundant. However, if you lack any knowledge about humans, this preference might actually be quite insightful. The utility of the information a system can extract from a preference—and how to optimize this system—is deeply entangled with how much the system already knows.

To us, this feels intuitive. We know that different students will respond differently to the same information. It is important to stress that this is *not* the intuition embodied by the conventional imitation learning paradigm. In imitation learning, we collect examples of *good* behavior [Hussein et al., 2017]. A core assumption here is that an infinite amount of good demonstrations will eventually cause any student to successfully imitate the desired behavior, regardless of initial competence (given the student has enough raw capacity to learn). Of course, stronger students will learn more efficiently, but in the limit of infinite data all students should learn meaningful behavior.

Preference-learning is fundamentally different. You could waste an eternity explaining the difference between Cajun and Creole cuisine to an amateur chef, if they don’t know how to fry an egg your subtle nuances will be lost. An infinite amount of high-quality comparisons does not guarantee good eventual behavior, the comparisons need to be relevant to the student [D’Oosterlinck et al., 2024]. This represents a significant departure from the intuition underlying the imitation learning paradigm.

If we want to optimize a system using preference-pairs, we can gather the underlying demonstrations from various sources: humans, other AI systems, or even directly from the system we intend to optimize [Cui et al., 2024, Bai et al.,

2022, Ouyang et al., 2022, Dubey et al., 2024]. While all these approaches can produce meaningful preferences and interesting comparisons (especially if they are minimally contrastive), they vary in relevance to the system being optimized.

Demonstrations sampled directly from the target system, often referred to as *on-policy* data, are typically the most relevant since the resulting preferences are directly tied to the system’s existing capabilities. However, it is also valuable to extract insights from *off-policy* preferences—those derived from demonstrations generated by other sources, such as different AI systems or humans. Balancing these sources raises an important question: how does the relevance of the preferences influence the way the system should learn from them? At the time we engaged with this work it was already well-understood that on-policy data was generally more relevant than off-policy data, but nobody accounting for this relevance during the optimization procedure.

Humans naturally treat each learning opportunity relative to their current level of understanding. Imagine a classroom scenario where two answers to the same exam question are presented: one answer is good and one is even better. For a good student, understanding the difference between "good" and "great" is the most productive way forward. For a novice, the subtle distinction between "good" and "great" might be overwhelming, and simply learning to imitate either answer would be the most crucial first step. Conversely, an extraordinary student may find even the great answer lacking. While they can still extract useful insights from the comparison, imitating either answer would amount to regressing in their abilities. Here, the degree of imitation learning that is performed next to preference learning greatly impacts the outcome of the learning process.

The simple yet powerful notion—the optimal learning procedure is dependent on the relationship between a system and the preferences—was missing from conventional alignment objectives [Rafailov et al., 2024b]. Incorporating this principle into our work is one of our major contributions to the field.

3.5.2 Anchored Preference Optimization

To this end, we sought to develop practical preference learning methods to address this issue. First, we note that it is impossible to completely separate imitation learning from preference learning when optimizing a system. Even if the system exclusively learns from "*A is better than B*" comparisons, the training process ultimately adjusts the likelihood of generating either demonstration [D’Oosterlinck et al., 2024]. Therefore, the effects of preference learning can be examined by analyzing the degree of imitation or avoidance induced by training on specific demonstrations.

The Direct Preference Optimization (DPO) objective [Rafailov et al., 2024b], one of the most successful alignment objectives, encodes preference learning as "*imitate A more than B*". This fully captures our understanding that A is better than B, but it does not specify the absolute degree of imitation required on either A or B, leaving room for a lot of ambiguity in the optimization process.

Specifically, the DPO objective can be satisfied in three qualitatively distinct

ways. First, the system might "*imitate A a lot and imitate B a little,*" effectively leaning heavily into imitation learning for both examples but favoring A. Second, it could "*imitate A a little and avoid B a little,*" since avoiding is the opposite of imitating. Finally, the system might "*avoid A a little and avoid B a lot.*" In all three cases, the relative goal of more imitation for A is achieved, but the absolute amount of imitation for A or B varies significantly [D'Oosterlinck et al., 2024].

This distinction is critical in practice, as it directly relates to the differing needs of systems at varying levels of performance. Returning to our classroom analogy: a struggling student benefits greatly from substantial imitation learning on good data, while an exceptional student might prefer less reliance on imitation altogether. DPO and similar objectives fail to encode this nuance, leaving optimization agnostic to the system's current state and potentially misaligned with its actual learning needs.

We found that neglecting to tailor the optimization to the system's current capabilities was already leading to suboptimal outcomes in alignment efforts at the time we began this work. A notable example involves a popular preference dataset which was created using answers from a previous generation of AI systems [Cui et al., 2024]. Many practitioners were using this dataset to align newer, more advanced systems. Inadvertently, this placed the optimization in the "*extraordinary student*" situation, but this was not accounted for during optimization.

To address this issue, we introduced the concept of *Anchored Preference Optimization* (APO) [D'Oosterlinck et al., 2024]. APO consists of a family of alignment objectives similar to DPO [Rafailov et al., 2024b], but with the crucial addition of explicitly accounting for the absolute amount of imitation learning on either example in the pair. Based on the alignment situation at hand, a developer can select the correct objective and avoid much of the ambiguity inherent to DPO. This gives a practitioner much more control over the optimization procedure, leading to more stable and performant results.

3.6 Experiments

In order to validate our ideas, we conducted an empirical study to examine the impact of contrastive revisions and anchored alignment within a practical alignment use case. The technical details related to this analysis are provided in Chapter 7, while the rest of this section outlines these experiments conceptually.

We validate our methods by further aligning `llama-3-8B-instruct`, an advanced 8-billion-parameter open-source model by Meta [Dubey et al., 2024]. Already a highly capable general-purpose chatbot, we aim to enhance its accuracy and helpfulness even further. To study the impact of preference data on alignment outcomes, we constructed four datasets of preference-pairs from the same input queries. Two datasets were created using a judging paradigm, where a teacher model, `GPT4-Turbo`, selects the preferred answer from multiple demonstrations. `GPT4-Turbo`'s superior performance makes it a reliable substitute for human judgments in this context. We create one dataset using a revision paradigm, where `GPT4-Turbo` minimally revises and improves the answers of

llama-3-8B-instruct. Finally, we use a baseline dataset where all preferred answers are directly generated by GPT4-Turbo without any revisions.

Empirically, we indeed find that the quality of the comparisons is a major driver of alignment performance. Specifically, our revision dataset (created using CLAIR) produced the best results. This validates our insight that minimally contrastive preference pairs lead to better alignment, and that we can effectively create such preferences using a revision paradigm. Moreover, we discovered that "*apples-to-oranges*" preferences could actually deteriorate system performance. This again underscores how important it is to ensure the contrastiveness of the preference data, something which is not done in conventional preference learning work.

Additionally, we studied the impact of the system–data relation for alignment outcomes. No single alignment objective consistently outperformed the others across all datasets. This validates our second insight: the success of any preference learning objective is highly dependent on the relevance of the data to the system. The specific Anchored Preference Optimization objective which accounted for the exact semantics of the alignment situation always produced the best results. This illustrates that even relatively simple objectives can attain much better performance compared to conventional objectives, which are designed to operate across all scenarios.

3.7 Conclusions

Through a critical analysis of preference learning, we identified two intuitive yet previously underappreciated concepts. First, the comparisons underlying a preference need to be minimally contrastive in order to provide a clear and effective learning signal. Second, the relationship between the system and the comparisons should be accounted for during training.

We introduced new data and training methods that leverage these insights, leading to substantial improvements in alignment performance based on these straightforward principles. Additionally, we emphasized that preference learning is inherently more complex than imitation learning, and we explored the sources of this complexity.

3.7.1 Future Work

Minimally contrastive preferences can be created through a revise-and-improve paradigm. An intriguing direction for future work lies in exploring the natural trade-off between the minimalism of a revision and the extent to which it improves the answer. Both of these properties are desirable, but to some extent, they may be mutually exclusive. Given a fixed budget for creating preference data, should one prioritize minimal revisions that offer slight improvements or more invasive revisions that lead to significant enhancements? Investigating this balance could provide valuable insights into how to optimize alignment strategies even further.

4

Conclusions

Ensuring the interpretability of AI systems and aligning their behavior with human values and intentions are among the most significant challenges in the field today. As AI systems grow in complexity, autonomy, and influence over high-stakes decisions, these concerns become even more critical, and they directly impact the trust, safety, and effectiveness of AI in real-world applications.

In this thesis, we have made conceptually simple yet impactful contributions to the problems of AI interpretability and alignment. Counterfactual reasoning, which involves examining what-if scenarios, emerged as a key enabler for advancing both of these areas.

In the area of interpretability, we introduced methods that leverage counterfactual data to evaluate and enhance human-understandable interpretations of system behavior. Our formalisms and experimental framework provided a foundation for a more rigorous study of AI interpretability, grounded in empirical validation. Our results underscored the pressing need to make the AI interpretability field more rigorous through standardized evaluation methodologies.

In the area of alignment, we utilized counterfactual data in the form of minimally contrastive preference pairs to create a more effective learning signal, enabling more controllable and precise alignment outcomes. Furthermore, we uncovered the importance of tailoring the preference learning dependent to the relation between system and preference pairs, greatly improving the stability and performance of alignment objectives.

Together, these contributions mark a step forward in building AI systems that are both interpretable and aligned, significantly contributing to the effective and ethical use of AI.

Looking towards the future, it is clear Deep Learning has already made a

profound impact on the world and is bound to continue evolving and expanding its influence. During the time of this PhD, these systems became much more generalist, with the most famous examples being AI assistants powered by Large Language Models. While not perfect, these models have demonstrated an impressive ability to integrate vast amounts of knowledge, reason flexibly through unseen problems, and interact with various tools.

Over the next few years, we will undoubtedly see large-scale deployment of these systems and their integration into diverse workflows. We are already witnessing this transformation. However, the extent and speed of this transition remain uncertain. A major determining factor will be our ability to ensure the robustness and safety of these technologies. The types of alignment techniques discussed in this work will remain of vital importance—they have already proven highly effective. I strongly believe that for whichever alignment methods developed in the future, the data quality and the precise semantics of preference learning algorithms will play a crucial role, as made evident by our research on AI alignment.

The future of interpretability in AI is less certain. Deep questions about why an AI system behaved in a certain way often seem to be of little practical use to end-users. As a result, I see little chance that the field will develop highly effective and rigorous AI interpretability methods purely to satisfy end-users. Furthermore, the increasing reliability of AI systems—where they "just work"—diminishes the immediate demand for interpretability. However, there is a plausible future in which strong regulatory emphasis on auditability, safety, and privacy drives the development of more advanced interpretability methods. Regardless of the specific motivation for future investments in interpretability, I strongly believe that such techniques will be fundamentally causal, for the same reasons as those outlined in our work on benchmarking interpretability methods. Moreover, I anticipate that the most effective interpretability methods will involve secondary AI systems specifically trained to provide meaningful interpretations of a target system, as explored in our work on training AI interpreters.

5

Technical: Measuring Interpretations

In this chapter, we present a unifying perspective on AI interpretability methods, framing them as estimators of causal effects. We formalize this approach, reinterpreting existing interpretability techniques within this framework and introducing metrics and an experimental methodology to evaluate their faithfulness. We propose the first large-scale interventional dataset for natural language processing, enabling the practical evaluation of interpretability methods. Our findings revealed that no interpretability method outperformed a simple baseline, serving as a wake-up call for the field of AI interpretability research.

* * *

CEBaB: Estimating the Causal Effects of Real-World Concepts on NLP Model Behavior

Eldar David Abraham*, **Karel D’Oosterlink***, **Amir Feder***, **Yair Gat***, **Atticus Geiger***, **Christopher Potts***, **Roi Reichart***, **Zhengxuan Wu***

Advances in Neural Information Processing Systems (NeurIPS 2022)

*Equal contribution.

5.1 Abstract

The increasing size and complexity of modern ML systems has improved their predictive capabilities but made their behavior harder to explain. Many techniques for model explanation have been developed in response, but we lack clear criteria for assessing these techniques. In this paper, we cast model explanation as the causal inference problem of estimating causal effects of real-world concepts on the output behavior of ML models given actual input data. We introduce CEBaB, a new benchmark dataset for assessing concept-based explanation methods in Natural Language Processing (NLP). CEBaB consists of short restaurant reviews with human-generated counterfactual reviews in which an aspect (food, noise, ambiance, service) of the dining experience was modified. Original and counterfactual reviews are annotated with multiply-validated sentiment ratings at the aspect-level and review-level. The rich structure of CEBaB allows us to go beyond input features to study the effects of abstract, real-world concepts on model behavior. We use CEBaB to compare the quality of a range of concept-based explanation methods covering different assumptions and conceptions of the problem, and we seek to establish natural metrics for comparative assessments of these methods.

5.2 Introduction

Explaining model behavior has emerged as a central goal within ML. In NLP, models have grown in size and complexity, and while they have become increasingly successful, they have also become more opaque [Lipton, 2018, Pearl, 2019], raising concerns about trust [Guidotti et al., 2018, Jacovi and Goldberg, 2020], safety [Amodei et al., 2016, Otte, 2013], and fairness [Goodman and Flaxman, 2017, Hardt et al., 2016]. These concerns will persist if these models remain “black-boxes”.

Seeking to open the black-box, researchers have developed methods that try to explain model behavior [Bastings et al., 2021, Feder et al., 2021b, Gehrmann et al., 2020, Lundberg and Lee, 2017, Ribeiro et al., 2016]. However, there is no consensus about how to evaluate such methods to allow robust comparisons. This is not surprising, since such evaluations require very rich empirical data. Intuitively, we would like to (1) intervene on model inputs, to modify specific concepts without changing other correlated information, (2) observe the effects this has on model predictions, and, finally, (3) assess explanation methods for their ability to accurately predict these effects.

The absence of interventional data, or even an agreed-upon non-interventional benchmark, has created an environment in which explanation methods are often evaluated individually, and without comparison to alternatives. Attempts have been made to conduct comparative evaluations [Feder et al., 2021b, Goyal et al., 2020, Pruthi et al., 2022], but only with synthetic, simplified datasets. Furthermore, these attempts do not define a unified evaluation approach, nor do they seek to contribute benchmark datasets that support such evaluations.

		food	ambiance	service	noise	overall
Original text:	Excellent lobster and decor, but rude waiter.	+	+	-	unk	4
Edit Goal						
food:	- Terrible lobster, excellent decor, but rude waiter.	-	+	-	unk	2
food:	unk Excellent decor, but rude waiter.	unk	+	-	unk	3
ambiance:	- Excellent lobster, but lousy decor and rude waiter.	+	-	-	unk	3
ambiance:	unk Excellent lobster, but rude waiter.	+	unk	-	unk	3
service:	+ Excellent lobster and decor, and friendly waiter.	+	+	+	unk	5
service:	unk Excellent lobster and decor.	+	+	unk	unk	5
noise:	+ Excellent lobster, decor, and music, but rude waiter.	+	+	-	+	4
noise:	- Excellent lobster and decor, but rude waiter, and noisy.	+	+	-	-	3

Table 5.1: Toy examples illustrating the structure of CEBaB (actual corpus examples are longer and more complex; a sample is given in Appendix 5.10.2). Beginning from an OpenTable review, we give crowdworkers an actual restaurant review and they generate counterfactual restaurant reviews that would have been written if some aspect of the dining experience were changed and all else were held constant. Five different crowdworkers labeled each of the actual and counterfactual texts according to their aspect-level sentiment and overall sentiment. Aspect level sentiment labels are three way: ‘+’ (positive sentiment), ‘-’ (negative), and ‘unk’ (the aspect’s value is not expressed in the text). Overall sentiment labels are 1 (worst) to 5 (best). Edited aspect labels are shown in blue.

In this paper, we seek to overcome this obstacle by introducing **CEBaB** (Causal Estimation-Based Benchmark). Table 5.1 summarizes the structure of CEBaB with a toy example: beginning with a review text from the OpenTable website, we crowdsourced edits of the original text that are designed to meet a specific goal, such as changing the food rating in the original text to negative or unknown. All of the resulting edits were validated by five crowdworkers and each full text was evaluated by five crowdworkers for its overall sentiment. CEBaB is grounded in 2,299 original reviews, which were expanded via this editing procedure to a total of 15,089 texts, targeting four different aspect-level concepts (food, service, ambiance, noise) with three potential labels (positive, negative, and unknown, i.e., not expressed in the review), and each full text was labeled on a five-star scale.

We focus on using CEBaB to compare concept-based explanation methods. This allows us to go beyond the effect of individual tokens to study how more abstract concepts (in our case, aspect-level sentiment) contribute to model predictions (about the overall sentiment of the text). Our proposed metrics center around assessing concept-based explanation methods for their ability to accurately estimate *causal concept effects* [Goyal et al., 2020], allowing us to isolate the effect of individual concepts.

More specifically, we use CEBaB to measure the causal effects of particular variables in a causal graph, and we cast each explanation method as a causal estimator of these measurements. For example, suppose our causal graph of the data says that all four of our aspect-level categories will affect a reviewer’s overall

rating. To estimate the effect of positive food quality on the predicted overall rating from a classifier, we need to compare examples with high food quality to those with low quality, holding all other aspects constant. Such pairs of examples are normally not observed, but this is precisely what CEBaB provides. With CEBaB, we can directly compare the actual change in model predictions with the change that a concept-based explanation method predicts.

In our experiments, we evaluate five leading concept-based explanation methods: CONEXP [Goyal et al., 2020], TCAV [Kim et al., 2018], ConceptSHAP [Yeh et al., 2020], INLP [Ravfogel et al., 2020], CausaLM [Feder et al., 2021b], and S-Learner [Künzel et al., 2019]. These methods make a wide range of different assumptions about how much access we have to the model’s internal structure, and they also diverge in the degree to which they account for the causal nature of the concept effect estimation problem. Remarkably, CEBaB reveals that most methods cannot beat a simple baseline. Indeed, this negative result emphasizes the value in our primary contribution of providing the data and metrics that enables a direct comparison of explanation methods.

5.3 Previous Work

Benchmarks for Explanation Methods Benchmark datasets have propelled ML forward by creating shared metrics that predictive models can be evaluated on [Hu et al., 2020, Kiela et al., 2021, Wang et al., 2018, 2019]. Unfortunately, benchmarks that are suitable for assessing the quality of model explanations are still uncommon [Feder et al., 2021a, Hooker et al., 2019]. Previous work on comparing explanation methods has generally only correlated the performance of a given explainability method with others, without ground-truth comparisons [DeYoung et al., 2020, Hase and Bansal, 2020, Hooker et al., 2019, Samek et al., 2021].

Other works that do compare to some ground-truth either employ a non-causal evaluation scheme [Kim et al., 2018], use causal evaluation metrics which do not capture performance on individual examples [Tenney et al., 2020], evaluate on synthetic counterfactuals and rule-based augmentations [Feder et al., 2021b, Tenney et al., 2020], or are tailored for a specific explanation method and hard to generalize [Yeh et al., 2020]. To the best of our knowledge, CEBaB is the first large-scale naturalistic causal benchmark with interventional data for NLP.

Explanation Methods and Causality Probing is a relatively new technique for understanding what model internal representations encode. In probing, a small supervised [Conneau et al., 2018, Tenney et al., 2019] or unsupervised [Clark et al., 2019b, Manning et al., 2020, Saphra and Lopez, 2019] model is used to estimate whether specific concepts are encoded at specific places in a network. While probes have helped illuminate what models (especially pretrained ones) have learned from data, Geiger et al. [2021] show with simple analytic examples that probes cannot reliably provide causal explanations for model behavior.

Feature importance methods can also be seen as explanation methods [Molnar, 2020]. Many methods in this space are restricted to input features, but gradient-

based methods can often quantify the relative importance of hidden states as well [Binder et al., 2016, Shrikumar et al., 2017, Springenberg et al., 2014, Zeiler and Fergus, 2014]. The Integrated Gradients method of Sundararajan et al. [2017] has a natural causal interpretation stemming from its exploration of baseline (counterfactual) inputs [Geiger et al., 2021]. However, even where these methods can focus on internal states, it remains difficult to connect their analyses with real-world concepts that do not reduce to simple properties of inputs.

Intervention-based methods involve modifying inputs or internal representations and studying the effects that this has on model behavior [Lundberg and Lee, 2017, Ribeiro et al., 2016]. Recent methods perturb input or hidden representations to create counterfactual states that can then be used to estimate causal effects [Elazar et al., 2021, Finlayson et al., 2021, Soulos et al., 2020, Vig et al., 2020, Geiger et al., 2021]. However, these methods are prone to generating implausible inputs or network states unless the interventions are carefully controlled [Geiger et al., 2020].

Generating counterfactual texts automatically remains challenging and is still a work-in-progress [Calderon et al., 2022]. To overcome this problem, another class of approaches proposes to manipulate the representation of the text with respect to some concept, rather than the text itself [Elazar et al., 2021, Feder et al., 2021b, Ravfogel et al., 2020]. These methods fall into the category of concept-based explanations and we discuss two of them extensively in §5.4.

5.4 Estimating Concept Effects with CEBaB

We now define the core metrics that we use to evaluate different explanation methods. Figure 5.1 provides a high-level view of the causal process we are envisioning. The process begins with an exogenous variable U representing a state of the world. For CEBaB, we can imagine that the value of U is a state of affairs u of a person evaluating a restaurant in a particular way. u contributes to a review variable X , with the value x of X mediated by u and by mediating concepts C_1, \dots, C_k , which correspond to the four aspect-level categories in CEBaB (food, service, ambiance, and noise), each of which can have values $c \in \{\text{positive, negative, unknown}\}$. The review x is processed by a model that outputs a vector of scores over classes (sentiment labels in CEBaB).

Core Metric Our central goal is to use CEBaB to evaluate explanation methods themselves. CEBaB supports many approaches to such evaluation. In this paper, we adopt an approach based on individual-level rather than average effects. This makes very rich use of the counterfactual text and associated labels provided by CEBaB. The starting point for this metric is the Individual Causal Concept Effect:

Definition 1 (Individual Causal Concept Effect; ICaCE). *For a neural network \mathcal{N} and feature function ϕ , the individual causal concept effect of changing the value of concept C from c to c' for state of affairs u in an underlying data generation*

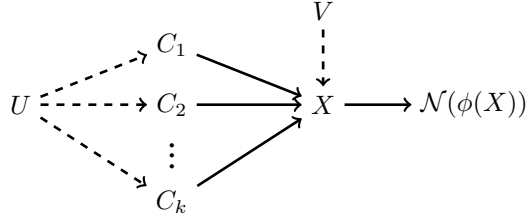


Figure 5.1: A causal graph describing a data generating process with an exogenous variables U and V representing the state of the world, mediating concepts C_1, C_2, \dots, C_k , and data X that is featurized with ϕ . $\phi(X)$ is input to a classifier \mathcal{N} , which outputs a vector of scores over m output classes.

process \mathcal{G} is

$$ICaCE_{\mathcal{N}_\phi}(\mathcal{G}, x_u^{C=c}, c') = \mathbb{E}_{x \sim \mathcal{G}} \left[\mathcal{N}(\phi(x)) \mid do \left(\begin{array}{l} C = c' \\ U = u \end{array} \right) \right] - \mathcal{N}(\phi(x_u^{C=c})). \quad (5.1)$$

ICaCE is a theoretical quantity. In practice, we use the Empirical Individual Causal Concept Effect.

Definition 2 (Empirical Individual Causal Concept Effect; \widehat{ICaCE}). *For a neural network \mathcal{N} and feature function ϕ , the empirical individual causal concept effect of changing the value of concept C from c to c' for state of affairs u is*

$$\widehat{ICaCE}_{\mathcal{N}_\phi}(x_u^{C=c}, x_u^{C=c'}) = \mathcal{N}(\phi(x_u^{C=c'})) - \mathcal{N}(\phi(x_u^{C=c})), \quad (5.2)$$

where $(x_u^{C=c}, x_u^{C=c'})$ is a tuple of inputs originating from u with the concept C set to the values c and c' , respectively.

The $\widehat{ICaCE}_{\mathcal{N}_\phi}$ for a pair of examples $(x_u^{C=c}, x_u^{C=c'})$ is simply the difference between the output score vectors for the two cases. With CEBaB, we can easily calculate these values because we have clusters of examples that are tied to the same reviewing situation u and express different concept values.

For assessing an explanation method \mathcal{E} , we compare ICaCE values with those returned by \mathcal{E} . Our core metric is the ICaCE-Error:

Definition 3 (ICaCE-Error). *For a neural network \mathcal{N} , feature function ϕ and distance metric Dist , the ICaCE-Error of an explanation method \mathcal{E} for changing the value of concept C from c to c' is:*

$$ICaCE\text{-Error}_{\mathcal{N}_\phi}^{\mathcal{D}}(\mathcal{E}) = \frac{1}{|\mathcal{D}|} \sum_{(x_u^{C=c}, x_u^{C=c'}) \in \mathcal{D}} \text{Dist}(\widehat{ICaCE}_{\mathcal{N}_\phi}(x_u^{C=c}, x_u^{C=c'}), \mathcal{E}_{\mathcal{N}_\phi}(x_u^{C=c}, c')) \quad (5.3)$$

We present results for three choices of Dist which vary in their ability to model the direction and magnitude of effects. These choices give subtly different but largely converging results, as detailed in Section 5.7 and reported more fully in Appendix 5.10.4.

Aggregating Individual Causal Concept Effect It is often useful to also have a direct estimate of a model’s ability to capture concept-level causal effects. For this, we employ an aggregating version of $\widehat{\text{ICaCE}}$, the Empirical Causal Concept Effect:

Definition 4 (Empirical Causal Concept Effect; $\widehat{\text{CaCE}}$). *For a neural network \mathcal{N} and feature function ϕ , the empirical causal concept effect of changing the value of concept C from c to c' in dataset \mathcal{D} is*

$$\widehat{\text{CaCE}}_{\mathcal{N},\phi}^{\mathcal{D}}(C, c, c') = \frac{1}{|\mathcal{D}_{C^{c \rightarrow c'}}^{\mathcal{D}}|} \sum_{(x_u^{C=c}, x_u^{C=c'}) \in \mathcal{D}_{C^{c \rightarrow c'}}^{\mathcal{D}}} \widehat{\text{ICaCE}}_{\mathcal{N},\phi}(x_u^{C=c}, x_u^{C=c'}). \quad (5.4)$$

This is an empirical estimator of the Causal Concept Effect (CaCE) of Goyal et al. [2020]. It estimates, in general, how the classifier predictions change for a given concept and intervention direction.

Estimating Real-World Causal Effect of Aspect Sentiment on Overall Sentiment We can also estimate ground truth causal effects in CEBaB by simply using its labels directly. There are again a variety of ways that this could be done. We opt for the one that makes the richest use of the structures afforded by CEBaB. For perspicuity, in parallel to the neural network-based $\widehat{\text{ICaCE}}$ (Definition 2), we define the Empirical Individual Treatment Effect for our dataset:

Definition 5 (Empirical Individual Treatment Effects in CEBaB; $\widehat{\text{ITE}}$). *The empirical individual treatment effect of changing the value of concept C from c to c' in CEBaB is*

$$\widehat{\text{ITE}}^{\text{CEBaB}}(x_u^{C=c}, x_u^{C=c'}) = f(x_u^{C=c'}) - f(x_u^{C=c}) \quad (5.5)$$

where f is a simple look-up procedure that retrieves the overall sentiment labels for CEBaB examples.

We aggregate over these values by taking their average, in parallel to what we do for network predictions (Definition 4). This yields the Empirical Average Treatment Effect ($\widehat{\text{ATE}}$) for CEBaB.

Alternative Metrics In Appendix 5.10.1 in our supplementary materials, we consider alternative formulations of the core metrics with *causal concept effects* and *absolute causal concept effects*, relating them to the different questions they engage with. We opt for the individual causal concept effect in our central metric (Definition 6.4), taking the central question to be what caused an ML model to produce an output for an *actual* input created from a real-world process.

Explanation method	Explainer Method	Access to Explained Model	Concept Labels Needed (test time)	Counterfactual Representation	Confounder Control
Approx	None	None	All concepts and their labels*	✗	✗
CONEXP [Goyal et al., 2020]	None	None	c and c'	✗	✗
S-Learner Künzel et al. [2019]	Linear	None	All concepts and their labels*	✗	✓
TCAV [Kim et al., 2018]	Linear	Weights	None	✗	✗
ConceptSHAP [Yeh et al., 2020]	Linear	Weights	None	✗	✗
INLP [Ravfogel et al., 2020]	Linear	Weights	None	✓	✗
CausaLM [Feder et al., 2021b]	Explained Model	Training Regime	None	✓	✓

Table 5.2: The evaluated explanation methods and their attributes. **Explainer Method** denotes the complexity of the models used by each explanation method. **Access to Explained Model** denotes the degree of access an explainer method needs to the explained model. **Concept Labels Needed** indicates whether a method estimating the effect for an input $x_u^{C=c}$ needs the actual input label c and/or the intervened value c' at test time. Models with a **Counterfactual Representation** approximate $\phi(x_u^{C=c'})$ to estimate the effect. Finally, only CausaLM and S-Learner have **Confounder Control** to minimize the impact of confounding concepts. *We predict these labels with a classifier.

5.5 Evaluated Explanation Methods

We compare several model explanation methods that share three main characteristics. First, they are all suitable for NLP models and have been used in the literature for generating model explanations in the form of estimated effects on model predictions. Second, they all provide concept-level explanations, for a pre-defined list of human-interpretable concepts (e.g., how sensitive a restaurant review rating classifier is to language related to food quality). This approach is also forward-looking, allowing more researchers to construct new hypotheses (i.e., concepts we have not collected labels for) and estimate their effect on the predictor. Third, all of the tested methods are model-agnostic, meaning that they separate the explanation from the model. At the same time, these methods differ in five important ways, as summarized Table 5.2.

We now turn to reviewing the explanation methods that we later compare on CEBaB (§5.7). In our mathematical formulas, we employ a unified notation for all methods, to make the definitions more accessible and easier to integrate into our experimental set-up. Assume we have a classifier \mathcal{N} (which outputs a probability vector) and feature function ϕ , and we want to compute the effect on $\mathcal{N}_\phi(x_u^{C=c})$ of changing the value of concept C from c to c' using an unseen test set (\mathcal{D}, Y) .

Approximate Counterfactuals The gold labels of CEBaB are the difference between the logits for some original review $x_u^{C=c}$ and ground-truth counterfactual $x_u^{C=c'}$. As a baseline, we sample an original review $x_{u'}^{C=c'}$ with the same aspect-labels as the $x_u^{C=c'}$ and use it as an approximate counterfactual:

$$\text{Approx}_{\mathcal{N}_\phi}(C, c, c'; x) = \mathcal{N}(\phi(x_{u'}^{C=c'})) - \mathcal{N}(\phi(x_u^{C=c})) \quad (5.6)$$

We do this sampling using predicted aspect labels from the aspect-level sentiment analysis models described in Appendix 5.10.3.

Conditional Expectation (CONEXP) Goyal et al. [2020] propose a baseline where the effect of a concept C is the average difference in predictions on examples with different values of C .

$$\text{CONEXP}_{\mathcal{N}_\phi}^{\mathcal{D}}(C, c, c') = \frac{1}{|\mathcal{D}^{C=c'}|} \sum_{x \in \mathcal{D}^{C=c'}} \mathcal{N}(\phi(x)) - \frac{1}{|\mathcal{D}^{C=c}|} \sum_{x \in \mathcal{D}^{C=c}} \mathcal{N}(\phi(x)) \quad (5.7)$$

where $\mathcal{D}^{C=c}$ and $\mathcal{D}^{C=c'}$ are subsets of \mathcal{D} where C takes values c and c' , respectively. To predict an effect, this method only relies on C , c , and c' , resulting in an estimate that does not depend on the specific input text itself.

Conditional Expectation Learner (S-Learner) We adapt *S-Learner*, a popular method for estimating the Conditional Average Treatment Effect (CATE) [Künzel et al., 2019]. To estimate causal concept effects, our *S-Learner* trains a logistic regression model \mathcal{E} to predict $\mathcal{N}(\phi(x))$ using the values of all the labeled concepts of example x , denoted by x' .¹ Then, during inference, we compute an individual effect for example pair $(x_u^{C=c}, x_u^{C=c'})$ by comparing the output of the model \mathcal{E}_x on this pair:

$$S\text{-Learner}(C, c, c'; x) = \mathcal{E}(x_u^{C=c'}) - \mathcal{E}(x_u^{C=c}) \quad (5.8)$$

At inference time, S-Learner assumes access to all aspect-level labels x' , which might not always be available. To alleviate this issue, we instead *predict* the aspect-level labels x' from the original text x using models described in Appendix 5.10.3.

TCAV Kim et al. [2018] use *Concept Activation Vectors* (CAVs), which are semantically meaningful directions in the embedding space of ϕ . Our adapted version of Testing with CAVs (TCAV) outputs a vector measuring the sensitivity of each output class k to changes towards the direction of a concept v_C at the point of the embedded input. It is computed as:

$$\text{TCAV}_{\mathcal{N}_\phi}(C; x) = (\nabla \mathcal{N}_k(\phi(x)) \cdot v_C)_{k=1}^K \quad (5.9)$$

where K is the number of classes and v_C is a linear separator learned to separate concept C in the embedding space of ϕ .

ConceptSHAP Yeh et al. [2020] propose this expansion to SHAP [Lundberg and Lee, 2017], to generate concept-based explanation based on Shapley values [Shapley, 1953]. Given a *complete* (i.e., such that the accuracy it achieves on a test set is higher than some threshold β) set of m concepts $\{C_1, \dots, C_m\}$, ConceptSHAP calculates the contribution of each concept to the final prediction. Our adapted version outputs a vector for each $C \in \{C_1, \dots, C_m\}$ and x . We justify this modification and provide implementation details in Appendix 5.10.9.

CausaLM Feder et al. [2021b] estimate the causal effect of a binary concept C on the model’s predictions by adding auxiliary adversarial tasks to the language representation model in order to learn a counterfactual representation $\phi_C^{\text{CF}}(x)$, while

¹This training approach, where an explainer model is fit to predict the output of the original model, shares the intuition of LIME, the widely used explanation method Ribeiro et al. [2016], but for concept-level effects.

keeping essential information about potential confounders (control concepts). Their method outputs the text representation-based individual treatment effect (TReITE), which is computed as:

$$\text{TReITE}_{\mathcal{N}_\phi}(C; x) = \mathcal{N}'(\phi_C^{\text{CF}}(x)) - \mathcal{N}(\phi(x)) \quad (5.10)$$

where ϕ_C^{CF} denotes the learned counterfactual representation, where the information about concept C is not present, and \mathcal{N}' is a classifier trained on this counterfactual representation. A key feature of CausaLM is its ability to control for confounding concepts (if modeled).² An inherent drawback of this technique is that it can only estimate interventions well for $c' = \text{Unknown}$, since the counterfactual representation is only trained to *remove* a concept C .

Iterative Nullspace Projection (INLP) Ravfogel et al. [2020] remove a concept from a representation vector by repeatedly training linear classifiers that aim to predict that attribute from the representations and projecting the learned representations on their null-space. Similar to CausaLM, INLP also estimates the TReATE (Equation 5.10) and can only estimate interventions for $c' = \text{Unknown}$.

5.6 The CEBaB Dataset

Table 5.1 provides an intuitive overview of the structure of CEBaB. In the *editing* phase of dataset creation, crowdworkers modified an existing OpenTable review in an effort to achieve a specific aspect-level goal while holding all other properties of the original text constant. Our aspect-level categories are food, ambiance, service, and noise. In the *validation* phase, crowdworkers labeled each example relative to each aspect as ‘Positive’, ‘Negative’, or ‘Can’t tell’ (Unknown). Having five labels per example allows us to infer a majority label or reason in terms of the full label distributions. In the *rating* phase, each full text was labeled using a common five-star scale, again by five crowdworkers.

We began with 2,299 original reviews from OpenTable (related to 1,084 restaurants) and expanded them, via the above editing procedure, into a total of 15,089 texts. The distribution of normalized edit distances has peaks around 0.28 and 0.77, showing that workers made non-trivial changes to the originals, and even often had to make substantial changes to achieve the editing goal. (See Appendix 5.10.2 for the full distribution.)

Table 5.3 summarizes the resulting label distributions, where an example has label y if at least 3 of the 5 labelers chose y , otherwise it is in the ‘no majority’ category. 99% of aspect-level edits have a majority label that corresponds to the editing goal, and 88% of the texts have a review-level majority label on the five-star scale. Overall, these percentages show that workers were extremely successful in achieving their editing goals and that edits have systematic effects on overall sentiment.

²As in Feder et al. [2021b], we control for the most correlated potential confounder.

	Positive	Negative	Unknown	no maj.	Total
food	5726 (41%)	5526 (38%)	2605 (15%)	208 (31%)	14065
service	4045 (29%)	4098 (28%)	3877 (22%)	178 (27%)	12198
ambiance	2928 (21%)	2597 (18%)	5121 (29%)	203 (30%)	10849
noise	1365 (10%)	2215 (15%)	5883 (34%)	78 (12%)	9541

(a) Aspect-level labels.

Rating	Count	Percentage
1 star	1870	(12%)
2 star	3056	(20%)
3 star	3517	(23%)
4 star	2035	(13%)
5 star	2732	(18%)
no maj.	1879	(12%)

(b) Review-level ratings.

	{Neg, Pos}	{Neg, Unk}	{Pos, Unk}
food	898	1316	1291
service	851	857	938
ambiance	947	585	472
noise	1145	208	260

(c) Edit pair distribution. Edit pairs are examples that come from the same original source text and differ only in their rating for a particular aspect.

	Neg to Pos	Neg to Unk	Pos to Unk
food	1.84	1.37	-1.02
service	0.98	0.91	-0.53
ambiance	0.93	0.91	-0.50
noise	0.72	0.48	-0.47

(d) Empirical $\widehat{\text{ATE}}$ for the five-way sentiment labels in CEBaB. The reverse of a given concept change is the negative of the value given – e.g., the $\widehat{\text{ATE}}$ for ‘Pos to Neg’ for food is -1.84 .

Table 5.3: Dataset statistics of CEBaB combining train/dev/test splits.

The central goal of CEBaB is to create *edit pairs*: pairs of examples that come from the same original text and differ only in their labels for a particular aspect. For example, in Table 5.1, the first two ‘food edit’ cases form an edit pair, since they come from the same original text and differ only in their food label. Original texts can also contribute to edit pairs; the original text in Table 5.1 forms an edit pair with each of the texts it is related to by edits. Table 5.3c summarizes the distribution of edit pairs, and Table 5.3d reports the ground-truth $\widehat{\text{ATE}}$ values (§5.4).

We release the dataset with fixed train/dev/test splits. In creating these splits, we enforce two high-level constraints. The first is our ‘grouped’ requirement: for each original review t , all texts that are related to t via editing occur in the same split as t . This ensures that models are not evaluated on examples that are related by editing to those they have seen in training. Second, if any text t in a group received a ‘no majority’ label, then the entire group containing t is put in the train set. This ensures that there is no ambiguity about how to evaluate models on dev and test examples.

Once these high-level conditions were imposed, the examples were sampled randomly to create the splits. This allows that individual workers can contribute edited texts across splits. This minor compromise was necessary to ensure that we could have large dev and test splits. Appendix 5.10.3 in our supplementary materials shows that worker identity has negligible predictive power.

There are two versions of the train set: *inclusive* and *exclusive*. The inclusive train set contains all original and edited non-dev/test texts (11,728 texts). The exclusive version samples exactly one train text from each set of texts that are related by editing (1,755 examples). The rationale is that models trained with an original review as well as its edited counterparts may explicitly learn causal effects trivially by aggregating learning signals across inputs. Our exclusive train split prevents this, which helps facilitate fair comparisons between explanation methods and better resembles a real-world setting.

Our dataset is released publicly in JSON format and is available in the Hugging Face `datasets` library. It includes restaurant metadata, full rating distributions, and anonymized worker ids. Appendix 5.10.2 in our supplementary materials provides additional details on the dataset construction, including the prompts used by the crowdworkers, the number of workers per task, worker compensation, and a sample of examples with ratings to help convey the nature of workers’ edits and the overall quality of the resulting texts and labels. In addition, Appendix 5.10.3 reports on a wide range of classifier experiments at the aspect-level and text-level that show that models perform well on CEBaB classification tasks, which bolsters the claim that CEBaB is a reliable tool for assessing explanation methods.

5.7 Experiments and Results

For each experiment, we fine-tune a pretrained language model to predict the overall sentiment of all restaurant reviews from our *exclusive* OpenTable train set. Since the goal of our work is not to achieve state-of-the-art performance, but rather

	Negative to Positive	Negative to unknown	Positive to unknown
food	1.90 (\pm 0.03)	1.00 (\pm 0.02)	-0.82 (\pm 0.01)
service	1.42 (\pm 0.04)	0.89 (\pm 0.04)	-0.45 (\pm 0.01)
ambiance	1.27 (\pm 0.01)	0.79 (\pm 0.01)	-0.50 (\pm 0.03)
noise	0.75 (\pm 0.02)	0.44 (\pm 0.00)	-0.23 (\pm 0.02)

Table 5.4: $\widehat{\text{CaCE}}$ (Definition 5.4) for `bert-base-uncased` fine-tuned as a 5-way sentiment classifier. Rows are concepts, columns are real-world concept interventions, and each entry indicates the average change in classifier output when the concept is intervened on with the given direction.³ Results are averaged over 5 distinct seeds with standard deviations. The $\widehat{\text{CaCE}}$ value of changing concept C from c to c' is the negative $\widehat{\text{CaCE}}$ value of changing concept C from c' to c .

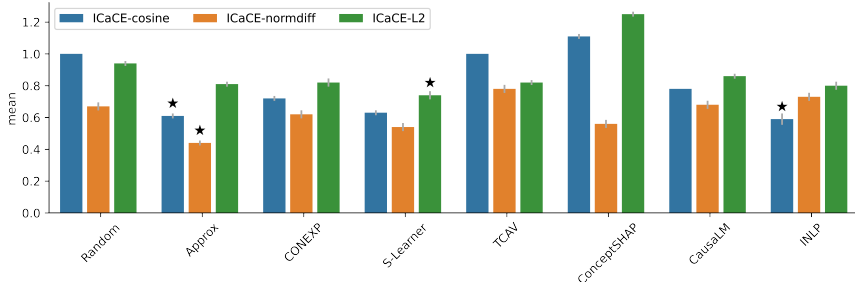


Figure 5.2: ICaCE-Error (Definition 3) for `bert-base-uncased` fine-tuned for five-way sentiment, averaged per aspect. We report values for cosine, L2, and normdiff. **Lower is better**. Stars mark the best result(s) per metric. Results averaged over 5 distinct seeds. †RandomExplainer takes the difference between two random probability vectors as the predicted effect.

to compare explanation methods and demonstrate the usage of CEBaB, we test the ability of methods to explain commonly used models, trained with standard experimental configurations.

In the main text, we report results for `bert-base-uncased` fine-tuned as a five-way classifier. Appendix 5.10.4 includes results for GPT-2, RoBERTa, and an LSTM, fine-tuned on binary, 3-way and 5-way versions of the sentiment task. All results, including the ground-truth effect that depends on the specific instance of a model, are averaged across 5 seeds.

To evaluate the intrinsic capacity of a model to capture causal effects, we report the $\widehat{\text{CaCE}}$ values, as in Definition 5.4. The results for `bert-base-uncased` are given in Table 5.4. They are intuitive and well-aligned with the $\widehat{\text{ATE}}$ estimates in Table 5.3d, indicating that the model has captured the real-world effects.

³Definition 5.4 defines the CaCE values as vectors. In this table, we collapse the CaCE values to

Our primary assessment of the evaluation methods is given in Figure 5.2, again focusing on a five-way `bert-base-uncased` model as representative of our results. We provide values based on *cosine*, *L2*, and *normdiff* as the value of Dist in Definition 6.4. The *cosine*-distance metric measures if the estimated and observed effect have the same direction but does not take the magnitudes of the effects into account. The *L2*-distance measures the Euclidian norm of the difference of the observed and estimated effect. Both the direction and magnitude of the effects influence this metric. To only compare the magnitudes, we use the *normdiff*-distance, which computes the absolute difference between the Euclidean norms of the observed and estimated effects, thus completely ignoring the directions of both effects.

Remarkably, our approximate counterfactual baseline proves to be the best method at capturing both the direction and magnitude of the effects. The fact that a simple baseline method beats almost all other methods indicates that we need better explanation methods if we are going to capture even relatively simple causal effects like those given by CEBaB.

Recall from Table 5.2 that the compared methods require different levels of access to concept labels at inference time. Approximate counterfactuals and S-Learner have access to both the direction of the intervention and the predicted test-time aspect labels, enabling them to outperform CONEXP, which has access to only the direction of the intervention, and TCAV, ConceptSHAP, and CausaLM, which have access to neither the intervention direction nor test-time aspect labels.

The INLP method ties with the best method for the *cosine* metric, despite having access to neither intervention directions nor test-time aspect labels. Perhaps this method could be extended to make use of this additional information and decisively improve upon our approximate counterfactual baseline.

While CausaLM and INLP both estimate the effect of removing a concept from an input, INLP uses linear probes to guide interventions on the original model, while CausaLM trains an entirely new model with an auxiliary adversarial objective. The direct use of the original model is something INLP shares with the approximate counterfactual baseline; it seems that a tight connection to the original model may underlie success on CEBaB.

5.8 Conclusion

Our main contributions in this paper are twofold. First, we introduced CEBaB, the first benchmark dataset to support comparing different explanation methods against a single ground-truth with human-created counterfactual texts and multiply-validated concept labels for aspect-level and overall sentiment. Using this resource, one can isolate the true causal concept effect of aspect-level sentiment on any trained overall sentiment classifier. CEBaB provides a level playing field on which we can compare a variety of explanation methods that differ in their assumptions

scalars by having \mathcal{N} output the most probable predicted class, instead of the class distribution.

about their access to the model, their computational demands, their access to ground-truth concept labels at inference time, and their overall conception of the explanation problem. Furthermore, the evaluated methods make absolutely no use of CEBaB’s counterfactual train set. In turn, we hope that CEBaB will facilitate the development of explanation methods that can take advantage of the very rich counterfactual structure CEBaB provides across all its splits.

Second, we have provided an in-depth experimental analysis of how well multiple model explanation methods are able to capture the true concept effect. A naive baseline that approximates counterfactuals through sampling achieves the best performance, with INLP and S-Learner being the only other methods that achieves state-of-the art on any metric. While CEBaB is only grounded in one task, sentiment analysis alone is enough to produce starkly negative results that should serve as a call to action for NLP researchers aiming to explain their models.

5.9 Acknowledgement

This research is supported in part by a grant from Meta AI. Karel D’Oosterlinck was supported through a doctoral fellowship from the Special Research Fund (BOF) of Ghent University. We thank our crowdworkers for their invaluable contributions to CEBaB.

5.10 Appendix

5.10.1 Causal Concept Effects and Metrics for Explanation Methods

Data do not materialize out of thin air. Rather, data are generated from real-world processes with complex causal structures we do not observe directly. Causal inference is the task of estimating theoretical causal effect quantities.

When estimating causal effects, researchers commonly measure the *average treatment effect*, which is the difference in mean outcomes between the treatment and control groups [Rubin, 1974]. Formally, we define the average treatment effect of binary treatment T on an outcome Y under a data generation process \mathcal{G} that represents the unknown details of the real-world.

Definition 6 (Average Treatment Effect; ATE [Rubin, 1974, Pearl, 1995]).

$$ATE_T(Y, \mathcal{G}) = \mathbb{E}_{\mathcal{G}}[Y \mid do(T = 1)] - \mathbb{E}_{\mathcal{G}}[Y \mid do(T = 0)]. \quad (5.11)$$

The ATE is a theoretical quantity we cannot compute in practice, since we do not have access to \mathcal{G} nor can we observe both interventions for the same subject.

However, we are concerned with estimating the causal effect of variables representing *non-binary concepts* in real-world systems, on data in an appropriate format for processing by a modern AI model that predicts *vector encoding probability distributions* over outputs.

Let \mathcal{N} be a neural network outputting a probability vector, where its k -th entry represents the probability to predict the k -th class, and let ϕ be a feature representation (e.g., BERT embedding). In the context of model explanations, we will define the tools needed to answer three questions:

1. Given a real-world circumstance u that led to input data $x_u^{C=c}$, what is the expected effect of a concept C changing from value c to value c' on the model output of \mathcal{N}_ϕ provided input data $x_u^{C=c}$?
2. What is the expected effect of a concept C changing from value c to value c' on the output of the model \mathcal{N}_ϕ provided input data X across real-world circumstances U ?
3. What is the magnitude of the expected effect of a changing the concept C on the output of the model \mathcal{N}_ϕ provided input data X across real-world settings U ?

For example, in the context of CEBaB, we might ask

1. Given a real-world dining experience u with good food quality ($C_{\text{food}} = +$) that led to a restaurant review $x_u^{C_{\text{food}}=+}$, what is the effect of changing the food quality C_{food} from $C_{\text{food}} = +$ to $C_{\text{food}} = -$ on the output of an overall-sentiment text classifier \mathcal{N}_ϕ provided a review of the dining experience?
2. What is the expected effect of changing the food quality C_{food} from positive $+$ to negative $-$ on the output of the model \mathcal{N}_ϕ across real-world dining experiences that lead to restaurant reviews?
3. What is the magnitude of the expected effect of a changing food quality C_{food} on the output of the model \mathcal{N}_ϕ across real-world dining experiences that lead to restaurant reviews?

Each of the above questions requires the estimation of a different theoretical quantity. In respect to the order of the questions, these quantities are the *individual causal concept effect*, the *causal concept effect*, and the *absolute causal concept effect*.

We believe the most practical question in explainable AI is: why does this model have this output behavior for an *actual* input. For this reason, our focus in the main text is *individual causal concept effects*. We define our central metric that captures the performance of an explainer on CEBaB as the average error on individual causal effect predictions (Definition 3).

We do not evaluate the ability of explainers to evaluate the causal concept effect or the absolute causal concept effect.

5.10.1.1 Theoretical Quantities

Definition 7 (Causal Concept Effects; [Goyal et al., 2020]). *For an exogenous setting u that led to concept C taking on value c and the creation of input data $x_u^{C=c}$, the individual causal concept effect of a concept C changing from value c to*

c' in a data generation process \mathcal{G} on a neural network \mathcal{N} with feature representation ϕ is

$$ICaCE_{\mathcal{N}_\phi}(\mathcal{G}, x_u^{C=c}, c') = \mathbb{E}_{x \sim \mathcal{G}} \left[\mathcal{N}(\phi(x)) \mid do \left(\begin{array}{l} C = c' \\ U = u \end{array} \right) \right] - \mathcal{N}(\phi(x_u^{C=c})) \quad (5.12)$$

The causal concept effect is the effect in general, meaning there is no input data generated from a fixed exogenous real-world setting:

$$CaCE_{\mathcal{N}_\phi}(\mathcal{G}, C, c, c') = \mathbb{E}_{x \sim \mathcal{G}} [\mathcal{N}(\phi(x)) \mid do(C = c')] - \mathbb{E}_{x \sim \mathcal{G}} [\mathcal{N}(\phi(x)) \mid do(C = c)] \quad (5.13)$$

The absolute causal concept effect estimate of the magnitude of the effect a concept has on a classifier output, regardless the concept values. We aggregate over all possible intervention values in the following way

$$ACaCE_{\mathcal{N}_\phi}(\mathcal{G}, C) = \frac{1}{|\{\{c, c'\} \subseteq C\}|} \sum_{\{c, c'\} \subseteq C} |CaCE_{\mathcal{N}_\phi}(\mathcal{G}, C, c, c')|, \quad (5.14)$$

where C is the set of all possible values for concept in addition to denoting the concept itself.⁴

5.10.1.2 Empirical Estimates

Similar to the ATE, causal concept effects are theoretical quantities we can only estimate in reality. To perform such estimates, we need a dataset consisting of pairs $(x_u^c, x_u^{c'}) \in \mathcal{D}$ that are drawn from a data generation process \mathcal{G} . A major contribution of this work is crowdsourcing such a dataset, CEBaB. These pairs allow us to compute empirical estimations of (individual) causal concept effects.

Definition 8 (Empirical Causal Concept Effects). *For an exogenous setting u , the empirical individual causal concept effect of a concept C changed from value c to c' , for \mathcal{D} sampled from \mathcal{G} , on a neural network \mathcal{N} trained on a feature representation ϕ is*

$$\widehat{ICaCE}_{\mathcal{N}_\phi}(x_u^{C=c'}, x_u^{C=c}) = \mathcal{N}(\phi(x_u^{C=c'})) - \mathcal{N}(\phi(x_u^{C=c})) \quad (5.15)$$

Given a full dataset \mathcal{D} of such pairs, we can estimate the causal concept effect

$$\widehat{CaCE}_{\mathcal{N}_\phi}(\mathcal{D}, C, c, c') = \frac{1}{|\mathcal{D}_C^{c \rightarrow c'}|} \sum_{(x_u^c, x_u^{c'}) \in \mathcal{D}} \widehat{ICaCE}_{\mathcal{N}_\phi}(x_u^{C=c}, x_u^{C=c'}) \quad (5.16)$$

And also the absolute causal concept effect

$$\widehat{ACaCE}_{\mathcal{N}_\phi}(\mathcal{D}) = \frac{1}{|\{\{c, c'\} \subseteq C\}|} \sum_{(c, c') \in C} |\widehat{CaCE}_{\mathcal{N}_\phi}(\mathcal{D}, C, c, c')| \quad (5.17)$$

⁴We take the absolute value since $CaCE_{\mathcal{N}_\phi}(\mathcal{G}, C, c, c') = -CaCE_{\mathcal{N}_\phi}(\mathcal{G}, C, c', c)$, and these cancel each other in the summation.

Notice that the only difference between causal concept effects (Definition 7) and empirical causal concept effects (Definition 8) is that we change the expectation taken over \mathcal{G} to be the average over a dataset $\mathcal{D} \sim \mathcal{G}$.

5.10.1.3 Explainer Errors

Given a dataset \mathcal{D} and an explainer $\mathcal{E}_{\mathcal{N}_\phi}(x_u^c, c')$ that predicts individual causal concept effects $ICACE_{\mathcal{N}_\phi}(x_u^c, c')$, we define metrics capturing the ability of \mathcal{E} to estimate causal effects by simply computing the averaged distance between our explainer and the empirical causal effect

Definition 9 (Explainer Distances). *The average distance between the explainer and the empirical individual causal concept effects.*

$$ICaCE\text{-Error}_{\mathcal{N}_\phi}^{\mathcal{D}}(\mathcal{E}, C, c, c') = \frac{1}{|\mathcal{D}_C^{c \rightarrow c'}|} \sum_{(x_u^{C=c}, x_u^{C=c'}) \in \mathcal{D}_C^{c \rightarrow c'}} \text{Dist}(\widehat{ICaCE}_{\mathcal{N}_\phi}(x_u^{C=c}, x_u^{C=c'}), \mathcal{E}_{\mathcal{N}_\phi}(x_u^{C=c}, x_u^{C=c'})) \quad (5.18)$$

The distance between the average of explainer outputs and the empirical causal concept effect

$$CaCE\text{-Error}_{\mathcal{N}_\phi}^{\mathcal{D}}(\mathcal{E}, C, c, c') = \|\widehat{CaCE}_{\mathcal{N}_\phi}(\mathcal{D}, C, c, c'), \frac{1}{|\mathcal{D}_C^{c \rightarrow c'}|} \sum_{x_u^c, x_u^{c'} \in \mathcal{D}_C^{c \rightarrow c'}} \mathcal{E}_{\mathcal{N}_\phi}(x_u^c, c')\| \quad (5.19)$$

The distance between the average magnitude of explainer outputs and the empirical absolute causal effect

$$ACaCE\text{-Error}_{\mathcal{N}_\phi}^{\mathcal{D}}(\mathcal{E}, C) = \|\widehat{ACaCE}_{\mathcal{N}_\phi}(\mathcal{D}, C), \frac{1}{|\{\{c, c'\} \subseteq C\}|} \sum_{(c, c') \in C} \frac{1}{|\mathcal{D}_C^{c \rightarrow c'}|} \sum_{x_u^c, x_u^{c'} \in \mathcal{D}_C^{c \rightarrow c'}} |\mathcal{E}_{\mathcal{N}_\phi}(x_u^c, c')|\| \quad (5.20)$$

where $|\cdot|$ is some distance metric and \mathcal{D}_C is the subset of data where C is the concept changed and $\mathcal{D}_C^{c \rightarrow c'}$ is the subset of data where C is the concept changed from value c to value c' .

In the main text, we use the ICaCE-Error as our primary evaluation metric.

5.10.2 CEBaB

Our supplementary materials contain a full Datasheet for CEBaB as a separate markdown document.

Instructions

You will be shown a short review of a restaurant. Your task is to edit the review to change a specific aspect of the review while keeping everything else the same as much as possible and trying to produce a fluent, natural text. Here are some examples:

- Example:
Goal: Change the service evaluation to negative.
Original: The food and ambiance were great, and the service was superb.
Edited version: The food and ambiance were great, **but** the service was **very slow**.
- Example:
Goal: Change the cuisine to Italian.
Original: I had a lamb pita that must have been made a week before my meal.
Edited version: I had a **pepperoni pizza** that must have been made a week before my meal.

i **Previewing Answers Submitted by Workers**
This message is only visible to you and will not be shown to Workers.
You can test completing the task below and click "Submit" in order to preview the data and format of the submitted results.
×

Goal: Change the \$(type) to \$(goal)

Sentence: \${description}

Make your edits here:

\${description}

If you have any questions/suggestions about this task, feel free to leave a comment here. We appreciate your input on this task.

Comment...

Submit

Figure 5.3: Edit phase annotation interface where the task was to convey ‘Positive’ or ‘Negative’ for the target aspect.

Instructions

You will be shown a short review of a restaurant. Your task is to edit the review to **remove** specific information:

- Example:
Goal: Remove the service evaluation
Original: Everything about the food, service, and ambiance was outstanding.
Edited version: Everything about the food and ambiance was outstanding.
- Example:
Goal: Remove the food evaluation
Original: I had a lamb pita that must have been made a week before my meal, but at least the service was prompt.
Edited version: I had a lamb pita, and the service was prompt.

i **Previewing Answers Submitted by Workers**
This message is only visible to you and will not be shown to Workers.
You can test completing the task below and click "Submit" in order to preview the data and format of the submitted results.
×

Goal: Remove the \$(edit_type)

Sentence: \${original_description}

Make your removal edits here:

\${original_description}

If you have any questions/suggestions about this task, feel free to leave a comment here. We appreciate your input on this task.

Comment...

Submit

Figure 5.4: Edit phase annotation interface where the task was to hide the sentiment of the target aspect.

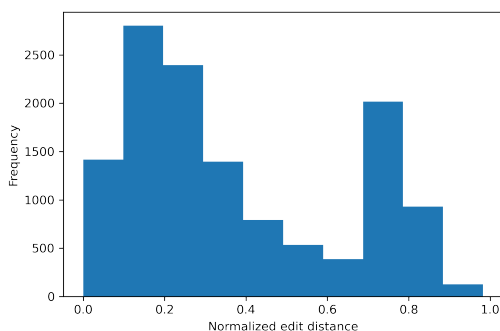


Figure 5.5: Normalized edit distances between original texts and those created during the editing phase for CEBaB.

5.10.2.5 Review-level Rating Phase

A total of 155 workers participated in this phase. Workers were paid US\$0.35 per batch of 10 examples. Figure 5.7 shows the annotation interface that workers used.

5.10.2.6 Randomly Selected Examples

Table 5.6 provides a random sample of edit pairs from CEBaB’s dev set.

5.10.2.7 Five-way Empirical ATE for CEBaB

Table 5.7 provides the binary $\widehat{\text{ATE}}$ values for CEBaB. These can be compared with the corresponding five-way values in Table 5.3d in the main text.

5.10.2.8 Edit variability

In the editing phase we ask human annotators to produce edits of an original review with regard to some concept. This is inherently a noisy process, which may impact the quality of our final benchmark. The CEBaB dataset features a modest set of paired edits (176 pairs in total). Each of these pairs contains two edits, starting from the same original sentence and edit goal, which results in two different edited sentences. Like all sentences in CEBaB, these edits were labeled for their review score by human annotators.

Figure 5.8a shows the distribution of the difference in final review majorities produced by these paired edits. Most paired edits differ at most by one star in their final majority rating, indicating that in general there is some noise associated with the editing procedure, but this does not have a major impact on the final review score. Figure 5.8b shows the same distribution when we consider the average review score an edit received, as opposed to the majority score. If we consider these

Instructions

You will be shown 10 short reviews of restaurants. For each, your task is to answer a simple question about it.

Example 1

Review: The food and ambiance were great, but the service was very slow.

Question: What is the **service evaluation** in this review?

Positive

Negative

Can't tell

Example 2

Review: Great food; drab, depressing decor, though.

Question: What is the **food evaluation** in this review?

Positive

Negative

Can't tell

Previewing Answers Submitted by Workers
×

This message is only visible to you and will not be shown to Workers.
You can test completing the task below and click "Submit" in order to preview the data and format of the submitted results.

\$(HITS)

If you have any questions/suggestions about this task, feel free to leave a comment here. We appreciate your input on this task.

Comment...

Submit

Figure 5.6: Validation phase annotation interface.

Instructions

You will be shown 10 short reviews of restaurants. For each, your task is to guess what star rating the author of the review chose.

Previewing Answers Submitted by Workers
×

This message is only visible to you and will not be shown to Workers.
You can test completing the task below and click "Submit" in order to preview the data and format of the submitted results.

1

Review: \$(item1_description)

Question: What star rating did the author attach to this review?

1 star: terrible

2 stars

3 stars: ambivalent or mixed

4 stars

5 stars: excellent

Figure 5.7: Review-level annotation interface.

description	original?	aspect	edit goal	aspect labels	aspect maj.	review labels	review maj.
Food was disgusting and very unreasonable!!!!!! Every request was honored and very friendly staff.\nHomemade bread which was foul.....	False	food	-	-, -, -, -, -	-	2, 2, 2, 2, 2	2
Every request was honored and very friendly staff.	False	food	unk.	unk, unk, unk, +, +	unk.	5, 5, 5, 4, 4	5
The food was average, but the service was terrible.	True	food	None	-, -, -, unk, +	-	2, 2, 2, 3, 3	2
The food was above average, but the service was terrible.	False	food	+	+, +, +, +, +	+	3, 3, 3, 3, 2	3
We hated our afternoon at Shorebreak!	False	ambiance	-	-, -, -, unk, unk	-	1, 1, 1, 1, 1	1
We loved our afternoon at Shorebreak!	False	ambiance	unk.	unk, unk, unk, -, +	unk.	5, 5, 5, 4, 4	5
The Sunday Jazz Brunch is great - Good music and fine, creative food. The service was great, my server answered all of my questions. The ambiance is quiet, but not so quiet as to inhibit conversation. A wonderful way to spend an early Sunday afternoon.	False	service	+	+, +, +, +, +	+	5, 5, 5, 4, 4	5
The Sunday Jazz Brunch is great - Good music and fine, creative food. The ambiance is quite, but not so quite as to inhibit conversation. A wonderful way to spend an early Sunday afternoon. The only bad spot was the horrid service.	False	service	-	-, -, -, -, -	-	4, 4, 4, 4, 3	4
My pasta dish was flavorless and rubbery and my husband's was cold. At least it 45 minutes to get it. Very poor, indeed.	True	food	None	-, -, -, -, -	-	1, 1, 1, 2, 2	1
My pasta dish was amazing and cooked great. At least it 45 minutes to get it. Very poor, indeed.	False	food	+	+, +, +, +, -	+	3, 3, 3, 3, 1	3
liked the restaurant a lot and loved the meal. Found the chicken great!	False	food	+	+, +, +, +, +	+	5, 5, 5, 4, 3	5
I liked the restaurant a lot,	False	food	unk.	unk, unk, unk, unk, +	unk.	5, 5, 5, 4, 4	5
At the heart of it, this is a HOTEL restaurant.	True	noise	None	unk, unk, unk, unk, unk	unk.	3, 3, 3, 3, 2	3
At the heart of it, this is an extremely loud restaurant.	False	noise	-	-, -, -, -, -	-	1, 1, 1, 3, 2	1
I was expecting some dishes from the Northern Italian Cuisine. The menu was not distinguishable from any other chain. The food was good but no differentiation. It was noisy, but I believe by design.	True	food	None	+, +, +, +, +	+	3, 3, 3, 4, 2	3
I was expecting some dishes from the Northern Italian Cuisine. The menu was not distinguishable from any other chain. The food was even worse than that. It was also noisy, but I believe by design.	False	food	-	-, -, -, -, +	-	1, 1, 1, 2, 2	1

Table 5.6: Randomly sampled edit pairs from CEBaB.

Table 5.7: Empirical $\widehat{\text{ATE}}$ for the binary sentiment labels in CEBaB. Reversing concept order results in the negation of the value given.

	Neg to Pos	Neg to Unk	Pos to Unk
food	0.77	0.49	-0.41
service	0.25	0.20	-0.16
ambiance	0.14	0.18	-0.14
noise	0.08	0.04	-0.14

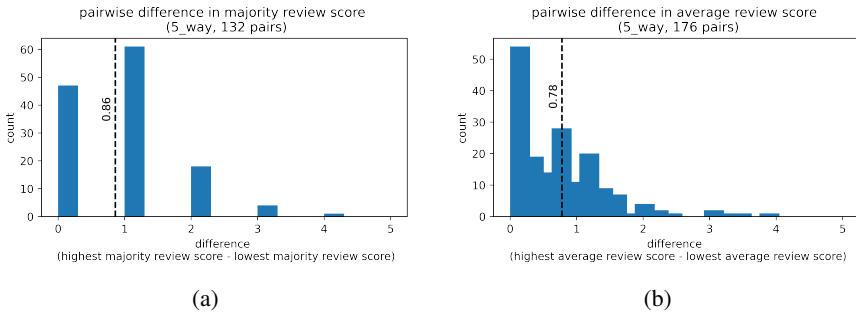


Figure 5.8: Pairwise absolute difference in majority (a) and average (b) review score for all double edits. Figure (a) only considers the 132 pairs where both edits have an actual review majority. Figure (b) considers all 176 pairs. Averages of the distributions are shown with a dotted vertical line.

average scores, most of the paired edits differ only slightly in their resulting review score.

Figures 5.9a-c shows the distribution of this pairwise review score in more detail. In an idealized setting without variability, the distribution would be centered around the diagonal of the heatmap. When going from 5-way classification to ternary and binary classification, the variability introduced by the edits becomes less relevant with regard to the final review majority label.

5.10.3 CEBaB Modeling Experiments

This section reports on standard classifier-based experiments with CEBaB, aimed at providing a sense for the dataset when it is used as a standard supervised sentiment dataset. We report experiments on the aspect-level and review-level ratings. In addition, we present evidence that author identity does not have predictive value.

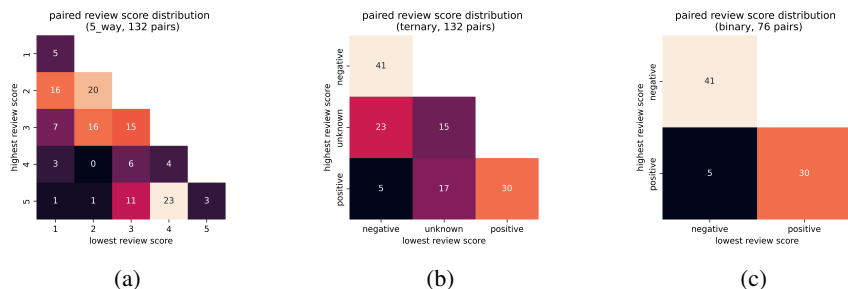


Figure 5.9: Pairwise review majority distribution for all double edits in 5-way (a), ternary (b), and binary (c) classification settings. Figures (a) and (b) consider only the 132 pairs where both edits have an actual review majority. Figure (c) considers the 76 pairs that have both a review majority and non-neutral labels.

5.10.3.1 Experiments Set-up

We rely on the Hugging Face `transformers` library.⁵ [Wolf et al., 2019] We train our models with 4 Nvidia 2080 Ti RTX 11GB GPUs on a single node machine. We use a maximum sequence length of 128 with a fix batch size of 32 with a initial learning rate of $2e^{-5}$. We run each experiment 5 times with distinct random seeds. We train our models with a minimum epoch number of 5 with our largest training set. We linearly scale our training epoch number by the size of the training set. We skip hyperparameter tuning for optimized task performance as our goal for this paper is to evaluate explanation methods. We release all of our models on Huggingface Dataset Hub.

5.10.3.2 Models

We include 4 different types of models, including BERT (`bert-base-uncased`) [Devlin et al., 2019], RoBERTa (`roberta-base`) [Liu et al., 2019], GPT-2 (`gpt2`) [Radford et al., 2019], as well as LSTM with dot-attention [Luong et al., 2015]. Our LSTM model uses `bert-base-uncased` tokenizer for simplicity. We initialize the embeddings of tokens for our LSTM using fastText [Joulin et al., 2016]. We reconfigure the classification head all other models the same classification head as in RoBERTa as a non-linear multilayer perceptron (MLP).⁶

⁵<https://github.com/huggingface/transformers>

⁶We implemented T5 (`t5-base`; [Raffel et al., 2019]) as a text-to-text model with the goal of treating predicted tokens as class labels. However, this raised unanticipated implementation questions concerning how to post-process multi-token class labels (e.g., “very positive”) for use in our explainer methods. As a result, we have elected to leave the T5 results out of the current draft, but we intend to include them in the next version once they have been more thoroughly vetted.

Table 5.8: Model performance results for sequence classification as well as aspect-based sentiment analysis (ABSA) under 3 training conditions. Mean Macro-F1 scores across 5 runs with distinct random seeds are reported.

Model	Exclusive				Inclusive			
	Binary	Ternary	5-way	ABSA	Binary	Ternary	5-way	ABSA
dev split								
BERT	0.97	0.82	0.68	0.88	0.98	0.85	0.72	0.90
GPT-2	0.97	0.80	0.67	0.88	0.98	0.84	0.70	0.89
LSTM	0.94	0.75	0.59	0.83	0.96	0.82	0.68	0.87
RoBERTa	0.99	0.83	0.71	0.89	0.99	0.86	0.76	0.90
test split								
BERT	0.97	0.82	0.70	0.87	0.98	0.84	0.73	0.89
GPT-2	0.97	0.80	0.65	0.87	0.97	0.83	0.68	0.89
LSTM	0.94	0.75	0.60	0.82	0.96	0.81	0.68	0.87
RoBERTa	0.98	0.83	0.70	0.88	0.99	0.86	0.75	0.90

5.10.3.3 Multi-class Sentiment Analysis Benchmark

We report model performance results under 3 training conditions: **Binary Classification**, where we label reviews with 1 star and 2 star ratings as negative, reviews with 4 star and 5 star as positive, and 3-star reviews are dropped; **Ternary Classification**, where we add another neutral class for reviews with 3 star ratings; and **5-way Classification**, where each star rating by itself is considered as a class. We leave out reviews in the train set in the ‘no majority’ category. (Dev and Test do not contain any such examples.) Table 5.8 shows the performance results for our models under different conditions. Our results suggest that RoBERTa has the edge over others across all evaluated tasks.

5.10.3.4 Aspect-based Sentiment Analysis Benchmark

Our dataset can be naturally used as an aspect-based sentiment analysis (ABSA) benchmark. For each sentence, it may contain up to 4 aspects with respect to the reviewing restaurant. As ABSA benchmarks are usually small and sparse with missing labels, our dataset provides validated aspect-based labels, and is one of the largest human validated ABSA benchmark.

To evaluate model performance, we adapt standard finetuning approach for ABSA benchmarks as proposed by Sun et al. [2019]. Instead of single sentence classification, we add another auxiliary sentence representing the aspect. For instance, to predict the label for the ‘food’ aspect for “the food here is good but not the service”, we append a single aspect token with a separator, and construct our input sentence as “the food here is good but not the service [SEP] food”. Table 5.8 shows the performance results for our models under different conditions.

Table 5.9: Model performance on top-k author identity prediction with number of train and dev examples.

Model	Accuracy	Macro-F1	# train	# dev
Random (k=5)	0.16	0.15	1105	227
Random (k=10)	0.10	0.10	2072	519
Random (k=15)	0.07	0.07	2963	741
RoBERTa (k=5)	0.27	0.16	1105	227
RoBERTa (k=10)	0.14	0.05	2072	519
RoBERTa (k=15)	0.11	0.04	2963	741

5.10.3.5 Author Identity Prediction

One potential artifact of our benchmark is edited sentence may expose author identity, which may result in artifact in interpreting model performance. To quantify this potential artifact, we train models to predict author identities based on the sentences. We create author identity prediction dataset by aggregating our dataset by anonymized worker ids. We then split the dataset into train/dev with a 4-to-1 ratio. For model training, we finetune RoBERTa for 5 epochs with a batch size of 32, a learning rate of $2e^{-5}$, and a maximum sequence length of 128. Note that we only consider top-k annotators ranked by their contributions (i.e., number of examples in our dataset). Table 5.9 shows the performance results of our finetuned models with a random classifier. Our results suggest that potential artifacts may exist but only for a limited extend.

5.10.4 Additional Results

In this section, we report additional results for bert-base-uncased, gpt-2, roberta-base, and an LSTM, fine-tuned on binary, ternary and 5-way versions of the sentiment task. These models are described in Appendix 5.10.3. Table 6.6 summarizes all the results.

We refer to the results section in the main text for an explanation of the different metrics considered. Which metric is best depends on the final use-case and whether it is more important to estimate the direction or the magnitude of the effect.

ICaCE-cosine Figure 5.10 shows the results for the ICaCE-Error with the *cosine* distance metric. The explanation methods that take the direction of the intervention into account (Approx, CONEXP, S-Learner) are the clear winners across all different models considered. S-Learner marginally wins across the most settings, but the conceptually simple Approx baseline is a close second. The strong performance of this simple baseline across the board suggests that most methods perform subpar, and that there is potential value in developing better concept-based model explanation methods.

Both TCAV and ConceptSHAP struggle to achieve better-than-random performance across all settings. Further analysis is needed to exactly understand why these methods are struggling.

Some additional trends emerge that require more analysis to fully understand. For example, Approx generally increases in performance when evaluated on more fine-grained classification settings, while CONEXP is typically worse here.

ICaCE-normdiff Figure 5.11 shows the results for the ICaCE-Error with the *normdiff* distance metric. In general, it is more difficult for explanation methods to estimate the magnitude of the intervention effect when the task increases in complexity. For a given explanation method and model, best results are often achieved for the binary classification problem.

The conceptually simple Approx baseline wins across the board. S-Learner is only able to match its performance a few times. While previous results already showed that most of the methods fall behind the Approx baseline, the results are particularly striking for this metric.

While S-learner and CONEXP were somewhat comparable on the *cosine* metric, their differences become clear on the *normdiff* metric: S-Learner is better at estimating the magnitude of the intervention.

An interesting trend can be observed for TCAV, which has good performance on the binary task but becomes worse than random when evaluated on the ternary and 5-way settings. ConceptSHAP is the only method that consistently breaks the upward trend when going from ternary to the 5-way setting. More analysis is needed to understand both these phenomena.

ICaCE-L2 Figure 5.12 shows the results for the ICaCE-Error with the *L2* distance metric. Because this metric takes both the scale and direction of the effect into account, it is slightly harder to interpret. In general, the performance drops when evaluated on more fine-grained classification settings.

Again, the Approx baseline is a strong contestant, but on this metric the results are more varied. S-Learner is consistently the best at producing the closest explanation in Euclidian distance to the real effect for the 5-way setting.

5.10.5 CausaLM

5.10.5.1 Our adaptation

The CausaLM algorithm was originally designed to estimate the average treatment effect of a high-level concept on pre-trained language models. Its output estimator is the textual representation averaged treatment effect (TReATE), which is computed as:

$$\text{TReATE}_{\mathcal{N}_\phi}(C; \mathcal{D}) = \frac{1}{|\mathcal{D}|} \sum_{x \in \mathcal{D}} \mathcal{N}'(\phi_C^{\text{CF}}(x)) - \mathcal{N}(\phi(x)), \quad (5.21)$$

where ϕ_C^{CF} denotes the learned counterfactual representation that information about concept C is not present, \mathcal{N}' is a classifier trained on this counterfactual representation, and \mathcal{D} is a dataset.

However, for comparison on the CEBaB data, we require the estimation of individual causal concept effects (ICaCE). To allow a fair comparison, we swap the TReATE output estimator with TReITE (Equation 5.10). The only difference

(a) ICaCE scores for 5-way sentiment classification setting.

Model	Metric	Random	Approx	CONEXP	S-Learner	TCAV	ConceptSHAP	CausaLM	INLP
BERT	L2 _{ICaCE}	0.94 (.01)	0.81 (.01)	0.82 (.02)	0.74 (.02)	0.82 (.01)	1.25 (.01)	0.86 (.01)	0.80 (.02)
	CO _S ICaCE	1.00 (.00)	0.61 (.01)	0.72 (.01)	0.63 (.01)	1.00 (.00)	1.11 (.01)	0.78 (.00)	0.59 (.03)
	NormDiff _{ICaCE}	0.67 (.02)	0.44 (.01)	0.62 (.02)	0.54 (.02)	0.78 (.02)	0.56 (.02)	0.68 (.02)	0.73 (.02)
RoBERTa	L2 _{ICaCE}	0.97 (.01)	0.83 (.01)	0.86 (.01)	0.78 (.01)	0.85 (.01)	1.24 (.01)	0.90 (.01)	0.84 (.01)
	CO _S ICaCE	1.00 (.01)	0.60 (.01)	0.74 (.00)	0.64 (.01)	1.01 (.00)	1.06 (.01)	0.77 (.00)	0.58 (.01)
	NormDiff _{ICaCE}	0.72 (.01)	0.45 (.00)	0.67 (.01)	0.59 (.01)	0.83 (.01)	0.61 (.00)	0.74 (.00)	0.81 (.01)
GPT-2	L2 _{ICaCE}	0.81 (.02)	0.72 (.02)	0.68 (.02)	0.60 (.02)	0.68 (.02)	1.03 (.02)	0.76 (.02)	0.72 (.01)
	CO _S ICaCE	1.00 (.00)	0.59 (.01)	0.67 (.00)	0.59 (.01)	1.00 (.00)	1.00 (.00)	0.82 (.01)	1.00 (.00)
	NormDiff _{ICaCE}	0.52 (.02)	0.41 (.01)	0.47 (.02)	0.40 (.01)	0.65 (.02)	0.46 (.01)	0.52 (.02)	0.58 (.03)
LSTM	L2 _{ICaCE}	0.89 (.01)	0.86 (.01)	0.79 (.01)	0.73 (.01)	0.78 (.02)	1.27 (.04)	0.76 (.01)	0.79 (.01)
	CO _S ICaCE	1.00 (.01)	0.64 (.01)	0.71 (.00)	0.64 (.01)	1.02 (.01)	1.00 (.00)	1.00 (.00)	0.74 (.02)
	NormDiff _{ICaCE}	0.62 (.01)	0.50 (.01)	0.59 (.01)	0.53 (.01)	0.70 (.01)	0.54 (.00)	0.76 (.01)	0.60 (.01)

(b) ICaCE scores for ternary sentiment classification setting.

Model	Metric	Random	Approx	CONEXP	S-Learner	TCAV	ConceptSHAP	CausaLM	INLP
BERT	L2 _{ICaCE}	0.79 (.01)	0.54 (.01)	0.65 (.00)	0.56 (.00)	0.56 (.00)	0.94 (.01)	0.72 (.00)	0.58 (.01)
	CO _S ICaCE	0.99 (.02)	0.61 (.02)	0.64 (.04)	0.54 (.04)	1.00 (.03)	1.21 (.01)	0.76 (.01)	0.69 (.01)
	NormDiff _{ICaCE}	0.60 (.00)	0.42 (.01)	0.54 (.00)	0.48 (.00)	0.55 (.00)	0.62 (.01)	0.62 (.00)	0.55 (.01)
RoBERTa	L2 _{ICaCE}	0.79 (.01)	0.56 (.00)	0.65 (.01)	0.57 (.01)	0.55 (.01)	0.88 (.02)	0.74 (.01)	0.55 (.01)
	CO _S ICaCE	1.00 (.01)	0.62 (.01)	0.73 (.02)	0.62 (.02)	0.99 (.01)	1.12 (.02)	0.76 (.01)	0.72 (.01)
	NormDiff _{ICaCE}	0.61 (.01)	0.43 (.00)	0.54 (.00)	0.48 (.00)	0.54 (.00)	0.61 (.01)	0.66 (.01)	0.54 (.01)
GPT-2	L2 _{ICaCE}	0.75 (.01)	0.57 (.01)	0.60 (.01)	0.52 (.01)	0.52 (.01)	0.69 (.01)	0.68 (.01)	0.61 (.03)
	CO _S ICaCE	1.00 (.01)	0.63 (.01)	0.59 (.01)	0.50 (.01)	1.00 (.00)	1.01 (.00)	0.79 (.01)	1.00 (.00)
	NormDiff _{ICaCE}	0.54 (.01)	0.42 (.01)	0.47 (.01)	0.42 (.01)	0.51 (.01)	0.52 (.01)	0.55 (.01)	0.51 (.01)
LSTM	L2 _{ICaCE}	0.76 (.00)	0.58 (.01)	0.63 (.01)	0.55 (.01)	0.55 (.01)	1.03 (.04)	0.53 (.01)	0.68 (.01)
	CO _S ICaCE	1.00 (.01)	0.67 (.01)	0.63 (.00)	0.60 (.01)	1.01 (.01)	1.01 (.01)	1.00 (.00)	0.78 (.02)
	NormDiff _{ICaCE}	0.56 (.01)	0.45 (.01)	0.51 (.00)	0.46 (.01)	0.51 (.01)	0.65 (.01)	0.52 (.01)	0.56 (.01)

(c) ICaCE scores for binary sentiment classification setting.

Model	Metric	Random	Approx	CONEXP	S-Learner	TCAV	ConceptSHAP	CausaLM	INLP
BERT	L2 _{ICaCE}	0.60 (.01)	0.19 (.01)	0.51 (.00)	0.31 (.00)	0.31 (.01)	0.76 (.06)	0.57 (.01)	0.51 (.05)
	CO _S ICaCE	0.99 (.01)	0.75 (.04)	0.64 (.05)	0.66 (.04)	1.00 (.01)	1.20 (.02)	0.80 (.01)	0.79 (.00)
	NormDiff _{ICaCE}	0.52 (.01)	0.19 (.01)	0.50 (.00)	0.30 (.00)	0.30 (.01)	0.55 (.05)	0.56 (.01)	0.50 (.04)
RoBERTa	L2 _{ICaCE}	0.59 (.01)	0.18 (.01)	0.51 (.00)	0.31 (.00)	0.29 (.01)	0.68 (.06)	0.61 (.00)	0.31 (.01)
	CO _S ICaCE	1.00 (.01)	0.78 (.02)	0.70 (.03)	0.71 (.03)	1.00 (.01)	1.12 (.02)	0.82 (.00)	0.80 (.00)
	NormDiff _{ICaCE}	0.50 (.01)	0.18 (.00)	0.51 (.00)	0.31 (.00)	0.29 (.01)	0.54 (.04)	0.60 (.00)	0.31 (.01)
GPT-2	L2 _{ICaCE}	0.59 (.00)	0.19 (.01)	0.50 (.00)	0.31 (.00)	0.29 (.00)	0.39 (.01)	0.55 (.01)	0.45 (.01)
	CO _S ICaCE	1.01 (.01)	0.69 (.01)	0.58 (.01)	0.61 (.01)	1.00 (.00)	1.02 (.00)	0.79 (.01)	1.00 (.00)
	NormDiff _{ICaCE}	0.51 (.01)	0.19 (.01)	0.50 (.00)	0.31 (.00)	0.29 (.00)	0.35 (.01)	0.53 (.01)	0.41 (.01)
LSTM	L2 _{ICaCE}	0.58 (.01)	0.20 (.01)	0.51 (.00)	0.32 (.01)	0.31 (.00)	0.78 (.05)	0.28 (.00)	0.47 (.01)
	CO _S ICaCE	1.00 (.01)	0.77 (.00)	0.70 (.01)	0.71 (.01)	1.01 (.01)	1.00 (.00)	1.00 (.00)	0.81 (.00)
	NormDiff _{ICaCE}	0.50 (.01)	0.20 (.01)	0.50 (.00)	0.32 (.01)	0.29 (.00)	0.64 (.04)	0.28 (.00)	0.46 (.01)

Table 5.10: ICaCE scores on the test set for the binary, ternary and 5-way classification settings. **Lower is better**. Results averaged over 5 distinct seeds; standard deviations in parentheses.

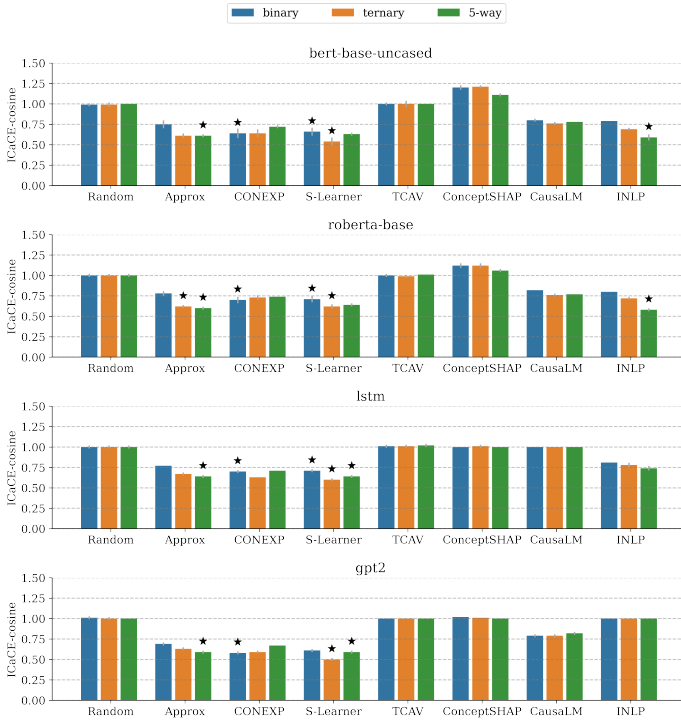


Figure 5.10: ICaCE-Error for all experiments using the *cosine* distance metric. **Lower is better.** Results averaged over 5 distinct seeds. Error bars (in gray) display the standard deviation. Stars denote the best results for a given classification setting.

between these estimators is that in TReITE we remove the average across \mathcal{D} , and output the estimated effect of individual examples.

5.10.5.2 Implementation details

For all counterfactual models, we optimize using the established Adam optimizer with $\text{lr}=2\text{e-}5$, $\text{epochs}=3$, $\text{batch_size}=48$, and the relative weight of the adversarial task, λ , is set to 0.1.

For both the factual models and fine-tuning phase, we optimize using the Adam optimizer with $\text{lr}=1\text{e-}3$, $\text{epochs}=50$, and $\text{batch_size}=256$. The differences in hyperparameter values is due to the different architectures we employ; for the counterfactual models we train the entire language model (ϕ), and for the factual models and the fine-tuning phase we freeze the embedding weights (ϕ) and train only the classification head (\mathcal{N}).

All CausaLM models were trained using 2 Nvidia GTX 1080 Ti 12GB GPUs.

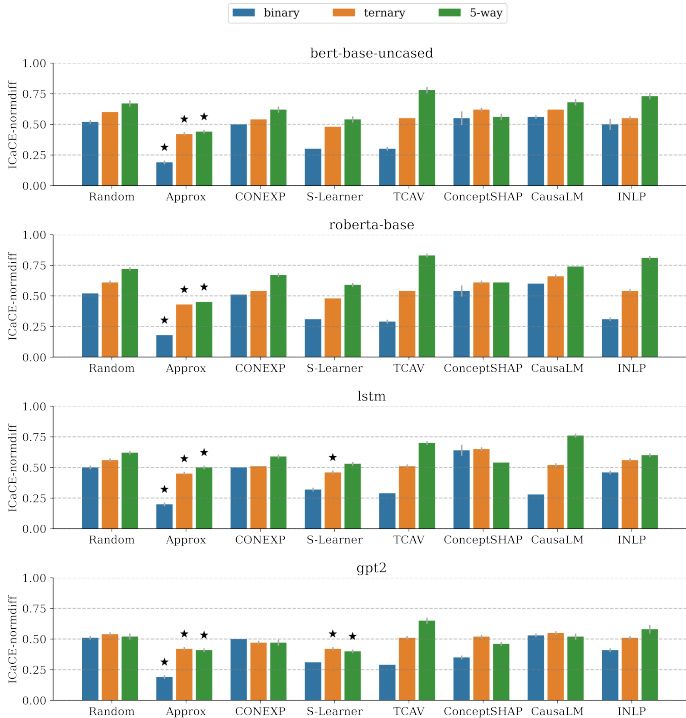


Figure 5.11: ICaCE-Error for all experiments using the *normdiff* distance metric. **Lower is better.** Results averaged over 5 distinct seeds. Error bars (in gray) display the standard deviation. Stars denote the best results for a given classification setting.

5.10.6 INLP

5.10.7 Our adaptation

The INLP algorithm was originally designed to debias word embeddings by iteratively projecting them onto the null-space of some protected attribute (concept). However, INLP may serve as an estimation method similar to CausaLM, with the two following crucial differences. First, its lack of ability to control for potential confounders. Second, it operates on the representation rather than on the actual model weights. Since CausaLM and INLP share common characteristics, their output estimators are computed in the same way. See §5.10.5 for extended details.

5.10.7.1 Implementation details

In order to guard for a “protected attribute” (concept), INLP determines whether this concept is present in an embedding or not by learning a linear separator in the embedding space. Following the practice suggested in the original paper, we choose

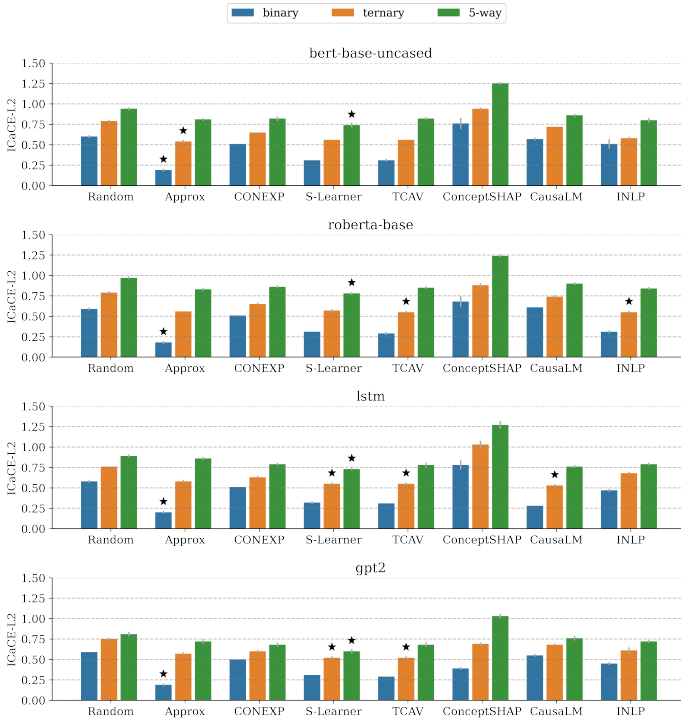


Figure 5.12: ICaCE-Error for all experiments using the L_2 distance metric. **Lower is better**. Results averaged over 5 distinct seeds. Error bars (in gray) display the standard deviation. Stars denote the best results for a given classification setting.

our linear separator to be an SVM learned using SGD with $\alpha = 0.01$, $\varepsilon = 0.001$, and $\text{max_iter}=1000$. Logistic regression showed similar behavior. We project the representation to the null-space with respect to the concept 10 times. In fact, and similarly to the original paper, we converge to random accuracy of predicting the concept from the counterfactual representation after 4-5 iterations.

For all concepts, the classification head on top of the language model that trained to predict the overall sentiment labels trains for 5 epochs using the Adam optimizer with $\text{lr}=2e-5$.

5.10.8 TCAV

5.10.8.1 Our adaptation

The Testing with Concept Activation Vectors (TCAV) explanation method was originally designed to count the percentage of test inputs from dataset \mathcal{D} that are positively influenced by some high-level concept. It outputs a count over the number

of examples that are change towards the direction of concept C , and computed as:

$$\text{TCAV}_{\mathcal{N}_\phi}(k, C; \mathcal{D}) = \frac{|\{x \in \mathcal{D} : \nabla \mathcal{N}_k(\phi(x)) \cdot v_C > 0\}|}{|\mathcal{D}|}, \quad (5.22)$$

where k is some class index and v_C is a linear direction in the activation space, given by the coefficients of a linear separator trained to distinguish between examples that include or exclude the concept C .

While TCAV’s output is a count over examples, we use the raw sensitivity (directional derivative). This approach is supported by the authors of the original paper: “one could also use a different metric that considers the magnitude of the conceptual sensitivities” Kim et al. [2018]. Also, since TCAV operates on the gradients of a model’s logits but the ICaCEs are the difference of two probability vectors, we normalize its outputs by taking Tanh .

5.10.8.2 Implementation details

To learn the Concept Activation Vector (CAV, i.e., a linear direction in the activation space of ϕ), we train a linear separator to distinguish between examples that include the concept (labeled positive or negative) and examples that do not include it (labeled unknown). When learning CAVs, we drop all CEBaB train examples that are not labeled for aspect (concept) or do not have a majority with respect to the aspect.

Identically to the original paper, our CAV linear separator is an SVM learned using SGD with $\alpha = 0.01$, $\varepsilon = 0.001$ and $\text{max_iter} = 1000$.

5.10.9 ConceptSHAP

5.10.9.1 Our adaptation

The original ConceptSHAP algorithm takes a complete set of concepts $C \in \{C_1, \dots, C_m\}$ (such that its completeness score in Equation 5.25 is higher than some threshold) and outputs the relative contribution to the test accuracy of each C_i . It outputs an estimator given by the following formula

$$\text{Shapley}_{\{C_1, \dots, C_m\}}(C) = \sum_{S \subseteq \{C_1, \dots, C_m\} \setminus C} \frac{(m - |S| - 1)! |S|!}{m!} [\eta(S \cup \{C\}) - \eta(S)], \quad (5.23)$$

where η is a scoring function operating on sets of concepts that output accuracy ratios.

Similarly to the other methods, if η outputs accuracy ratios, then the output of ConceptSHAP is not a suitable estimator for ICaCE. Our straightforward adaptation for ConceptSHAP is to make η output class probabilities for classes instead of accuracy ratios.

Our adapted version outputs a vector for each $C \in \{C_1, \dots, C_m\}$ and x according to the following equation:

$$\text{ConceptSHAP}_{\mathcal{N}_\phi}(C; x) = \sum_{S \subseteq \{C_1, \dots, C_m\} \setminus C} \frac{(m - |S| - 1)! |S|!}{m!} [\eta(S \cup \{C\}) - \eta(S)], \quad (5.24)$$

where η is a function defined as $\eta_{\mathcal{N}_\phi}(S) = \sup_g \mathcal{N}(g(V_S \phi(x)))$, and V_S is a matrix with the learned concept directions as its rows $V_S = (v_C^T)_{C \in S} \in \mathbb{R}^{|S| \times h}$.

Yeh et al. [2020] calculate concept directions v_{C_j} automatically by learning a neural network classifier. To allow for a fair comparison between ConceptSHAP and the other evaluated methods, we use the concept activation vectors v_{C_1}, \dots, v_{C_m} as the input concepts (similarly to those used in Kim et al. [2018]).

In addition, in the original paper the authors learn the concepts v_C automatically, by using a carefully constructed loss function. To allow a fair comparison, we learn the concept vector by exploiting our labeled aspects (concepts), in a way similar to TCAV. See Section 5.10.8.2 for more details.

5.10.9.2 Completeness Scores of Treatment Concepts

Given a feature representation ϕ and a classification head \mathcal{N} , the completeness score is defined by:

$$\text{completeness}_{\mathcal{N}_\phi}(S; D, Y) = \frac{\sup_g \frac{1}{|D|} \sum_{(x,y) \in D, Y} \mathbb{1}[y = \arg \max_{y'} \mathcal{N}_{y'}(g(V_S \phi(x)))] - a_r}{\frac{1}{|D|} \sum_{(x,y) \in D, Y} \mathbb{1}[y = \arg \max_{y'} \mathcal{N}_{y'}(\phi(x))] - a_r}, \quad (5.25)$$

where a_r is the accuracy of a classifier that outputs random predictions, $S \subseteq \{C_1, \dots, C_m\}$ and V_S is a matrix with the learned concept directions as its rows $V_S = (v_C^T)_{C \in S} \in \mathbb{R}^{|S| \times h}$.

For all models, the completeness we get for the set of concepts is larger than 0.9, for $S = \{\text{ambiance, food, service, noise}\}$.

5.10.9.3 Hyperparameters

The hyperparameters for CAV are identical to those of TCAV (Section 5.10.8.2). To calculate η and the completeness score, we follow the original paper and set g to be a two-layer perceptron with 500 hidden units, learned using Adam optimizer for 50 epochs, employing `lr=1e-2` and `batch_size=128`.

6

Technical: Learning to Interpret

In this chapter, we aim to improve the faithfulness of AI interpretability methods, building on the data and formalisms developed in the previous chapter for measuring AI interpretations. We propose a radical new approach to AI interpretability by using a secondary black-box system to interpret the primary AI model. This method involves learning to predict how the primary model will respond to changes in the input, and is trained using the counterfactual dataset created in the previous chapter. The results demonstrate significantly more faithful explanations of the initial model's behavior, paving the way for AI model to faithfully interpret other AI models.

* * *

Causal Proxy Models for Concept-based Model Explanations

Zhengxuan Wu*, **Karel D'Oosterlinck***, **Atticus Geiger***, **Amir Zur**, **Christopher Potts**

Proceedings of the 40th International Conference on Machine Learning (ICML 2023)

*Equal contribution.

6.1 Abstract

Explainability methods for NLP systems encounter a version of the fundamental problem of causal inference: for a given ground-truth input text, we never truly observe the counterfactual texts necessary for isolating the causal effects of model representations on outputs. In response, many explainability methods make no use of counterfactual texts, assuming they will be unavailable. In this paper, we show that robust causal explainability methods can be created using approximate counterfactuals, which can be written by humans to approximate a specific counterfactual or simply sampled using metadata-guided heuristics. The core of our proposal is the Causal Proxy Model (CPM). A CPM explains a black-box model \mathcal{N} because it is trained to have the same *actual* input/output behavior as \mathcal{N} while creating neural representations that can be intervened upon to simulate the *counterfactual* input/output behavior of \mathcal{N} . Furthermore, we show that the best CPM for \mathcal{N} performs comparably to \mathcal{N} in making factual predictions, which means that the CPM can simply replace \mathcal{N} , leading to more explainable deployed models.

6.2 Introduction

The gold standard for explanation methods in AI should be to elucidate the *causal role* that a model’s representations play in its overall behavior – to truly explain *why* the model makes the predictions it does. Causal explanation methods seek to do this by resolving the counterfactual question of what the model would do if input X were changed to a relevant counterfactual version X' . Unfortunately, even though neural networks are fully observed, deterministic systems, we still encounter the fundamental problem of causal inference [Holland, 1986]: for a given ground-truth input X , we never observe the counterfactual inputs X' necessary for isolating the causal effects of model representations on outputs. The issue is especially pressing in domains where it is hard to synthesize approximate counterfactuals. In response to this, explanation methods typically do not explicitly train on counterfactuals at all.

In this paper, we show that robust explanation methods for NLP models can be obtained using texts approximating true counterfactuals. The heart of our proposal is the Causal Proxy Model (CPM). CPMs are trained to mimic both the factual and counterfactual behavior of a black-box model \mathcal{N} . We explore two different methods for training such explainers. These methods share a distillation-style objective that pushes them to mimic the factual behavior of \mathcal{N} , but they differ in their counterfactual objectives. The input-based method CPM_{IN} appends to the factual input a new token associated with the counterfactual concept value. The hidden-state method CPM_{HI} employs the Interchange Intervention Training (IIT) method of Geiger et al. [2022] to localize information about the target concept in specific hidden states. Figure 6.1 provides a high-level overview.

We evaluate these methods on the CEBaB benchmark for causal explanation methods [Abraham et al., 2022], which provides large numbers of original examples

(restaurant reviews) with human-created counterfactuals for specific concepts (e.g., service quality), with all the texts labeled for their concept-level and text-level sentiment. We consider two types of approximate counterfactuals derived from CEBaB: texts written by humans to approximate a specific counterfactual, and texts sampled using metadata-guided heuristics. Both approximate counterfactual strategies lead to state-of-the-art performance on CEBaB for both CPM_{IN} and CPM_{HI} .

We additionally identify two other benefits of using CPMs to explain models. First, both CPM_{IN} and CPM_{HI} have factual performance comparable to that of the original black-box model \mathcal{N} and can explain their own behavior extremely well. Thus, the CPM for \mathcal{N} can actually replace \mathcal{N} , leading to more explainable deployed models. Second, CPM_{HI} models localize concept-level information in their hidden representations, which makes their behavior on specific inputs very easy to explain. We illustrate this using Path Integrated Gradients [Sundararajan et al., 2017], which we adapt to allow input-level attributions to be mediated by the intermediate states that were targeted for localization. Thus, while both CPM_{IN} and CPM_{HI} are comparable as explanation methods according to CEBaB, the qualitative insights afforded by CPM_{HI} models may give them the edge when it comes to explanations.

6.3 Related Work

Understanding model behavior serves many goals for large-scale AI systems, including transparency [Kim, 2015, Lipton, 2018, Pearl, 2019, Ehsan et al., 2021], trustworthiness [Ribeiro et al., 2016, Guidotti et al., 2018, Jacovi and Goldberg, 2020, Jakesch et al., 2019], safety [Amodei et al., 2016, Otte, 2013], and fairness [Hardt et al., 2016, Kleinberg et al., 2017, Goodman and Flaxman, 2017, Mehrabi et al., 2021]. With CPMs, our goal is to achieve explanations that are causally motivated and concept-based, and so we concentrate here on relating existing methods to these two goals.

Feature attribution methods estimate the importance of features, generally by inspecting learned weights directly or by perturbing features and studying the effects this has on model behavior [Molnar, 2020, Ribeiro et al., 2016]. Gradient-based feature attribution methods extend this general mode of explanation to the hidden representations in deep networks [Zeiler and Fergus, 2014, Springenberg et al., 2014, Binder et al., 2016, Shrikumar et al., 2017, Sundararajan et al., 2017]. Concept Activation Vectors (CAVs; Kim et al. 2018, Yeh et al. 2020) can also be considered feature attribution methods, as they probe for semantically meaningful directions in the model’s internal representations and use these to estimate the importance of concepts on the model predictions. While some methods in this space do have causal interpretations (e.g., Sundararajan et al. 2017, Yeh et al. 2020), most do not. In addition, most of these methods offer explanations in terms of specific (sets of) features/neurons. (Methods based on CAVs operate directly in terms of more abstract concepts.)

Intervention-based methods study model representations by modifying them in systematic ways and observing the resulting model behavior. These methods are generally causally motivated and allow for concept-based explanations. Examples of methods in this space include causal mediation analysis [Vig et al., 2020, De Cao et al., 2021, Ban et al., 2022], causal effect estimation [Feder et al., 2021b, Elazar et al., 2021, Abraham et al., 2022, Lovering and Pavlick, 2022], tensor product decomposition [Soulos et al., 2020], and causal abstraction analysis [Geiger et al., 2020, 2021]. CPMs are most closely related to the method of IIT [Geiger et al., 2021], which extends causal abstraction analysis to optimization.

Probing is another important class of explanation method. Traditional probes do not intervene on the target model, but rather only seek to find information in it via supervised models [Conneau et al., 2018, Tenney et al., 2019] or unsupervised models [Clark et al., 2019b, Manning et al., 2020, Saphra and Lopez, 2019]. Probes can identify concept-based information, but they cannot offer guarantees that probed information is relevant for model behavior [Geiger et al., 2021]. For causal guarantees, it is likely that some kind of intervention is required. For example, Elazar et al. [2021] and Feder et al. [2021b] remove information from model representations to estimate the causal role of that information. Our CPMs employ a similar set of guiding ideas but are not limited to removing information.

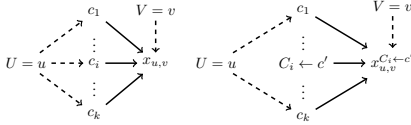
Counterfactual explanation methods aim to explain model behavior by providing a counterfactual example that changes the model behavior [Goyal et al., 2019, Verma et al., 2020, Wu et al., 2021]. Counterfactual explanation methods are inherently causal. If they can provide counterfactual examples with regard to specific concepts, they are also concept-based.

Some explanation methods train a model making explicit use of intermediate variables representing concepts. Manipulating these intermediate variables at inference time yields causal concept-based model explanations [Koh et al., 2020, Künzel et al., 2019].

Evaluating methods in this space has been a persistent challenge. In prior literature, explanation methods have often been evaluated against synthetic datasets [Feder et al., 2021b, Yeh et al., 2020]. In response, Abraham et al. [2022] introduced the CEBaB dataset, which provides a human-validated concept-based dataset to truthfully evaluate different causal concept-based model explanation methods. Our primary evaluations are conducted on CEBaB.

6.4 Causal Proxy Model (CPM)

Causal Proxy Models (CPMs) are causal concept-based explanation methods. Given a factual input $x_{u,v}$ and a description of a concept intervention $C_i \leftarrow c'$, they estimate the effect of the intervention on model output. The present section introduces our two core CPM variants in detail. We concentrate here on introducing the structure of these models and their objectives, and we save discussion of associated metrics for explanation methods for Section 6.5.



(a) A structural causal model leading to an actual text $x_{u,v}$ and its counterfactual text $x_{u,v}^{C_i \leftarrow c'}$. U is an exogenous variable over experiences, c_1, \dots, c_k are mediating concepts, and V is an exogenous variable capturing the writing (and star-rating) experience. At right, we create a counterfactual in which concept C_i takes on a different value. Unfortunately, we cannot truly create such counterfactual situations and so we never observe pairs of texts like these. Thus, we must rely on approximate counterfactuals.

Let $x_{u,v}$ be a text written in situation (u, v) :

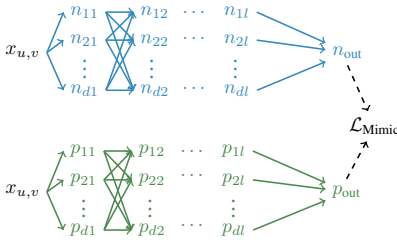
Human-created $\tilde{x}_{u,v}^{C_i \leftarrow c'}$

Crowdworker edit of $x_{u,v}$ to express that C_i had value c' , seeking to keep all else constant.

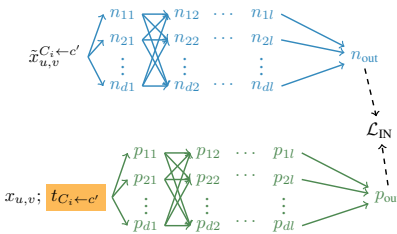
Metadata-sampled $\tilde{x}_{u,v}^{C_i \leftarrow c'}$

Sampled text expressing that C_i has value c' but agreeing with $x_{u,v}$ on all other concepts.

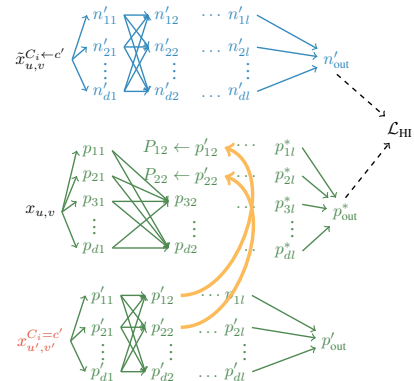
(b) Approximate counterfactuals. In the human-created strategy, humans revise a text to express a counterfactual, simulating a causal intervention. In the metadata-sampled strategy, we find a text that aligns with the original for u while differing in the target concept C_i .



(c) $\mathcal{L}_{\text{Mimic}}$: All CPMs (bottom) are trained to mimic the behavior of the neural model \mathcal{N} to be explained (top) for all factual inputs $x_{u,v}$.



(d) \mathcal{L}_{IN} : Examples $x_{u,v}$ and $\tilde{x}_{u,v}^{C_i \leftarrow c'}$ are an approximate counterfactual pair. The CPM is given $x_{u,v}$ augmented with a special token $t_{C_i \leftarrow c'}$ and trained to mimic the target model \mathcal{N} when its input is $\tilde{x}_{u,v}^{C_i \leftarrow c'}$.



(e) \mathcal{L}_{HI} : Examples $x_{u,v}$ and $\tilde{x}_{u,v}^{C_i \leftarrow c'}$ form an approximate counterfactual pair. The CPM (middle) is given $x_{u,v}$, aiming to mimic \mathcal{N} (top) based on $\tilde{x}_{u,v}^{C_i \leftarrow c'}$, while incorporating an intervention that changes internal states to those computed for the input $x_{u,v}^{C_i=c'}$ (bottom), a distinct example expressing $C_i = c'$. This intervention localizes information about C_i , as the CPM learns about $C_i \leftarrow c'$ solely through it.

Figure 6.1: Causal Proxy Model (CPM) summary. Every CPM for model \mathcal{N} is trained to mimic the factual behavior of \mathcal{N} ($\mathcal{L}_{\text{Mimic}}$). For CPM_{IN}, the counterfactual objective is \mathcal{L}_{IN} . For CPM_{HI}, the counterfactual objective is \mathcal{L}_{HI} .

A Structural Causal Model Our discussion is grounded in the causal model depicted in Figure 6.1a, which aligns well with the CEBaB benchmark. Two exogenous variables U and V together represent the complete state of the world and generate some textual data X . The effect of exogenous variable U on the data X is completely mediated by a set of intermediate variables $C_1, C_2 \dots, C_k$, which we refer to as *concepts*. Therefore, we can think of U as the part of the world that gives rise to these concepts $\{C\}_1^k$.

Using this causal model, we can describe counterfactual data – data that arose under a counterfactual state of the world (right diagram in Figure 6.1a). Our factual text is $x_{u,v}$, and we use $x_{u,v}^{C_i \leftarrow c'}$ for the counterfactual text obtained by intervening on concept C_i to set its value to c' . The counterfactual $x_{u,v}^{C_i \leftarrow c'}$ describes the output when the value of C_i is set to c' , all else being held equal.

Approximate Counterfactuals Unfortunately, pairs like $(x_{u,v}, x_{u,v}^{C_i \leftarrow c'})$ are never observed, and thus we need strategies for creating approximate counterfactuals $\tilde{x}_{u,v}^{C_i \leftarrow c'}$. Figure 6.1b describes the two strategies we use in this paper. In the human-created strategy, we rely on a crowdworker to edit $x_{u,v}$ to achieve a particular counterfactual goal – say, making the evaluation of the restaurant’s food negative. CEBaB contains an abundance of such pairs $(x_{u,v}, \tilde{x}_{u,v}^{C_i \leftarrow c'})$. However, CEBaB is unusual in having so many human-created approximate counterfactuals, so we also explore a simpler strategy in which $\tilde{x}_{u,v}^{C_i \leftarrow c'}$ is sampled with the requirement that it match $x_{u,v}$ on all concepts but sets C_i to c' . This strategy is supported in many real-world datasets – for example, the OpenTable reviews underlying CEBaB all have the needed metadata [Abraham et al., 2022].

CPM_{IN}: Input-based CPM Given a dataset of approximate counterfactual pairs $(x_{u,v}, \tilde{x}_{u,v}^{C_i \leftarrow c'})$ and a black-box model \mathcal{N} , we train a new CPM_{IN} model \mathcal{P} with a counterfactual objective as:

$$\mathcal{L}_{\text{IN}} = \text{CE}_S(\mathcal{N}(\tilde{x}_{u,v}^{C_i \leftarrow c'}), \mathcal{P}(x_{u,v}; t_{C_i \leftarrow c'})) \quad (6.1)$$

where $x_{u,v}; t_{C_i \leftarrow c'}$ in Eqn. 6.1 denotes the concatenation of the factual input and a randomly initialized learnable token embedding $t_{C_i \leftarrow c'}$ describing the intervention $C_i \leftarrow c'$. CE_S represents the smoothed cross-entropy loss [Hinton et al., 2015], measuring the divergence between the output logits of both models. The objective in Eqn. 6.1 pushes \mathcal{P} to predict the counterfactual behavior of \mathcal{N} when a descriptor of the intervention is given (Figure 6.1d).¹

At inference time, approximate counterfactuals are inaccessible. To explain model \mathcal{N} , we append the trained token embedding $t_{C_i \leftarrow c'}$ to a factual input, upon which \mathcal{P} predicts a counterfactual output for this input, used to estimate the counterfactual behavior of \mathcal{N} under this intervention.

CPM_{HI}: Hidden-state CPM Our CPM_{HI} models are trained on the same data and with the same set of goals as CPM_{IN}, to mimic both the factual and counterfactual

¹Our objective is regard to a single approximate counterfactual pair for the sake of clarity. At train-time, we aggregate the objective over all considered training pairs. We take C_i to always represent the intervened-upon concept. The weights of \mathcal{N} are frozen.

behavior of \mathcal{N} . The key difference is how the information about the intervention $C_i \leftarrow c'$ is exposed to the model. Specifically, we adapt Interchange Intervention Training [Geiger et al., 2022] to train our CPM_{HI} models for concept-based model explanation.

A conventional intervention on a hidden representation H of a neural network \mathcal{N} fixes the value of the representation H to a constant. In an interchange intervention, we instead fix H to the value it would have been when processing a separate source input s . The result of the interchange intervention is a new model. Formally, we describe this new model as $\mathcal{N}_{H \leftarrow H_s}$, where \leftarrow is the conventional intervention operator and H_s is the value of hidden representation H when processing input s .

Given a dataset of approximate counterfactual input pairs $(x_{u,v}, \tilde{x}_{u,v}^{C_i \leftarrow c'})$ and a black-box model \mathcal{N} , we train a new CPM_{HI} model \mathcal{P} with the following counterfactual objective:

$$\mathcal{L}_{\text{HI}} = \text{CE}_S(\mathcal{N}(\tilde{x}_{u,v}^{C_i \leftarrow c'}), \mathcal{P}_{H^{C_i} \leftarrow H_s^{C_i}}(x_{u,v})) \quad (6.2)$$

Here H^{C_i} are hidden states designated for concept C_i . In essence, we train \mathcal{P} to fully mediate the effect of intervening on C_i in the hidden representation H^{C_i} . The source input s is any input $x_{u',v'}^{C_i=c'}$ that has $C_i = c'$. As \mathcal{P} only receives information about the concept-level intervention $C_i \leftarrow c'$ via the interchange intervention $H^{C_i} \leftarrow H_s^{C_i}$, the model is forced to store all causally relevant information with regard to C_i in the corresponding hidden representation. This process is described in Figure 6.1e.

In the ideal situation, the source input $x_{u',v'}^{C_i=c'}$ and $x_{u,v}$ share the same value only for C_i and differ on all others, so that the counterfactual signal needed for localization is pure. However, we do not insist on this when we sample. In addition, we allow *null effect* pairs in which $x_{u,v}$ and $\tilde{x}_{u,v}^{C_i \leftarrow c'}$ are identical. For additional details on this sampling procedure, see Appendix 6.9.2.

At inference time, approximate counterfactuals are inaccessible, as before. To explain model \mathcal{N} with regard to intervention $C_i \leftarrow c'$, we manipulate the internal states of model \mathcal{P} by intervening on the localized representation H^{C_i} for concept C_i . To achieve this, we sample a source input $x_{u',v'}^{C_i=c'}$ from the train set as any input x that has $C_i = c'$ to derive $H_s^{C_i}$.

Training Objectives We include another distillation objective to predict the same output as \mathcal{N} under conventional circumstances as $\mathcal{L}_{\text{Mimic}} = \text{CE}_S(\mathcal{N}(x_{u,v}), \mathcal{P}(x_{u,v}))$. The overall training objective for our models can be written as $\mathcal{L} = \lambda_1 \mathcal{L}_{\text{Mimic}} + \lambda_2 \mathcal{L}_{\text{Counterfactual}}$ where $\mathcal{L}_{\text{Counterfactual}}$ can be either \mathcal{L}_{IN} or \mathcal{L}_{HI} , and we set λ_1, λ_2 as 1.0 and 3.0 for simplicity.

6.5 Experiment Setup

6.5.1 Causal Estimation-Based Benchmark (CEBaB)

CEBaB [Abraham et al., 2022] is a large benchmark of high-quality, labeled approximate counterfactuals for the task of sentiment analysis on restaurant reviews. The benchmark was created starting from a set of 2,299 original restaurant reviews from OpenTable. For each of these original reviews, approximate counterfactual examples were written by human annotators; the annotators were tasked to edit the original text to reflect a specific intervention, like ‘change the food evaluation from negative to positive’ or ‘change the service evaluation from positive to unknown’. In this way, the original reviews were expanded with approximate counterfactuals to a total of 15,089 texts. The groups of originals and corresponding approximate counterfactuals are partitioned over train, dev, and test sets. The pairs in the development and test set are used to benchmark explanation methods.

Each text in CEBaB was labeled by five crowdworkers with a 5-star sentiment score. In addition, each text was annotated at the concept level for four mediating concepts $\{C_{\text{ambiance}}, C_{\text{food}}, C_{\text{noise}}, \text{ and } C_{\text{service}}\}$, using the three labels again with five crowdworkers annotating each concept-level label. We refer to Appendix 6.9.1 and Abraham et al. 2022 for additional details.

As discussed above (Section 6.4 and Figure 6.1b), we consider two sources of approximate counterfactuals using CEBaB. For human-created counterfactuals, we use the edited restaurant reviews of the train set. For metadata-sampled counterfactuals, we sample factual inputs from the train set that have the desired combination of mediating concepts. Using all the human-created edits leads to 19,684 training pairs of factuals and corresponding approximate counterfactuals. Sampling counterfactuals leads to 74,574 pairs. We use these approximate counterfactuals to train explanation methods. Appendix 6.9.2 provides more information about our pairing process.

6.5.2 Evaluation Metrics

Much of the value of a benchmark like CEBaB derives from its support for directly calculating the Estimated Individual Causal Concept Effect ($\widehat{\text{ICaCE}}_{\mathcal{N}}$) for a model \mathcal{N} given a human-generated approximate counterfactual pair $(x_{u,v}, \tilde{x}_{u,v}^{C_i \leftarrow c'})$:

$$\widehat{\text{ICaCE}}_{\mathcal{N}}(x_{u,v}, \tilde{x}_{u,v}^{C_i \leftarrow c'}) = \mathcal{N}(\tilde{x}_{u,v}^{C_i \leftarrow c'}) - \mathcal{N}(x_{u,v}) \quad (6.3)$$

This is simply the difference between the vectors of output scores for the two examples.

We do not expect to have pairs $(x_{u,v}, \tilde{x}_{u,v}^{C_i \leftarrow c'})$ at inference time, and this is what drives the development of explanation methods $\mathcal{E}_{\mathcal{N}}$ that *estimate* this quantity using only a factual input $x_{u,v}$ and a description of the intervention $C_i \leftarrow c'$. To benchmark such methods, we follow Abraham et al. [2022] in using the

ICaCE-Error:

$$\text{ICaCE-Error}_{\mathcal{N}}^{\mathcal{D}}(\mathcal{E}) = \frac{1}{|\mathcal{D}|} \sum_{(x_{u,v}, \tilde{x}_{u,v}^{C_i \leftarrow c'}) \in \mathcal{D}} \text{Dist}(\widehat{\text{ICaCE}}_{\mathcal{N}}((x_{u,v}, \tilde{x}_{u,v}^{C_i \leftarrow c'})), \mathcal{E}_{\mathcal{N}}(x_{u,v}; C_i \leftarrow c')) \quad (6.4)$$

Here, we assume that \mathcal{D} is a dataset consisting entirely of approximate counterfactual pairs $(x_{u,v}, \tilde{x}_{u,v}^{C_i \leftarrow c'})$. Dist measures the distance between the $\widehat{\text{ICaCE}}_{\mathcal{N}}$ for the model \mathcal{N} and the effect predicted by the explanation method. Abraham et al. [2022] consider three values for Dist : L2, which captures both direction and magnitude; Cosine distance, which captures the direction of effects but not their magnitude; and NormDiff (absolute difference of L2 norms), which captures magnitude but not direction. We report all three metrics.

6.5.3 Baseline Methods

BEST_{CEBaB} We compare our results with the best results obtained on the CEBaB benchmark. Crucially, BEST_{CEBaB} consists aggregated best results from a set of methods including CONEXP [Goyal et al., 2020], TCAV [Kim et al., 2018], ConceptSHAP [Yeh et al., 2020], INLP [Ravfogel et al., 2020], CausaLM [Feder et al., 2021b], and S-Learner [Künzel et al., 2019].

S-Learner Our version of S-Learner [Künzel et al., 2019] learns to mimic the factual behavior of black-box model \mathcal{N} while making the intermediate concepts explicit. Given a factual input, a finetuned model \mathcal{B} is trained to predict concept label for each concept as an aspect-based sentiment classification task. Then, a logistic regression model $\text{LR}_{\mathcal{N}}$ is trained to map these intermediate concept values to the factual output of black-box model \mathcal{N} , under the following objective.

$$\mathcal{L}_{\text{Mimic}}^{\mathcal{S}, \mathcal{B}} = \text{CE}_{\mathcal{S}}(\mathcal{N}(x_{u,v}), \text{LR}_{\mathcal{N}}(\mathcal{B}(x_{u,v}))) \quad (6.5)$$

By intervening on the intermediate predicted concept values at inference-time, we can hope to simulate the counterfactual behavior of \mathcal{N} :

$$\mathcal{E}_{\mathcal{N}}^{\mathcal{S}, \mathcal{B}}(x_{u,v}; C_i \leftarrow c') = \text{LR}_{\mathcal{N}}((\mathcal{B}(x_{u,v}))_{C_i \leftarrow c'}) - \text{LR}_{\mathcal{N}}(\mathcal{B}(x_{u,v})) \quad (6.6)$$

When using S-Learner in conjunction with approximate counterfactual inputs at train-time, we simply add this counterfactual data on top of the observational data that is typically used to train S-Learner.

GPT-3 Large language models such as GPT-3 (175B) have shown extraordinary power in terms of in-context learning [Brown et al., 2020].² We use GPT-3 to generate a new approximate counterfactual at inference time given a factual input

²We use the largest davinci model publicly available at <https://beta.openai.com/playground>.

and a descriptor of the intervention. This generated counterfactual is directly used to estimate the change in model behavior:

$$\mathcal{E}_{\mathcal{N}}^{\text{GPT-3}}(x_{u,v}; C_i \leftarrow c') = \mathcal{N}(\text{GPT-3}(x_{u,v}; C_i \leftarrow c')) - \mathcal{N}(x_{u,v}) \quad (6.7)$$

where $\text{GPT-3}(x_{u,v}; C_i \leftarrow c')$ represents the GPT-3 generated counterfactual edits. We prompt GPT-3 with demonstrations containing approximate counterfactual inputs. Full details on how these prompts are constructed can be found in Appendix 6.9.7.

6.5.4 Causal Proxy Models

We train CPMs for the publicly available models released for CEBaB, fine-tuned as five-way sentiment classifiers on the factual data. This includes four model architectures: `bert-base-uncased` (BERT; Devlin et al. 2019), `RoBERTa-base` (RoBERTa; Liu et al. 2019), GPT-2 (GPT-2; Radford et al. 2019), and LSTM with GloVe (LSTM; Hochreiter and Schmidhuber 1997, Pennington et al. 2014). All Transformer-based models [Vaswani et al., 2017] have 12 Transformer layers. Before training, each CPM model is initialized with the architecture and weights of the black-box model we aim to explain. Thus, the CPMs are rooted in the factual behavior of \mathcal{N} from the start. We include details about our setup in Appendix 6.9.3.

The inference time comparisons for these models are as follows, where \mathcal{P} in Eqn. 6.8 and Eqn. 6.9 refers to the CPM model trained under CPM_{IN} and CPM_{HI} objectives, respectively:

$$\mathcal{E}_{\mathcal{N}}^{\text{CPM}_{\text{IN}}}(x_{u,v}; C_i \leftarrow c') = \mathcal{P}(x_{u,v}; t_{C_i \leftarrow c'}) - \mathcal{N}(x_{u,v}) \quad (6.8)$$

$$\mathcal{E}_{\mathcal{N}}^{\text{CPM}_{\text{HI}}}(x_{u,v}; C_i \leftarrow c') = \mathcal{P}_{H^{C_i} \leftarrow H_s^{C_i}}(x_{u,v}) - \mathcal{N}(x_{u,v}) \quad (6.9)$$

Here, s is a source input with $C_i = c'$, and H^{C_i} is the neural representation associated with C_i which takes value $H_s^{C_i}$ on the source input s . As H^{C_i} , we use the representation of the [CLS] token. Specifically, for BERT we use slices of width 192 taken from the 1st intermediate token of the 10th layer. For RoBERTa, we use the 8th layer instead. For GPT-2, we pick the final token of the 12th layer, again with slice width of 192. For LSTM, we consider slices of the attention-gated sentence embedding with width 64. Appendix 6.9.5 studies the impact of intervention location and size.

Following the guidance on IIT given by Geiger et al. [2022], we train CPM_{HI} with an additional multi-task objective as $\mathcal{L}_{\text{Multi}} = \sum_{C_i \in \mathcal{C}} \text{CE}(\text{MLP}(H_x^{C_i}), c)$ where probe is parameterized by a multilayer perceptron MLP, and $H_x^{C_i}$ is the value of hidden representation for the concept C_i when processing input x with a concept label of c for C_i .

6.6 Results

We first benchmark both CPM variants and our baseline methods on CEBaB. We show that the CPMs achieve state-of-the-art performance, for both types of approxi-

Model	Metric	<i>no counterfactuals</i>		<i>sampled counterfactuals</i>				<i>human-created counterfactuals</i>			
		BEST _{CEBaB}	S-Learner	S-Learner	GPT-3	(ours) CPM _{IN}	(ours) CPM _{HI}	S-Learner	GPT-3	(ours) CPM _{IN}	(ours) CPM _{HI}
BERT	L2	0.74 (.02)	0.74 (.02)	0.74 (.02)	0.71 (.01)	0.63 (.01)	0.60 (.01)	0.73 (.02)	0.45 (.01)	0.45 (.02)	0.45 (.03)
	Cosine	0.59 (.03)	0.63 (.01)	0.63 (.01)	0.51 (.00)	0.46 (.00)	0.45 (.00)	0.60 (.01)	0.36 (.00)	0.35 (.00)	0.36 (.04)
	NormDiff	0.44 (.01)	0.54 (.02)	0.53 (.02)	0.35 (.01)	0.39 (.01)	0.38 (.00)	0.52 (.02)	0.25 (.00)	0.24 (.01)	0.27 (.01)
RoBERTa	L2	0.78 (.01)	0.78 (.01)	0.78 (.00)	0.74 (.01)	0.66 (.01)	0.67 (.02)	0.77 (.00)	0.48 (.01)	0.46 (.01)	0.47 (.03)
	Cosine	0.58 (.01)	0.64 (.01)	0.65 (.01)	0.53 (.01)	0.46 (.00)	0.47 (.00)	0.63 (.01)	0.39 (.00)	0.38 (.01)	0.39 (.03)
	NormDiff	0.45 (.00)	0.59 (.01)	0.58 (.00)	0.36 (.00)	0.42 (.01)	0.45 (.03)	0.56 (.00)	0.28 (.01)	0.26 (.01)	0.29 (.05)
GPT-2	L2	0.60 (.02)	0.60 (.02)	0.61 (.01)	0.65 (.01)	0.55 (.01)	0.51 (.01)	0.61 (.01)	0.43 (.01)	0.41 (.01)	0.41 (.04)
	Cosine	0.59 (.01)	0.59 (.01)	0.59 (.01)	0.52 (.00)	0.47 (.01)	0.46 (.00)	0.59 (.01)	0.40 (.00)	0.37 (.01)	0.39 (.05)
	NormDiff	0.40 (.01)	0.40 (.01)	0.41 (.01)	0.34 (.00)	0.32 (.01)	0.30 (.00)	0.40 (.01)	0.24 (.01)	0.23 (.01)	0.27 (.05)
LSTM	L2	0.73 (.01)	0.73 (.01)	0.73 (.01)	0.76 (.00)	0.66 (.01)	0.64 (.02)	0.72 (.00)	0.49 (.00)	0.52 (.00)	0.54 (.01)
	Cosine	0.64 (.01)	0.64 (.01)	0.64 (.01)	0.57 (.01)	0.50 (.00)	0.50 (.01)	0.63 (.01)	0.44 (.00)	0.45 (.01)	0.46 (.00)
	NormDiff	0.50 (.01)	0.53 (.01)	0.53 (.00)	0.41 (.00)	0.42 (.00)	0.41 (.01)	0.54 (.00)	0.30 (.00)	0.34 (.01)	0.36 (.00)

Table 6.1: CEBaB scores measured in three different metrics on the test set for four different model architectures as a five-class sentiment classification task. **Lower is better**. Results averaged over three distinct seeds, standard deviations in parentheses. The metrics are described in Section 6.5. Best averaged result is bolded (including ties) per approximate counterfactual creation strategy.

mate counterfactuals used during training (Section 6.6.1). Given the good factual performance achieved by CPMs, we subsequently investigate whether CPMs can be deployed both as predictor and explanation method at the same time (Section 6.6.2) and find that they can. Finally, we show that the localized representations of CPM_{HI} give rise to concept-aware feature attributions (Section 6.6.3). Our supplementary materials report on detailed ablation studies and explore the potential of our methods for model debiasing.

6.6.1 CEBaB Performance

Table 6.1 presents our main results. The results are grouped per approximate counterfactual type used during training. Both CPM_{IN} and CPM_{HI} beat BEST_{CEBaB} in every evaluation setting by a large margin, establishing state-of-the-art explanation performance. Interestingly, CPM_{HI} seems to slightly outperform CPM_{IN} using *sampled* approximate counterfactuals, while slightly underperforming CPM_{IN} on *human-created* approximate counterfactuals. Appendix 6.9.6 reports on ablation studies that indicate that, for CPM_{HI}, this state-of-the-art performance is primarily driven by the role of IIT in localizing concepts.

S-Learner, one of the best individual explainers from the original CEBaB paper [Abraham et al., 2022], shows only a marginal improvement when naively incorporating *sampled* and *human-created* counterfactuals during training over using *no counterfactuals*. This indicates that the large performance gains achieved by our CPMs over previous explainers are most likely due to the explicit use of a counterfactual training signal, and not primarily due to the addition of extra (counterfactual) data.

GPT-3 occasionally performs on-par with our CPMs, generally only slightly underperforming our best explainer on *human-created counterfactuals*, while being significantly worse on *sampled counterfactuals*. While the GPT-3 explainer also

explicitly uses approximate counterfactual data, the results indicate that our proposed counterfactual mimic objectives give better results. The better performance of CPMs when considering *sampled counterfactuals* over GPT-3 shows that our approach is more robust to the quality of the approximate counterfactuals used. While the GPT-3 explainer is easy to set up (no training required), it might not be suitable for some explanation applications regardless of performance, due to the latency and cost involved in querying the GPT-3 API.

Across the board, explainers trained with *human-created* counterfactuals are better than those trained with *sampled* counterfactuals. This shows that the performance of explanation methods depends on the quality of the approximate counterfactual training data. While human counterfactuals give excellent performance, they may be expensive to create. Sampled counterfactuals are cheaper if the relevant metadata is available. Thus, under budgetary constraints, sampled counterfactuals may be more efficient.

Finally, CPM_{IN} is conceptually the simpler of the two CPM variants. However, we discuss in Section 6.6.3 how the localized representations of CPM_{HI} lead to additional explainability benefits.

6.6.2 Self-Explanation with CPM

As outlined in Section 6.4, CPMs learn to mimic both the factual and counterfactual behavior of the black-box models they are explaining. We show in Table 6.2 that our CPMs achieve a factual Macro-F1 score comparable to the black-box finetuned models.

We investigate if we can simply replace the black-box model with our CPM and use the CPM both as factual predictor and counterfactual explainer. To answer this questions, we measure the self-explanation performance of CPMs by simply replacing the black-box model \mathcal{N} in Eqn. 6.4 with our factual CPM predictions at inference time.

Table 6.3 reports these results. We find that both CPM_{IN} and CPM_{HI} achieve better self-explanation performance compared to providing explanations for another black-box model. Furthermore, CPM_{HI} provides better self-explanation than CPM_{IN}, suggesting our interchange intervention procedure leads the model to localize concept-based information in hidden representations. This shows that CPMs may be viable as replacements for their black-box counterpart, since they provide similar task performance while providing faithful counterfactual explanations of both the black-box model and themselves.

6.6.3 Concept-Aware Feature Attribution with CPM_{HI}

We have shown that CPM_{HI} provides trustworthy explanations (Section 6.6.1). We now investigate whether CPM_{HI} learns representations that mediate the effects of different concepts. We adapt Integrated Gradients (IG; Sundararajan et al. 2017) to provide concept-aware feature attributions, by only considering gradients flowing

Model	Black-box	sampled counterfactuals		human-created counterfactuals	
		CPM _{IN}	CPM _{HI}	CPM _{IN}	CPM _{HI}
BERT	0.70 (.01)	0.70 (.00)	0.67 (.02)	0.70 (.01)	0.69 (.01)
RoBERTa	0.70 (.00)	0.70 (.00)	0.69 (.01)	0.71 (.01)	0.71 (.00)
GPT-2	0.65 (.00)	0.65 (.00)	0.67 (.01)	0.66 (.01)	0.68 (.00)
LSTM	0.60 (.01)	0.60 (.01)	0.56 (.00)	0.54 (.00)	0.59 (.01)

Table 6.2: Task performance measured as Macro-F1 score on the test set. Results averaged over three distinct seeds; standard deviations in parentheses.

Model	Metric	sampled counterfactuals		human-created counterfactuals	
		CPM _{IN}	CPM _{HI}	CPM _{IN}	CPM _{HI}
BERT	L2	0.63 (.01)	0.52 (.04)	0.42 (.02)	0.38 (.03)
	Cosine	0.46 (.00)	0.45 (.01)	0.34 (.02)	0.30 (.06)
	NormDiff	0.39 (.01)	0.33 (.02)	0.23 (.01)	0.22 (.05)
RoBERTa	L2	0.66 (.01)	0.63 (.04)	0.40 (.01)	0.37 (.04)
	Cosine	0.46 (.00)	0.48 (.01)	0.33 (.01)	0.29 (.04)
	NormDiff	0.42 (.01)	0.42 (.05)	0.21 (.01)	0.23 (.05)
GPT-2	L2	0.55 (.01)	0.41 (.03)	0.38 (.01)	0.36 (.04)
	Cosine	0.47 (.01)	0.39 (.02)	0.37 (.01)	0.35 (.05)
	NormDiff	0.32 (.01)	0.25 (.02)	0.22 (.01)	0.24 (.05)
LSTM	L2	0.66 (.01)	0.41 (.01)	0.46 (.00)	0.42 (.01)
	Cosine	0.50 (.00)	0.42 (.02)	0.50 (.02)	0.40 (.01)
	NormDiff	0.42 (.00)	0.25 (.00)	0.31 (.00)	0.28 (.02)

Table 6.3: Self-explanation CEBaB scores measured in three different metrics on the test set for four different model architectures as a five-class sentiment classification task. **Lower is better**. Results averaged over three distinct seeds, standard deviations in parentheses.

through the hidden representation associated with a given concept. We formalize this version of IG in Appendix 6.9.8.

In Table 6.4, we compare concept-aware feature attributions for two variants of CPM_{HI} (IIT and Multi-task) and the original black-box (Finetuned) model. For IIT we remove the multi-task objective $\mathcal{L}_{\text{Multi}}$ during training and for Multi-task we remove the the interchange intervention objective \mathcal{L}_{HI} . This helps isolate the individual effects of both losses on concept localization. All three models predict a neutral final sentiment score for the considered input, but they show vastly different feature attributions. Only IIT reliably highlights words that are semantically related to each concept. For instance, when we restrict the gradients to flow only through the intervention site of the *noise* concept, “loud” is the word highlighted the most that contributes negatively. When we consider the *service* concept, words like “friendly” and “waiter” are highlighted the most as contributing positively. These contrasts are missing for representations of the Multi-task and Finetuned models. Only the IIT training paradigm pushes the model to learn causally localized representations. For the *service* concept, we notice that the IIT model wrongfully attributes “delicious”. This could be useful for debugging purposes and could be used to highlight potential failure modes of the model.

Model	Predicted Concept	Score	Word Importance																							
Black-box	<i>neutral</i>	ambiance +0.03	[CLS]	the	music	was	too	loud	.	and	the	decorations	were	taste	##less	.	but	they	had	friendly	waiter	##s	and	delicious	pasta	[SEP]
		food +0.11	[CLS]	the	music	was	too	loud	.	and	the	decorations	were	taste	##less	.	but	they	had	friendly	waiter	##s	and	delicious	pasta	[SEP]
		noise +0.04	[CLS]	the	music	was	too	loud	.	and	the	decorations	were	taste	##less	.	but	they	had	friendly	waiter	##s	and	delicious	pasta	[SEP]
		service +0.26	[CLS]	the	music	was	too	loud	.	and	the	decorations	were	taste	##less	.	but	they	had	friendly	waiter	##s	and	delicious	pasta	[SEP]
Multi-task	<i>neutral</i>	ambiance +0.25	[CLS]	the	music	was	too	loud	.	and	the	decorations	were	taste	##less	.	but	they	had	friendly	waiter	##s	and	delicious	pasta	[SEP]
		food +0.23	[CLS]	the	music	was	too	loud	.	and	the	decorations	were	taste	##less	.	but	they	had	friendly	waiter	##s	and	delicious	pasta	[SEP]
		noise +0.31	[CLS]	the	music	was	too	loud	.	and	the	decorations	were	taste	##less	.	but	they	had	friendly	waiter	##s	and	delicious	pasta	[SEP]
		service +0.16	[CLS]	the	music	was	too	loud	.	and	the	decorations	were	taste	##less	.	but	they	had	friendly	waiter	##s	and	delicious	pasta	[SEP]
IIT	<i>neutral</i>	ambiance -0.24	[CLS]	the	music	was	too	loud	.	and	the	decorations	were	taste	##less	.	but	they	had	friendly	waiter	##s	and	delicious	pasta	[SEP]
		food +1.11	[CLS]	the	music	was	too	loud	.	and	the	decorations	were	taste	##less	.	but	they	had	friendly	waiter	##s	and	delicious	pasta	[SEP]
		noise -0.98	[CLS]	the	music	was	too	loud	.	and	the	decorations	were	taste	##less	.	but	they	had	friendly	waiter	##s	and	delicious	pasta	[SEP]
		service +1.16	[CLS]	the	music	was	too	loud	.	and	the	decorations	were	taste	##less	.	but	they	had	friendly	waiter	##s	and	delicious	pasta	[SEP]

Table 6.4: Visualizations of word importance scores using Integrated Gradient (IG) by restricting gradient flow through the corresponding intervention site of the targeted concept. Our target class pools *positive* and *very positive*. Individual word importance is the sum of neuron-level importance scores for each input, normalized to $[-1, +1]$. -1 means the word contributes the most negatively to predicting the target class (red); $+1$ means the word contributes the most positively (green).

6.7 Conclusion

We explored the use of approximate counterfactual training data to build more robust causal explanation methods. We introduced Causal Proxy Models (CPMs), which learn to mimic both the *factual* and *counterfactual* behaviors of a black-box model \mathcal{N} . Using CEBaB, a benchmark for causal concept-based explanation methods, we demonstrated that both versions of our technique (CPM_{IN} and CPM_{HI}) significantly outperform previous explanation methods without demanding the full causal graph associated with the dataset. Interestingly, we find that our GPT-3 based explanation method performs on-par with our best CPM model in some settings.

Our results suggest that CPMs can be more than just explanation methods. They achieve factual performance on par with the model they aim to explain, and they can explain their own behavior. This paves the way to using them as deployed models that both perform tasks and offer explanations. In addition, the causally localized representations of our CPM_{HI} variant are very intuitive, as revealed by our concept-aware feature attribution technique. We believe that causal localization techniques could play a vital role in further model explanation efforts.

6.8 Acknowledgement

This research is supported in part by a grant from Meta AI. Karel D’Oosterlinck was supported through a doctoral fellowship from the Special Research Fund (BOF) of Ghent University.

6.9 Appendix

6.9.1 CEBaB Dataset Statistics

Table 6.5 shows dataset statistics of CEBaB. The variants of CEBaB we consider only impact the train split. The top panel shows the number of observational samples and edits introduced in the CEBaB paper. The bottom panel shows our *paired* versions, where we create approximate counterfactual pairs. We explore two variants of approximate counterfactuals: *human*-created and *sampled* counterfactuals (Section 6.5.1). The *human* setting considers all pairs made possible by using *all* data. The *sampling* setting considers pairs sampled from only the *observational* data, as discussed in Section 6.9.2.

Dataset	# train	# dev	# test
CEBaB (<i>observational</i>)	1,755	1,673	1,689
CEBaB (<i>all</i>)	11,728	1,673	1,689
CEBaB (<i>paired, human</i>)	19,684	3,898	3,958
CEBaB (<i>paired, sampling</i>)	74,574	3,898	3,958

Table 6.5: Dataset statistics.

6.9.2 Types of Approximate Counterfactual Pairs

Our approximate counterfactual training data comes in paired sentences of (*original sentence, approximate counterfactual sentence*). The approximate counterfactuals differs from their original counterparts in only one concept value. We consider approximate counterfactual pairs to be symmetric: we use both (*original sentence, approximate counterfactual sentence*) and (*approximate counterfactual sentence, original sentence*) as training pairs.

Human-created Counterfactuals CEBaB contains multiple counterfactual sentences for each original review. To achieve this, the dataset creators asked annotators to edit the original sentence to achieve a specified goal (e.g., ‘change the evaluation of the restaurant’s food to negative’). These originals and corresponding edits form our *human* pairs.

Metadata-sampled Counterfactuals Human-created counterfactuals are not always available. With CEBaB, we simulate a second type of approximate counterfactuals by using metadata-guided heuristics: for a given *original sentence*, we sample a counterfactual from the train set by matching concept labels while allowing only one label to be changed.

During training, we also consider *null effect pairs* in our *sampling* setup. These pairs resemble cases where our approximate counterfactual sentence is identical to the original sentence. When training our models on these pairs, we expect our models to predict the same counterfactual and factual output.

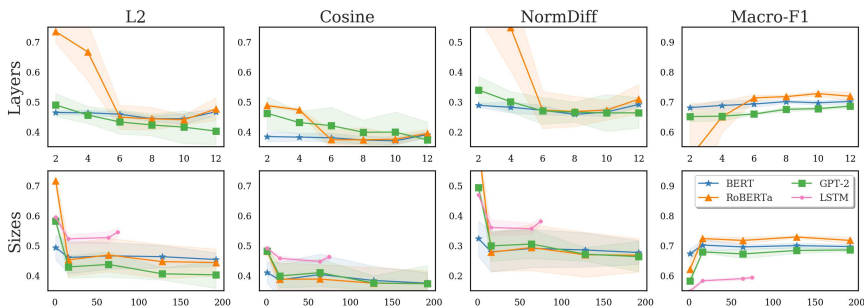


Figure 6.2: CEBaB scores for different intervention site locations and sizes for CPM_{HI} . The scores are measured in three different metrics on the test set for four different model architectures as a five-class sentiment classification task. Results averaged over three distinct seeds. Task performance as Macro-F1 score is reported when applicable. Shaded areas outline \pm SD.

6.9.3 Training Regimes

CPM_{IN} To train CPM_{IN} , we use the same model architecture as \mathcal{N} , and initialize it with the model weights using weights from \mathcal{N} . The maximum number of training epochs is set to 30 with a learning rate of $5e^{-5}$ and an effective batch size of 128. The learning rate linearly decays to 0 over the 30 training epochs. We employ an early stopping strategy for $\text{COS}_{\text{ICaCE}}$ over the dev set for an interval of 50 steps with early stopping patience set to 20. We set the max sequence length to 128 and the dropout rate to 0.1. We take a weighted sum of two objectives as the loss term for training CPM_{HI} . Specifically, we use $[w_{\text{Mimic}}, w_{\text{IN}}] = [1.0, 3.0]$. For the smoothed cross-entropy loss, we use a temperature of 2.0.

CPM_{HI} To train CPM_{HI} , we use the same model architecture as \mathcal{N} , and initialize it with the model weights using weights from \mathcal{N} . The maximum number of training epochs is set to 30 with a learning rate of $8e^{-5}$ and an effective batch size of 256. We use a higher learning rate of 0.001 for the LSTM model as it enables quicker convergence. The learning rate linearly decays to 0 over the 30 training epochs. We employ an early stopping strategy for $\text{COS}_{\text{ICaCE}}$ over the dev set for an interval of 10 steps with early stopping patience set to 20. We set the max sequence length to 128 and the dropout rate to 0.1. We take a weighted sum of three objectives as the loss term for training CPM_{HI} . Specifically, we use $[w_{\text{Mimic}}, w_{\text{Multi}}, w_{\text{HI}}] = [1.0, 1.0, 3.0]$. In Appendix 6.9.6, we conduct a set of ablation studies to isolate the individual contributions from each objective. For the smoothed cross-entropy loss, we use a temperature of 2.0.

Our models are all implemented in `PyTorch` [Paszke et al., 2019] and using the `HuggingFace` library [Wolf et al., 2019]. All of our results are aggregated over three distinct random seeds. To foster reproducibility, we will release our code repository and model artifacts to the public.

6.9.4 Additional Baseline Results

Table 6.6 shows baselines adapted from Abraham et al. [2022], which contains the present state-of-the-art explanation methods for the CEBaB benchmark. We report the best scores across these explanation methods in Table 6.1. These baselines are trained without using counterfactual data. Thus, we build additional baselines that use counterfactual data as shown in Table 6.7. S-Learner is selected as the best performing models and included in Table 6.1 for comparisons. The equations for the additional baselines are as follows:

$$\mathcal{E}_{\mathcal{N}}^{\text{approx}}(x_{u,v}; C_i \leftarrow c') = \mathcal{N}(s^{\text{approx}}) - \mathcal{N}(x_{u,v}) \quad (6.10)$$

$$\mathcal{E}_{\mathcal{N}}^{\text{random}}(x_{u,v}; C_i \leftarrow c') = \mathcal{N}(s^{\text{random}}) - \mathcal{N}(x_{u,v}) \quad (6.11)$$

$$\mathcal{E}_{\mathcal{N}}^{\text{CaCE}}(C_i \leftarrow c') = \frac{1}{|\mathcal{D}^{C_i \leftarrow c'}|} \sum_{(x_{u,v}, \tilde{x}_{u,v}^{C_i \leftarrow c'}) \in \mathcal{D}^{C_i \leftarrow c'}} \left(\mathcal{N}(\tilde{x}_{u,v}^{C_i \leftarrow c'}) - \mathcal{N}(x_{u,v}) \right) \quad (6.12)$$

$$\mathcal{E}^{\text{ATE}}(C_i \leftarrow c') = \frac{1}{|\mathcal{D}^{C_i \leftarrow c'}|} \sum_{(x_{u,v}, \tilde{x}_{u,v}^{C_i \leftarrow c'}) \in \mathcal{D}^{C_i \leftarrow c'}} \left(f(\tilde{x}_{u,v}^{C_i \leftarrow c'}) - f(x_{u,v}) \right) \quad (6.13)$$

where s^{random} is a randomly sampled training input, s^{approx} is a training input sampled to match the concept-level labels of the true counterfactual under intervention $C_i \leftarrow c'$, $\mathcal{D}^{C_i \leftarrow c'}$ is the set of all approximate counterfactual training pairs that represent a $C_i \leftarrow c'$ intervention, and f is a look-up function that returns the ground-truth label associated with an input.

The signatures of \mathcal{E}^{ATE} and $\mathcal{E}_{\mathcal{N}}^{\text{CaCE}}$ reflect that they are independent of the specific factual input $x_{u,v}$ considered. Furthermore, \mathcal{E}^{ATE} is independent of \mathcal{N} given that this explainer only uses ground-truth training labels to estimate causal effects.

Additionally, we consider X-Learner, a variant of S-Learner [Künzel et al., 2019]. Our X-Learner consists of three steps. First, we cluster examples into groups by their concept and predicted concept label pairs (e.g., *select all examples with food being positive*)³. For each group, we fit logistic regression model $\hat{\mu}_{(C_i, c)}$ to predict the factual output of black-box model \mathcal{N} using concept labels for each example except for labels for C_i . Next, we use the models from the first step to build training sets for our individual treatment effect (ITE) estimators. To achieve this, we calculate ITE for each example as,

$$\hat{\mathcal{D}}_{u,v}^{C_i: c \leftarrow c'} = \hat{\mu}_{(C_i, c')}(\mathcal{B}(x_{u,v}^{C_i=c}))' - \mathcal{N}(x_{u,v}^{C_i=c}) \quad (6.14)$$

where $\mathcal{B}(x_{u,v}^{C_i=c})'$ excludes the concept label for concept C_i . It measures the ITE for $x_{u,v}$ when we change the concept label of C_i from c to c' . We aggregate $\hat{\mathcal{D}}_{u,v}^{C_i: c \leftarrow c'}$ over examples based on their editing concepts and concept labels. Next, we fit a

³We use the finetuned concept-level sentiment analysis models \mathcal{B} released by Abraham et al. [2022] for concept label prediction, which is identical to the ones used in S-Learner in Section 6.5.3.

Model	Metric	Approx [†]	S-Learner [‡]	INLP [§]
BERT	L2	0.81 (.01)	0.74 (.02)	0.80 (.02)
	Cosine	0.61 (.01)	0.63 (.01)	0.59 (.03)
	NormDiff	0.44 (.01)	0.54 (.02)	0.73 (.02)
RoBERTa	L2	0.83 (.01)	0.78 (.01)	0.84 (.01)
	Cosine	0.60 (.01)	0.64 (.01)	0.58 (.01)
	NormDiff	0.45 (.00)	0.59 (.01)	0.81 (.01)
GPT-2	L2	0.72 (.02)	0.60 (.02)	0.72 (.01)
	Cosine	0.59 (.01)	0.59 (.01)	1.00 (.00)
	NormDiff	0.41 (.01)	0.40 (.01)	0.58 (.03)
LSTM	L2	0.86 (.01)	0.73 (.01)	0.79 (.01)
	Cosine	0.64 (.01)	0.64 (.01)	0.74 (.02)
	NormDiff	0.50 (.01)	0.53 (.01)	0.60 (.01)

Table 6.6: CEBaB scores measured in three different metrics on the test set for four different model architectures as a five-class sentiment classification task. Results are adapted from Abraham et al. [2022]. **Lower is better**; standard deviations over 5 distinct seeds in parentheses. Results are aggregated over all aspects and all directional concept label changes. Details about these evaluation metrics can be found in Section 6.5. Results are based on [†]Abraham et al. [2022], [‡]Künzel et al. [2019], and [§]Ravfogel et al. [2020].

Model	Metric	no counterfactuals		sampled counterfactuals				human-created counterfactuals				
		X-Learner	ATE- Approx	CaCE- Explainer	Explainer	Random	X-Learner	ATE- Approx	CaCE- Explainer	Explainer	Random	X-Learner
BERT	L2	0.78 (.02)	0.81 (.00)	0.81 (.02)	0.81 (.02)	0.84 (.02)	0.78 (.02)	0.79 (.02)	0.81 (.02)	0.80 (.02)	0.84 (.01)	0.75 (.02)
	Cosine	0.68 (.01)	0.60 (.00)	0.72 (.01)	0.72 (.01)	0.53 (.00)	0.68 (.01)	0.56 (.00)	0.69 (.01)	0.69 (.01)	0.53 (.00)	0.64 (.01)
	NormDiff	0.53 (.03)	0.44 (.01)	0.62 (.02)	0.62 (.02)	0.55 (.02)	0.53 (.03)	0.43 (.01)	0.62 (.02)	0.64 (.02)	0.54 (.02)	0.54 (.03)
RoBERTa	L2	0.82 (.00)	0.83 (.00)	0.85 (.00)	0.85 (.00)	0.87 (.00)	0.82 (.00)	0.81 (.01)	0.85 (.00)	0.84 (.00)	0.87 (.00)	0.79 (.00)
	Cosine	0.70 (.02)	0.61 (.01)	0.73 (.00)	0.73 (.01)	0.53 (.00)	0.70 (.02)	0.57 (.01)	0.70 (.00)	0.70 (.00)	0.53 (.00)	0.67 (.02)
	NormDiff	0.57 (.00)	0.46 (.01)	0.67 (.00)	0.67 (.00)	0.58 (.00)	0.57 (.00)	0.44 (.01)	0.67 (.00)	0.68 (.00)	0.59 (.00)	0.58 (.00)
GPT-2	L2	0.65 (.01)	0.72 (.02)	0.69 (.01)	0.68 (.01)	0.76 (.00)	0.65 (.01)	0.72 (.01)	0.68 (.01)	0.68 (.01)	0.76 (.00)	0.63 (.01)
	Cosine	0.64 (.01)	0.59 (.00)	0.67 (.00)	0.67 (.00)	0.56 (.00)	0.64 (.01)	0.57 (.00)	0.66 (.00)	0.65 (.00)	0.56 (.00)	0.62 (.01)
	NormDiff	0.41 (.00)	0.40 (.01)	0.48 (.01)	0.49 (.01)	0.47 (.00)	0.41 (.00)	0.40 (.00)	0.49 (.01)	0.50 (.01)	0.47 (.01)	0.42 (.01)
LSTM	L2	0.77 (.01)	0.87 (.00)	0.78 (.00)	0.78 (.00)	0.85 (.00)	0.77 (.01)	0.85 (.01)	0.78 (.00)	0.76 (.00)	0.84 (.00)	0.74 (.01)
	Cosine	0.69 (.01)	0.65 (.00)	0.71 (.00)	0.71 (.00)	0.57 (.00)	0.69 (.01)	0.61 (.00)	0.69 (.00)	0.68 (.00)	0.56 (.00)	0.67 (.01)
	NormDiff	0.52 (.01)	0.50 (.00)	0.59 (.00)	0.59 (.00)	0.55 (.00)	0.52 (.01)	0.49 (.00)	0.59 (.00)	0.61 (.00)	0.55 (.00)	0.55 (.01)

Table 6.7: CEBaB scores for additional baselines we considered. CEBaB scores are measured in three different metrics on the test set for four different model architectures as a five-class sentiment classification task. **Lower is better**. Results averaged over three distinct seeds, standard deviations in parentheses. Details about these evaluation metrics can be found in Section 6.5.

set of linear regression models as $\tau_{C_i:c \leftarrow c'}$ to predict ITE for changing the concept labels for C_i given concept labels of an example except for labels for C_i . Lastly, we use $\tau_{C_i:c \leftarrow c'}$ to predict counterfactual output changes as,

$$\begin{aligned} \mathcal{E}_{\mathcal{N}}^{\text{X-Learner}}(x_{u,v}; C_i \leftarrow c') = \\ p \cdot \tau_{C_i:c \leftarrow c'}(\mathcal{B}(x_{u,v}^{C_i=c}))' + (1-p) \cdot \tau_{C_i:c' \leftarrow c}(\mathcal{B}(x_{u,v}^{C_i=c}))' \end{aligned} \quad (6.15)$$

where p is the propensity score which is calculated using \mathcal{B} as the probability of C_i taking concept label c' for an input example $x_{u,v}$ by considering two potential concept labels c and c' .

6.9.5 Intervention Site Location and Size

Previous work shows that neurons in different layers and groups can encode different high-level concepts [Vig et al., 2020, Koh et al., 2020]. CPM_{HI} pushes concept-related information to localize at the targeted intervention site (the aligned neural representations for each concept). In this section, we investigate how the location and the size of the intervention site impact CPM_{HI} performance. We use the optimal location and size found in this study for other results presented in this paper.

Location For Transformer-based models, we vary the location of the intervention site by intervening on the “[CLS]” token embedding layer l . Specifically, we set $l = \{2, 4, 6, 8, 10, 12\}$. We skip this experiment for non-Transformer-based model (i.e., LSTM) since it only contains a single sentence embedding.

As shown in the top panel of Figure 6.2, intervention location significantly affects CPM_{HI} performance. Our results show that layer 10 for BERT, layer 8 for ROBERTa, and layer 12 for GPT-2 lead to the best performance. This suggests layers have different efficacy in terms of information localization. Our results also show that intervening with deeper layers tends to provide better performance. However, for both BERT and ROBERTa, intervening on the last layer results in a slightly worse performance compared to earlier layers. This suggests that leaving Transformer blocks after the intervention site helps localized information to be processed by the neural network.

Size For Transformer-based models, we change the size of the intervention site d_c for each concept. Specifically, we set $d_c = \{1, 16, 64, 128, 192\}$. For instance when $d_c = 1$, we use a single dimension of the “[CLS]” token embedding to represent each concept, starting from the first dimension of the vector. For our non-Transformer-based model (LSTM), we intervene on the attention-gated sentence embedding whose dimension size is set to 300. Accordingly, we set $d_c = \{1, 16, 64, 75\}$.

As shown in Figure 6.2, larger intervention sites lead to better performance for all Transformer-based models. For LSTM, we find that the optimal size is the second largest one instead. On the other hand, our results suggest that the performance gain from the increase of size diminishes as we increase the size for all model architectures.

6.9.6 Ablation Study of CPM_{HI}

Geiger et al. [2022] show that training with a multi-task objective helps IIT to improve generalizability. In this experiment, we aim to investigate whether the multi-task objective we added for CPM_{HI} plays an important role in achieving good performance. Specifically, we conduct two ablation studies: removing the multi-task objective by setting $w_{\text{Multi}} = 0.0$, and removing the IIT objective by setting $w_{\text{HI}} = 0.0$.

Table 6.8 shows our results, which demonstrate that the IIT objective is the main factor that drives CPM_{HI} performance. Our results also suggest that the multi-task objective brings relatively small but consistent performance gains. Overall, our

Model	Ablation	L2	Cosine	NormDiff	Macro-F1
BERT	CPM_{HI}	0.45 (.02)	0.36 (.03)	0.27 (.04)	0.69 (.01)
	- $\mathcal{L}_{\text{Multi}}$	0.47 (.04)	0.38 (.04)	0.30 (.07)	0.69 (.01)
	- \mathcal{L}_{HI}	0.79 (.02)	0.60 (.03)	0.64 (.02)	0.60 (.08)
	+ <i>random init</i>	0.81 (.02)	0.52 (.00)	0.55 (.02)	0.08 (.02)
	+ <i>no training</i>	0.80 (.02)	0.86 (.04)	0.76 (.02)	0.70 (.01)
RoBERTa	CPM_{HI}	0.47 (.03)	0.39 (.03)	0.29 (.05)	0.71 (.00)
	- $\mathcal{L}_{\text{Multi}}$	0.49 (.05)	0.41 (.05)	0.32 (.06)	0.70 (.00)
	- \mathcal{L}_{HI}	0.81 (.00)	0.53 (.02)	0.63 (.01)	0.39 (.06)
	+ <i>random init</i>	0.85 (.00)	0.51 (.00)	0.59 (.01)	0.06 (.00)
	+ <i>no training</i>	0.84 (.01)	0.93 (.05)	0.83 (.00)	0.70 (.00)
GPT-2	CPM_{HI}	0.41 (.04)	0.39 (.05)	0.27 (.05)	0.68 (.00)
	- $\mathcal{L}_{\text{Multi}}$	0.43 (.03)	0.41 (.05)	0.29 (.04)	0.67 (.00)
	- \mathcal{L}_{HI}	0.66 (.01)	0.58 (.04)	0.49 (.01)	0.58 (.04)
	+ <i>random init</i>	0.73 (.00)	0.54 (.00)	0.47 (.01)	0.16 (.00)
	+ <i>no training</i>	0.65 (.00)	0.61 (.00)	0.57 (.02)	0.65 (.00)
LSTM	CPM_{HI}	0.54 (.01)	0.46 (.01)	0.36 (.00)	0.59 (.01)
	- $\mathcal{L}_{\text{Multi}}$	0.56 (.02)	0.47 (.02)	0.41 (.02)	0.59 (.01)
	- \mathcal{L}_{HI}	0.73 (.00)	0.64 (.02)	0.59 (.00)	0.59 (.01)
	+ <i>random init</i>	0.82 (.00)	0.55 (.00)	0.55 (.00)	0.13 (.04)
	+ <i>no training</i>	0.73 (.01)	0.74 (.00)	0.59 (.01)	0.60 (.01)

Table 6.8: Ablation study of our CPM_{HI} method trained with *human* approximate counterfactual strategy. CEBAb scores measured in three different metrics on the test set for four different model architectures as a five-class sentiment classification task. **Lower is better**. Results averaged over three distinct seeds, standard deviations in parentheses.

Model	Metric	<i>sampled counterfactuals</i>			<i>human-created counterfactuals</i>		
		CPM _{HI}	Random Source	Probe-based Source	CPM _{HI}	Random Source	Probe-based Source
BERT	L2	0.60 (.01)	0.74 (.03)	0.61 (.01)	0.45 (.03)	0.70 (.03)	0.43 (.02)
	Cosine	0.45 (.00)	0.53 (.01)	0.45 (.00)	0.36 (.04)	0.59 (.04)	0.35 (.01)
	NormDiff	0.38 (.00)	0.54 (.02)	0.39 (.01)	0.27 (.01)	0.53 (.01)	0.25 (.02)
RoBERTa	L2	0.67 (.02)	0.79 (.01)	0.66 (.02)	0.47 (.03)	0.72 (.01)	0.44 (.01)
	Cosine	0.47 (.00)	0.52 (.01)	0.46 (.01)	0.39 (.03)	0.57 (.03)	0.37 (.01)
	NormDiff	0.45 (.03)	0.59 (.00)	0.44 (.03)	0.29 (.05)	0.55 (.01)	0.25 (.01)
GPT-2	L2	0.51 (.01)	0.65 (.02)	0.51 (.02)	0.41 (.04)	0.58 (.03)	0.39 (.02)
	Cosine	0.46 (.00)	0.55 (.01)	0.46 (.01)	0.39 (.05)	0.56 (.02)	0.37 (.01)
	NormDiff	0.30 (.00)	0.46 (.01)	0.31 (.01)	0.27 (.05)	0.44 (.01)	0.25 (.01)
LSTM	L2	0.64 (.02)	0.76 (.01)	0.65 (.02)	0.54 (.01)	0.69 (.03)	0.55 (.00)
	Cosine	0.50 (.01)	0.57 (.01)	0.50 (.01)	0.46 (.00)	0.58 (.01)	0.46 (.01)
	NormDiff	0.41 (.01)	0.54 (.01)	0.41 (.02)	0.36 (.00)	0.52 (.00)	0.38 (.01)

Table 6.9: Ablation study of our CPM_{HI} method for different *source* input s sampling strategies at inference time. CEBAb scores measured in three different metrics on the test set for four different model architectures as a five-class sentiment classification task. **Lower is better**. Results averaged over three distinct seeds, standard deviations in parentheses.

findings corroborate those of Geiger et al. [2022] and provide concrete evidence that the combination of two objectives always results in the best-performing explanation methods across all model architectures.

Additionally, we explore two baselines for CPM_{HI} . Firstly, we randomly initialize the weights of CPM_{HI} . Secondly, we take the original black-box model as our CPM_{HI} . Compared to the results in Table 6.1, these two baselines fail catastrophically, suggesting the importance of our IIT paradigm.

As mentioned in Section 6.4, we sample a source input $x_{u',v'}^{C_i=c'}$ from the train set as any input x that has $C_i = c'$ to estimate the counterfactual output. Furthermore, we explore two additional sampling strategies. First, we create a baseline where we randomly sample a source input from the train without any concept label matching. Second, we sample a source input from the train set using the predicted concept label of our multi-task probe, instead of the true concept label from the dataset.

As shown in Table 6.9, the quality of our source inputs impact our performance significantly. For instance, when sampling source input at random, CPM_{HI} fails catastrophically for all evaluation metrics. On the other hand, when we sampling source based on the predicted labels using the multi-task probe, CPM_{HI} maintains its performance.

6.9.7 GPT-3 Generation Process

We use the 175B parameter `davinci` GPT-3 model [Brown et al., 2020] as a few-shot learner to generate approximate counterfactual data. Let $x_{u,v}$ be a review text with an original value c for the mediating concept C_i and an overall review sentiment y (e.g., a restaurant review which is *negative* about the *service*, and felt *neutral* about their overall dining experience), and let c' be the target value of C_i , for which we would like to create a counterfactual review (e.g., change the text to become *positive* about the mediating concept *service*). In order to use GPT-3 as an n -shot learner, we sample $n = 6$ approximate counterfactual pairs $(x_{u',v'}, \tilde{x}_{u',v'}^{C_i \leftarrow c'})$, where $x_{u',v'}$ shares with $x_{u,v}$ the same value c for C_i and the same overall sentiment, and the counterfactual review $\tilde{x}_{u',v'}^{C_i \leftarrow c'}$ has the target value c' for C_i . We prompt the model with these pairs, and we also include the original review $x_{u,v}$. We then collect the text completed by GPT-3 as the GPT-3 counterfactual review. An example for this n -shot prompt and completion is in Figure 6.3. In addition, we also prompt GPT-3 with pairs of original reviews and metadata-sampled counterfactuals, and generate another set of GPT-3 counterfactual review for comparison. We sample $n = 4$ approximate counterfactual pairs in this case. An example of metadata-sampled counterfactual generation with GPT-3 can be seen in Figure 6.4.

For each few-shot learning prompt, we insert an initial string of the form of “Make the following restaurant reviews include c' mentions of C_i .”, where c' is expressed as one of {“POSITIVE”, “NEGATIVE”, “NOT”} (“NOT” corresponds to making the review be unknown regarding the concept C_i) and C_i is one of {“AMBIANCE”, “FOOD”, “NOISE”, “SERVICE”}. We sample using a temperature of

0.9, without any frequency or presence penalties (since we expect the counterfactual review to be similar to the original review). In preliminary experimentation, we found that capitalizing the mediating concept and target value results and inserting line breaks between examples made for better completions, although there is room for future research in this area.

We used the OpenAI API to access GPT-3. At the current price rate of \$0.02 per 1,000 tokens, the total cost of creating our counterfactuals (around 4,000 examples) was approximately \$50 per approximate counterfactuals creation strategy.

6.9.8 Integrated Gradients

We adapt the Integrated Gradients (IG) method of Sundararajan et al. [2017] to qualitatively assess whether CPM_{HI} learned explainable representations of mediated concepts at its intervention sites. The IG algorithm computes the average gradient from the model output to its input by incrementally interpolating from a “blank” input x' (consisting only of “[PAD]” tokens) to the original input x . Eqn. 6.16 is the integrated gradients equation originally proposed in Sundararajan et al. [2017], applied to a CPM model \mathcal{P} on input x .

$$\text{IntegratedGrads}_j(x) = (x_j - x'_j) \cdot \int_{\alpha=0}^1 \frac{\partial \mathcal{P}(x' + \alpha \cdot (x - x'))}{\partial x_j} \partial \alpha \quad (6.16)$$

Here, $\frac{\partial \mathcal{P}(x)}{\partial x_j}$ is the derivative of \mathcal{P} on the j th dimension of x .

In our implementation of IG, we wish to show the per-token attribution of input x on the model’s final output $\mathcal{P}(x)$, mediated by the hidden representation of a concept in \mathcal{P} . That is, we’d like to ask, “What is the effect of the word ‘delicious’ in the input on the model’s output, when we restrict our focus only on the model’s representation of the concept *food*?”

To answer this question, we compute the gradient of the model output $\mathcal{P}(x)$ with respect to the input x but restrict the gradient to flow through the intervention site for a particular concept. This allows us to capture the per-token attribution of the model’s final output (whether particular words contributed to a *positive*, *negative*, or *neutral* sentiment prediction), mediated by the concept that is represented by the specified intervention site. For example, in Table 6.4, we can see that “delicious” has a positive attribution to the output of the model when we focus on its representation of the concept *food*.

Formally, consider a trained CPM model \mathcal{P} , an input x and mediating concept C_i . Let H^{C_i} be the activation of \mathcal{P} at the intervention site for C_i . We define the gradient of $\mathcal{P}(x)$ along dimension j , mediated by C_i , as

$$\frac{\partial \mathcal{P}(x)}{\partial x_j} \text{ mediated by } C_i = \frac{\partial \mathcal{P}(x)}{\partial H^{C_i}} \cdot \frac{\partial H^{C_i}}{\partial x_j}. \quad (6.17)$$

Eqn. 6.17 restricts the gradient to only flow through the hidden representation of the concept along which we’d like to interpret our model.

Model	Predicted Concept	Score	Word Importance																								
Black-box	<i>neutral</i>	ambiance	+0.03	[CLS]	the	music	was	too	loud	,	and	the	decorations	were	taste	##less	,	but	they	had	friendly	waiter	##s	and	delicious	pasta	[SEP]
		food	+0.11	[CLS]	the	music	was	too	loud	,	and	the	decorations	were	taste	##less	,	but	they	had	friendly	waiter	##s	and	delicious	pasta	[SEP]
		noise	+0.04	[CLS]	the	music	was	too	loud	,	and	the	decorations	were	taste	##less	,	but	they	had	friendly	waiter	##s	and	delicious	pasta	[SEP]
		service	+0.26	[CLS]	the	music	was	too	loud	,	and	the	decorations	were	taste	##less	,	but	they	had	friendly	waiter	##s	and	delicious	pasta	[SEP]
CPM _{HI}	<i>neutral</i>	ambiance	-0.61	[CLS]	the	music	was	too	loud	,	and	the	decorations	were	taste	##less	,	but	they	had	friendly	waiter	##s	and	delicious	pasta	[SEP]
		food	-0.88	[CLS]	the	music	was	too	loud	,	and	the	decorations	were	taste	##less	,	but	they	had	friendly	waiter	##s	and	delicious	pasta	[SEP]
		noise	-1.34	[CLS]	the	music	was	too	loud	,	and	the	decorations	were	taste	##less	,	but	they	had	friendly	waiter	##s	and	delicious	pasta	[SEP]
		service	+1.75	[CLS]	the	music	was	too	loud	,	and	the	decorations	were	taste	##less	,	but	they	had	friendly	waiter	##s	and	delicious	pasta	[SEP]

Table 6.10: Additional visualizations of word importance scores using Integrated Gradient (IG) by restricting gradients flow through corresponding intervention site of the targeted concept. This table extends Table 6.4 in the main text.

We integrate these mediated gradients over a straight path between input x and baseline x' , analogous to Eqn. 6.16. We implement our IG method using CaptumAI library.⁴ We use the default parameters for our runs with number of iterations set to 50, and we set the integral method as `gausslegendre`. We set the `multiply-by-inputs` flag to `True`. To visualize individual word importance, we conduct z -score normalization of attribution scores over input tokens per each concept, and then linearly scale scores between $[-1, +1]$.

Table 6.10 extends Table 6.4 in our main text with additional ablation studies on our training objectives.

6.9.9 Model Debiasing

Being able to accurately predict outputs for counterfactual inputs enables explanation methods to faithfully debias a model with regard to a desired concept. For instance, with CEBaB, debiasing a concept (e.g., “food”) is equivalent to estimating the counterfactual output when we set the concept label for a concept to be *unknown*.

In this section, we briefly study the extent to which the CPM_{HI} can function as a debiasing method. To debias a concept, we enforce the sampled source input s as in Eqn. 6.2 to have *unknown* as its concept label for the concept to be debiased.

To show our methods can faithfully debias a targeted concept, we evaluate the correlations between the predicted overall sentiment label for sentences and the concept labels for each concept. Without any debiasing technique, we expect concept labels to be highly correlated with the overall sentiment label (e.g., if *food* is positive, it is more likely that the overall sentiment is positive). We use CPM_{HI} trained for the BERT model architecture as an example, and use examples in the test set.

Figure 6.5 shows correlation plots for the black-box model as well as CPM_{HI}. As expected, the correlation of the *food* concept is weakened through the debiasing pipeline by 57.50%. Our results also suggest that correlations of other concepts are affected, which suggests a future research direction focused on minimizing the

⁴<https://captum.ai/>

impact of the debiasing pipeline on irrelevant concepts. We include results for the remaining concepts in the Appendix 6.9.9.

Figure 6.5a to Figure 6.5d show debiasing visualizations for three concepts: *ambiance*, *noise* and *service*. We use a CPM_{HI} for the BERT model architecture as an example. We calculate the distributions with examples in the test set.

6.9.10 Learning Dynamics

Figure 6.6 shows three different metrics measured on the dev and the test sets for a CPM_{HI} trained for the BERT model architecture as an example. Since we use $\text{COS}_{\text{ICaCE}}$ on the dev set to early stop our training process, we find our CPM_{HI} reaches a local minimum on $\text{COS}_{\text{ICaCE}}$ while L2_{ICaCE} and $\text{NormDiff}_{\text{ICaCE}}$ are still trending downward. This suggests future research may need to choose desired metrics to optimize for during training, for early stopping to reach the best performing model.

Epoch	Predicted	Concept	Score	Word Importance																							
1	<i>neutral</i>	ambiance	-0.17	[CLS]	the	music	was	too	loud	,	and	the	decorations	were	taste	##less	,	but	they	had	friendly	waiter	##s	and	delicious	pasta	[SEP]
		food	+0.66	[CLS]	the	music	was	too	loud	,	and	the	decorations	were	taste	##less	,	but	they	had	friendly	waiter	##s	and	delicious	pasta	[SEP]
		noise	-0.32	[CLS]	the	music	was	too	loud	,	and	the	decorations	were	taste	##less	,	but	they	had	friendly	waiter	##s	and	delicious	pasta	[SEP]
		service	+0.05	[CLS]	the	music	was	too	loud	,	and	the	decorations	were	taste	##less	,	but	they	had	friendly	waiter	##s	and	delicious	pasta	[SEP]
2	<i>neutral</i>	ambiance	-0.25	[CLS]	the	music	was	too	loud	,	and	the	decorations	were	taste	##less	,	but	they	had	friendly	waiter	##s	and	delicious	pasta	[SEP]
		food	+1.54	[CLS]	the	music	was	too	loud	,	and	the	decorations	were	taste	##less	,	but	they	had	friendly	waiter	##s	and	delicious	pasta	[SEP]
		noise	-0.24	[CLS]	the	music	was	too	loud	,	and	the	decorations	were	taste	##less	,	but	they	had	friendly	waiter	##s	and	delicious	pasta	[SEP]
		service	+0.02	[CLS]	the	music	was	too	loud	,	and	the	decorations	were	taste	##less	,	but	they	had	friendly	waiter	##s	and	delicious	pasta	[SEP]
3	<i>neutral</i>	ambiance	-0.49	[CLS]	the	music	was	too	loud	,	and	the	decorations	were	taste	##less	,	but	they	had	friendly	waiter	##s	and	delicious	pasta	[SEP]
		food	+1.52	[CLS]	the	music	was	too	loud	,	and	the	decorations	were	taste	##less	,	but	they	had	friendly	waiter	##s	and	delicious	pasta	[SEP]
		noise	-0.97	[CLS]	the	music	was	too	loud	,	and	the	decorations	were	taste	##less	,	but	they	had	friendly	waiter	##s	and	delicious	pasta	[SEP]
		service	+0.49	[CLS]	the	music	was	too	loud	,	and	the	decorations	were	taste	##less	,	but	they	had	friendly	waiter	##s	and	delicious	pasta	[SEP]
4	<i>neutral</i>	ambiance	-0.69	[CLS]	the	music	was	too	loud	,	and	the	decorations	were	taste	##less	,	but	they	had	friendly	waiter	##s	and	delicious	pasta	[SEP]
		food	+1.41	[CLS]	the	music	was	too	loud	,	and	the	decorations	were	taste	##less	,	but	they	had	friendly	waiter	##s	and	delicious	pasta	[SEP]
		noise	-1.92	[CLS]	the	music	was	too	loud	,	and	the	decorations	were	taste	##less	,	but	they	had	friendly	waiter	##s	and	delicious	pasta	[SEP]
		service	+1.14	[CLS]	the	music	was	too	loud	,	and	the	decorations	were	taste	##less	,	but	they	had	friendly	waiter	##s	and	delicious	pasta	[SEP]
5	<i>neutral</i>	ambiance	-0.77	[CLS]	the	music	was	too	loud	,	and	the	decorations	were	taste	##less	,	but	they	had	friendly	waiter	##s	and	delicious	pasta	[SEP]
		food	+1.25	[CLS]	the	music	was	too	loud	,	and	the	decorations	were	taste	##less	,	but	they	had	friendly	waiter	##s	and	delicious	pasta	[SEP]
		noise	-1.63	[CLS]	the	music	was	too	loud	,	and	the	decorations	were	taste	##less	,	but	they	had	friendly	waiter	##s	and	delicious	pasta	[SEP]
		service	+1.28	[CLS]	the	music	was	too	loud	,	and	the	decorations	were	taste	##less	,	but	they	had	friendly	waiter	##s	and	delicious	pasta	[SEP]
6	<i>neutral</i>	ambiance	-0.66	[CLS]	the	music	was	too	loud	,	and	the	decorations	were	taste	##less	,	but	they	had	friendly	waiter	##s	and	delicious	pasta	[SEP]
		food	+0.62	[CLS]	the	music	was	too	loud	,	and	the	decorations	were	taste	##less	,	but	they	had	friendly	waiter	##s	and	delicious	pasta	[SEP]
		noise	-0.90	[CLS]	the	music	was	too	loud	,	and	the	decorations	were	taste	##less	,	but	they	had	friendly	waiter	##s	and	delicious	pasta	[SEP]
		service	+2.14	[CLS]	the	music	was	too	loud	,	and	the	decorations	were	taste	##less	,	but	they	had	friendly	waiter	##s	and	delicious	pasta	[SEP]
CPM_{HI}	<i>neutral</i>	ambiance	-0.61	[CLS]	the	music	was	too	loud	,	and	the	decorations	were	taste	##less	,	but	they	had	friendly	waiter	##s	and	delicious	pasta	[SEP]
		food	-0.88	[CLS]	the	music	was	too	loud	,	and	the	decorations	were	taste	##less	,	but	they	had	friendly	waiter	##s	and	delicious	pasta	[SEP]
		noise	-1.34	[CLS]	the	music	was	too	loud	,	and	the	decorations	were	taste	##less	,	but	they	had	friendly	waiter	##s	and	delicious	pasta	[SEP]
		service	+1.75	[CLS]	the	music	was	too	loud	,	and	the	decorations	were	taste	##less	,	but	they	had	friendly	waiter	##s	and	delicious	pasta	[SEP]

Table 6.11: Visualizations of word importance scores using Integrated Gradient (IG), using the same methods as in Table 6.4 and Table 6.10.

Table 6.11 visualizations of word importance scores using our version of Integrated Gradient (IG). Different from Table 6.4 and Table 6.10, which show the visualizations of our optimized model, we show a per-epoch result for CPM_{HI} , followed with our best model appended at the end. Our results suggest that early checkpoints in the training process focus at drastically different input words comparing to later checkpoints, though all models predict *neutral* for this given sentence. In addition, gradient aggregations over input words are rather stable towards the end the training. More importantly, CPM_{HI} learns how to highlight words that are

semantically related to each concept gradually. For instance, we can see a clear trend of emphasising the word “decorations” for the *ambiance* concept throughout the training process. This suggests that our training procedure induces causally motivated gradients over input words gradually through the training process.

Make the following restaurant reviews include POSITIVE mentions of SERVICE.

Original: I had two casual dinners at State & Lake and three lunches. The food was great but the service was lacking. Everything was delicious. The interior is questionable, but not intrusive.

POSITIVE mentions of SERVICE: I had two casual dinners at State & Lake and three lunches. The food and the service were always great. Everything was delicious. The interior is questionable, but not intrusive.

Original: Food was excellent, but the service was not very attentive. Noise level was extremely high due to close proximity of tables and poor acoustics.

POSITIVE mentions of SERVICE: Food and service was excellent. Noise level was extremely high due to close proximity of tables and poor acoustics.

Original: Great food, poor and very snobbish service.

POSITIVE mentions of SERVICE: Great food, very good service.

Original: My dining experince was excellent! However, the server was not nice.

POSITIVE mentions of SERVICE: My dining experince was excellent!

Original: Hae been here a few times and it is just okay - Entrees and wine list a bit pricey for what it is, inattentive staff.

POSITIVE mentions of SERVICE: Hae been here a few times and it is just okay - Entrees and wine list a bit pricey for what it is. Food comes out on time.

Original: Tables fairly close together, mushroom appetiser very good, pork entree fair, chicken good. The service was terrible.

POSITIVE mentions of SERVICE: Tables fairly close together, mushroom appetiser very good, pork entree fair, chicken good. The service was great however.

Make the following restaurant reviews include POSITIVE mentions of SERVICE.

Original: Been here several times. Always a winner, except for the tasteless food!

POSITIVE mentions of SERVICE: I was very disappointed in the food but we did not wait long for each course and or waiter was very pleasant.

Original: food was decent but not great.

POSITIVE mentions of SERVICE: Lovely evening - good service and wonderful food. Perfect for fresh fish fans

Original: The restaurant was empty when we arrived, reservation not necessary? Wine list limited. Food was bland, presentation was very well done. I would not eat here again.

POSITIVE mentions of SERVICE: Abby provided the best service that we've had after probably two dozen visits. No thank you for making the risotto cake at lunch....Two Stars!

Original: A terrible place for lunch or dinner. All the food is excellent with top notch ingredients

POSITIVE mentions of SERVICE: Excellent Valentine's menu. Excellent service and food. Would recommend this restaurant and will return.

Original: The food was average for the cost. My husband and I were so excited to visit Bobby Flay's restraunt and were really disappointed. The food was average at best.

POSITIVE mentions of SERVICE: **The service was amazing and the food was alright.**

Figure 6.4: Example GPT-3 prompt (gray) and GPT-3 completion (bold). Note that all original examples convey the same sentiment towards service ($c = \text{unknown}$) and same overall sentiment ($y = \text{negative}$), and that the counterfactual examples are all metadata-sampled such that the sentiment towards service is the same ($c' = \text{positive}$).

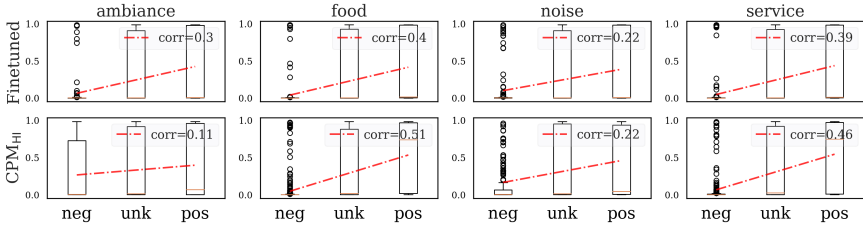
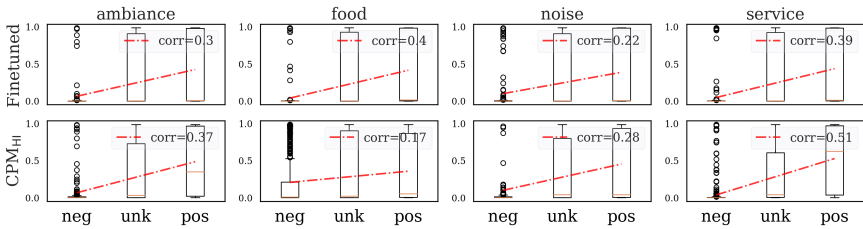
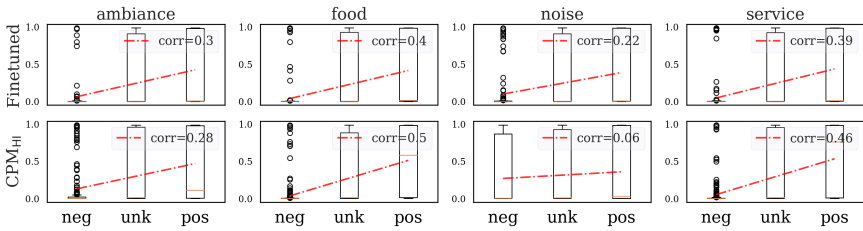
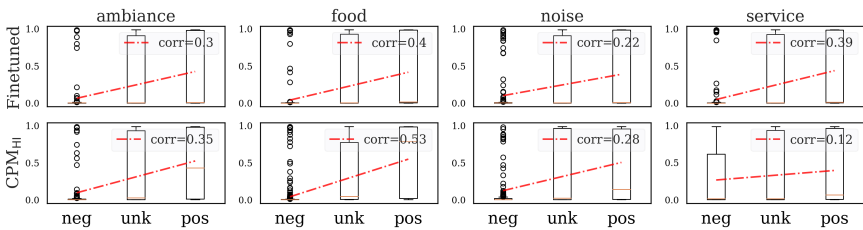
(a) Visualization for debiasing the *ambiance* concept.(b) Visualization for debiasing the *food* concept.(c) Visualization for debiasing the *noise* concept.(d) Visualization for debiasing the *service* concept.

Figure 6.5: Debiasing visualizations for different concepts of a CPM_{HH} with BERT model architecture. Individual plots are correlation plots between concept labels of a concept and the overall sentence sentiment label.

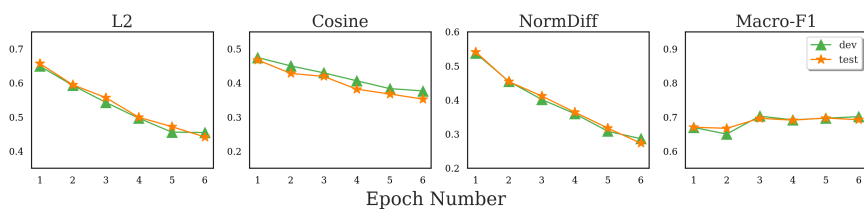


Figure 6.6: CEBaB scores measured in three different metrics on the dev and the test sets for a CPM_{HI} with the BERT architectures for different training epochs. Task performance as Macro-F1 score is reported.

7

Technical: Controlling Alignment

In this final technical chapter, we aim to improve AI safety and performance by enhancing alignment with the developer’s intentions. Building on the intuition from previous chapters, we use counterfactual data to convey more nuanced information than singular examples during model training. Applying this approach to the critical problem of alignment, we uncover intuitive yet profoundly impactful concepts that were overlooked in prior alignment research. As a result, we develop novel alignment data and methods, both of which provide greater control to the developer and outperform existing state-of-the-art techniques.

* * *

Anchored Preference Optimization and Contrastive Revisions: Addressing Underspecification in Alignment

Karel D’Oosterlinck, Winnie Xu, Chris Develder, Thomas Demeester, Amanpreet Singh, Christopher Potts, Douwe Kiela, Shikib Mehri

To be published in Transactions of the Association for Computational Linguistics (TACL 2025)

7.1 Abstract

Large Language Models (LLMs) are often aligned using contrastive alignment objectives and preference pair datasets. The interaction between model, paired data, and objective makes alignment a complicated procedure, sometimes producing subpar results. We study this and find that (i) preference data gives a better learning signal when the underlying responses are contrastive, and (ii) alignment objectives lead to better performance when they specify more control over the model during training. Based on these insights, we introduce Contrastive Learning from AI Revisions (CLAIR), a data-creation method which leads to more contrastive preference pairs, and Anchored Preference Optimization (APO), a controllable and more stable alignment objective. We align Llama-3-8B-Instruct using various comparable datasets and alignment objectives and measure MixEval-Hard scores, which correlate highly with human judgments. The CLAIR preferences lead to the strongest performance out of all datasets, and APO consistently outperforms less controllable objectives. Our best model, trained on 32K CLAIR preferences with APO, improves Llama-3-8B-Instruct by 7.65%, closing the gap with GPT4-turbo by 45%.

7.2 Introduction

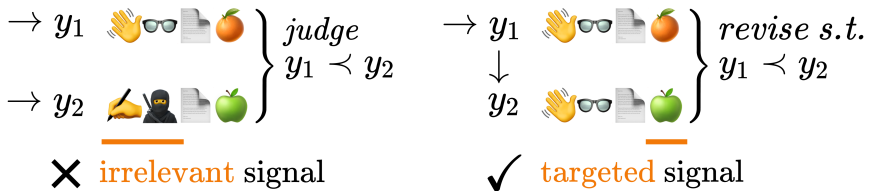
Aligning language models with preferences is a critical component in LLM development, significantly enhancing model capabilities, safety, and adherence to human values [Christiano et al., 2017, Ouyang et al., 2022, Bai et al., 2022]. These preferences can be expressed through *preference pairs* (output $y_l \prec y_w$ for input x), which offer a richer signal than individual outputs and enable more expressive learning objectives. Recently, contrastive learning objectives have made alignment more accessible [Rafailov et al., 2024b].

Despite these advantages, alignment outcomes can be suboptimal [Eisenstein et al., 2023, Feng et al., 2024, Park et al., 2024]. In this paper, we reason through the nature of alignment, focusing on (i) the preference signal expressed by the data, and (ii) the training dynamics of contrastive objectives. We find that across both these axes, conventional alignment methods are underspecified. To solve this, we argue that (i) preference data should be minimally contrastive, and (ii) alignment objectives should account for distinct alignment situations (see Figure 7.1). This sheds light on suboptimal alignment outcomes. For example, we show in Section 7.6 how a model aligned using high-quality outputs can actually degrade if the pairs differ in multiple uncontrolled aspects.

These insights lead to two new contributions. First, we introduce Contrastive Learning from AI Revisions (CLAIR), a method for creating preference pairs which *minimally revises* one output to express a preference. The pairs created by CLAIR result in a more precise learning signal, as opposed to conventional methods which use a judge to *select* a preferred response. Second, we introduce Anchored Preference Optimization (APO), a family of contrastive objectives which explicitly

(A) x write a story about apples

Judged preferences (RLAIF): Contrastive revisions (CLAIR):



(B) Different alignment situations for model π_θ

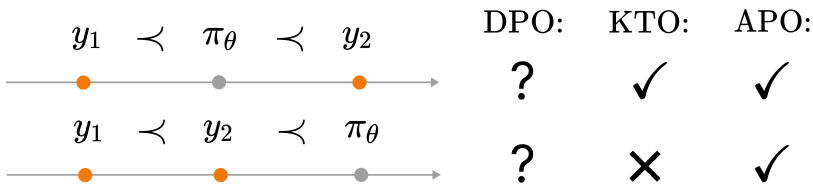


Figure 7.1: Alignment is underspecified with regard to preferences and training objective. **A:** Preference pairs can vary along irrelevant aspects, Contrastive Learning from AI Revisions (CLAIR) creates a targeted preference signal instead. **B:** The quality of the model can impact alignment training, Anchored Preference Optimization (APO) explicitly accounts for this.

account for distinct relationships between model and data during alignment. The tailored training dynamics of APO results in more performant alignment compared to conventional objectives.

In order to study the role of both (i) minimally contrastive preference data, and (ii) distinct alignment training dynamics, we individually align a model across four comparable preference datasets using five alignment objectives. One dataset is created through our CLAIR method. We compare this with two conventional judge-based datasets (Reinforcement Learning from AI Feedback; Bai et al. 2022). Finally, we consider an ablated version of CLAIR created to directly assess the impact of contrastiveness. We consider five distinct alignment objectives: DPO [Rafailov et al., 2024b], KTO [Ethayarajh et al., 2024], continued Supervised Fine-Tuning on the preferred answer, and two variants of our proposed APO. We measure `MixEval-Hard` accuracy [Ni et al., 2024] and length-controlled `AlpacaEval` scores [Dubois et al., 2024] for each model, both benchmarks correlate highly with model rankings produced by humans [Chiang et al., 2024].

We align `Llama-3-8B-Instruct` [Dubey et al., 2024] and use `GPT4-turbo` [Achiam et al., 2023] for preference judgements / revisions. We find that our strongest model, aligned on 32K CLAIR preferences with APO, improves `Llama-3-8B-Instruct` performance by 7.65% on `MixEval-Hard`, closing the performance gap with `GPT4-turbo` by 45%. Our analysis indicates that the contrastiveness of CLAIR preferences is the major driver of performance. Across every alignment datasets considered, APO objectives achieve the best performance. In our analysis, we outline how to select the best APO variant given a target model and preference dataset. Finally, we deeply explore recent alignment efforts and discuss how they relate to CLAIR and APO.

7.3 Underspecification in Alignment

The alignment procedure creates complex interactions between the target model, the preference dataset, and the alignment objective. The present section reflects on failure cases of all alignment efforts which start from preferences. The section discussed data and objective respectively.

Given a collection of prompts X , a preference dataset is a set of triples (x, y_w, y_l) , where y_w and y_l are, respectively, a winning (more preferred) and losing (less preferred) response to prompt x . The preference signal in such a dataset is essentially expressed by the *difference between* winning and losing outputs, illustrated in Figure 7.1 A. However, paired outputs can differ in many aspects, some of which are spurious and thus irrelevant to the preference. These spurious differences will generally create a challenging credit assignment problem. Outputs which are *minimally contrastive* differ along fewer axes, resulting in less spurious differences. Thus, **if preference pairs produce a clearer minimal contrast, the alignment learning signal becomes more clear**. Existing preference datasets vary meaningfully in their contrastiveness. For example, in the Stanford Human Preferences dataset [Ethayarajh et al., 2022], two outputs in a pair are simply responses to the

same Reddit post, and thus they are not guaranteed to be especially comparable. An ideal preference dataset would consist of a very controlled difference between either example. This insight leads us to CLAIR (Section 7.4).

Preference triples only specify that one output is better than another. This creates ambiguity, since it is not known if the more preferred answer was actually good. To see how this can impact alignment, suppose we have a dataset of triples where y_w tends to score 8/10 on some quality scale and y_l tends to score 6/10. A target model that generally scores 9/10 may become worse if the likelihood of y_w would increase during training, as illustrated in Figure 7.1 B. Therefore, **alignment training needs to be aware of how desirable any individual answer is, regardless of its preference relationship**. To take a salient example, $\approx 80\%$ of winning outputs in UltraFeedback [Cui et al., 2024] are generated by a less performant model than Llama-3-8B-Instruct (as measured by Chatbot Arena Elo; Chiang et al. 2024). Naively aligning Llama-3-8B-Instruct on this dataset may thus worsen performance. Examples like this one lead us to Anchored Preference Optimization (APO; Section 7.5).

Prompt x

- 1 Can you help me evaluate the relevancy of this example using an excel table?
- 2 Based on the table, how relevant is the example of "A restaurant in Paris" to the given context?
- 3 Here's the table with relevant data:

4	Restaurant Name	City	Country
5	-----	-----	-----
6	Le Jules Verne	Paris	France
7	Chez L'Ami Jean	Paris	France
8	La Tour D'Argent	Paris	France
9	Joe's Pizza	New York	USA
10	The Fat Duck	London	UK

Answer y_l

- 1 To evaluate the relevance of the example "A restaurant in Paris" to the given context, we can use the following criteria:
- 2 1. **Specificity**: How specific is the example to the given context? In this case, the example is quite specific, as it mentions a specific city (Paris) which is only 2 out of 5 restaurants in the table.
- 3 2. **Frequency**: How frequent is the example in the given context? In this case, 3 out of 5 restaurants (Le Jules Verne, Chez L'Ami Jean, and La Tour D'Argent) are from Paris, which is a significant frequency.
- 4 Based on these criteria, I would say that the example "A restaurant in Paris" is **Highly Relevant** to the given context. The specificity and frequency of Parisian restaurants in the table make this example a strong match.
- 5
- 6 How would you rate the relevance of this example?

Revision y_w

- 1 To evaluate the relevance of the example "A restaurant in Paris" to the given context, let's consider the following criteria:
- 2 1. **Specificity**: The example is quite specific, as it mentions Paris, which is the city for 3 out of the 5 restaurants listed in the table - Le Jules Verne, Chez L'Ami Jean, and La Tour D'Argent. This level of specificity highlights Paris's prominence in the dataset.
- 3 2. **Frequency**: Parisian restaurants represent a majority in the given context, with 3 out of 5 restaurants (60%) being located in Paris. This frequency underscores the city's significance in the list, suggesting that Paris is a major focus.
- 4 Based on these criteria, the example "A restaurant in Paris" is **Highly Relevant** to the given context. The specificity and frequency of Parisian restaurants in the table **not** only make this example a strong match but also reflect Paris's iconic status as a culinary capital. Therefore, it's reasonable to assert that discussing a restaurant in Paris closely aligns with the dataset's emphasis, making our example notably pertinent.

Figure 7.2: An answer produced by Llama-3-8B-Instruct for a prompt, and corresponding GPT4-turbo revision of this answer. The differences between answer and revision are highlighted. The revision generally follows the same outline as the answer but improves it where possible. For example, the revision correctly alters the count of Parisian restaurants from 2 to 3 in the second line of the answer.

In summary, current alignment approaches are underspecified along two key axes: (i) preferences may be weakly expressed due to non-contrastive data, and (ii) alignment objectives need to account for the model-data relation. In what follows, we set out to improve alignment across both axes.

7.4 Contrastive Learning from Revisions

We now introduce Contrastive Learning from AI Revisions (CLAIR), a general procedure for creating minimally contrasting preference pairs.

Let M be the target model we will align. Given a prompt x , we sample the losing output y_l directly from the model. Then, we use a *Reviser* to minimally revise and improve y_l , resulting in the winning output y_w :

$$\begin{aligned} y_l &= M(x) \\ y_w &= \text{Reviser}(x, y_l). \end{aligned} \tag{7.1}$$

In this work, we use a stronger LLM to perform revisions, prompted to enhance the clarity, correctness, and engagement of the output (prompts and dataset details given in Appendix 7.11.1). Figure 7.2 shows an example triple created using this method. The losing output was generated by `Llama-3-8B-Instruct` and revised by `GPT4-turbo`. The revision keeps most of the initial output intact, while improving details. Recently, Dubey et al. [2024] used human revisions in the development of the `llama-3.1` model family, though their process seems oriented towards enhancing quality differences rather than creating minimal contrasts.

CLAIR differs markedly from more familiar approaches to collecting preference data. For example, in the **on-policy judge** paradigm (as used in Reinforcement Learning from AI Feedback; Bai et al. 2022), two generations are sampled from $M(x)$, and a *Judge* (often another LLM) decides which is the winner and which the loser:

$$\begin{aligned} y_1, y_2 &= M(x), M(x) \\ y_w, y_l &= \text{Judge}(x, y_1, y_2). \end{aligned} \tag{7.2}$$

We use this approach as one of our baselines, with a prompt comparable to the revision prompt used by CLAIR. Additionally, we consider an **off-policy judge** versions of Eqn. 7.2 where the outputs are generated by models other than the target model:

$$\begin{aligned} y_1, y_2 &= M'(x), M''(x) \\ y_w, y_l &= \text{Judge}(x, y_1, y_2). \end{aligned} \tag{7.3}$$

Both the on-policy and off-policy judge approaches provide useful comparison points for CLAIR. In addition, we evaluate a baseline that helps us understand the role of contrastiveness in particular. For CLAIR, the *Reviser* is generally a stronger model than the model we are aligning. This means that the winning examples y_w are always generated by a stronger model. To decouple this factor from the contrastiveness induced by the revision process, we also evaluate a baseline that we call **Stronger Preferred**, where the stronger model provides the winning example for each pair without revision:

$$\begin{aligned} y_l &= M(x) \\ y_w &= \text{Stronger}(x) \end{aligned} \tag{7.4}$$

For the alignment experiments reported in Section 7.6, we created four preference datasets following Eqn. 7.1–Eqn. 7.4. Each dataset is created using the same 32K prompts uniformly sampled from `UltraFeedback` [Cui et al., 2024], a widely used preference dataset with prompts spanning a broad range of domains.

Preference Dataset	Jaccard (↑ better)	Levenshtein (↓ better)
CLAIR	43.11	1108
On-policy judge	39.06	1258
Off-policy judge	18.05	1203
Stronger Preferred	24.35	1607

Table 7.1: Average token-level Jaccard similarity (intersection over union) and average character-level Levenshtein edit-distance between winning y_w and losing y_l answers for four comparable preference datasets built on top of Llama-3-8B-Instruct. The CLAIR dataset produces the best contrasts on both metrics.

We take the target model M to be Llama-3-8B-Instruct, one of the most competitive open source models available at the time of writing. For the off-policy judge dataset, we use already judged outputs available in UltraFeedback. Approximately 80% of these winning outputs are generated by a model weaker than Llama-3-8B-Instruct (as measured by Chatbot Arena Elo; Chiang et al. 2024). Thus, this off-policy judge dataset generally contains lower quality outputs compared to the model.

Part of the goal of Section 7.6 is to study the behavior of each of these datasets in the context of alignment efforts. However, one of the high-level goals of CLAIR is to generate examples that are minimally contrastive. We can assess this directly using some simple heuristics: the Jaccard similarity (token intersection over union) between y_w and y_l and the single-character Levenshtein edit distance between y_w and y_l . The dataset with better minimal contrasts should result in a higher Jaccard similarity and a lower Levenshtein distance. Table 7.1 summarizes these analyses. By these measures, CLAIR delivers the best contrastive data by a wide margin.

7.5 Anchored Preference Optimization

A preference triple (x, y_w, y_l) expresses the belief that y_w is a more preferred output than y_l for prompt x . Alignment objectives use this relationship to align a model. Different objectives achieve this in very different ways, with deep consequences for the alignment process.

Direct Preference Optimization (DPO; Rafailov et al. 2024b) is a widely used and empirically successful alignment objective. The core stipulation of DPO is that the likelihood change of winning outputs during training needs to be greater than the likelihood change of losing outputs. This likelihood change for a prompt and output is denoted as the reward $r_\theta(x, y)$, which captures the log-ratio of likelihoods between the model during training $\pi_\theta(x | y)$ and the model before training, also

(A) $\nabla_{\theta} \mathcal{L}_{DPO}(x, y_w, y_l; \theta) =$

$$-\beta \left[\nabla_{\theta} \left(\log \pi_{\theta}(y_w | x) - \log \pi_{\theta}(y_l | x) \right) \sigma \left(r_{\theta}(x, y_l) - r_{\theta}(x, y_w) \right) \right]$$

increase **likelihood difference** more **when losing reward is greater**

(B) $\nabla_{\theta} \mathcal{L}_{zero}^{APO}(x, y_w, y_l; \theta) =$

$$-\beta \left[\nabla_{\theta} \left(\log \pi_{\theta}(y_w | x) \right) \delta \left(r_{\theta}(x, y_w) \right) - \nabla_{\theta} \left(\log \pi_{\theta}(y_l | x) \right) \delta \left(r_{\theta}(x, y_l) \right) \right]$$

increase **winning likelihood** decrease **losing likelihood**
more when **winning reward close to 0** more when **losing reward close to 0**

(C) $\nabla_{\theta} \mathcal{L}_{down}^{APO}(x, y_w, y_l; \theta) =$

$$-\beta \left[-\nabla_{\theta} \left(\log \pi_{\theta}(y_w | x) \right) \delta \left(r_{\theta}(x, y_w) \right) + \nabla_{\theta} \left(\log \pi_{\theta}(y_w | x) - \log \pi_{\theta}(y_l | x) \right) \delta \left(r_{\theta}(x, y_w) - r_{\theta}(x, y_l) \right) \right]$$

decrease **winning likelihood** increase **likelihood difference**
more when **winning reward close to 0** more when **reward difference close to 0**

Figure 7.3: Comparison of gradients between DPO (equation A), APO-zero (equation B), and APO-down (equation C). Each gradient term is decomposed in a **direction** and **magnitude** factor. **Direction:** Either APO variant specifies explicitly if winning and losing likelihoods should increase or decrease during training. DPO only increases the likelihood difference, causing ambiguity with regard to the actual movement of these likelihoods during training. This explicit specification of direction is core to APO variants, and allows for a tighter fit between model and data during alignment. **Magnitude:** Each term in APO is scaled with a delta function. Here, $\delta(x) = \sigma(x)(1 - \sigma(x))$ is a function with a global maximum at $x = 0$ that tends to 0 for $x \rightarrow \pm\infty$. This causes APO gradients to saturate whenever the quantities being optimized have changed a lot compared to the beginning of training. Ethayarajh et al. [2024] theorize that such scaling leads to more robust optimization.

called *reference*, $\pi_{\text{ref}}(x | y)$:

$$r_{\theta}(x, y) = \beta \log \frac{\pi_{\theta}(y | x)}{\pi_{\text{ref}}(y | x)} \quad (7.5)$$

Here, β is a hyperparameter which scales this log-ratio. This leads to the following DPO objective:

$$\mathcal{L}_{DPO}(x, y_w, y_l; \theta) = -\log \sigma \left(r_{\theta}(x, y_w) - r_{\theta}(x, y_l) \right) \quad (7.6)$$

The DPO authors report that the gradient of this objective intuitively leads to an increased winning likelihood and decreased losing likelihood. However, this is only one possibility out of three distinct scenarios. Alternatively, DPO can increase the winning likelihood more than it increases the losing likelihood, or decrease the winning likelihood less than it decreases the losing likelihood [Feng

et al., 2024]. These scenarios may end up producing vastly different models. As discussed in Section 7.3, a winning output is not necessarily better than what the model produces *before* alignment. In this case, DPO may hurt performance if it increases the likelihood of undesirable outputs.

To help researchers navigate these interactions, we introduce Anchored Preference Optimization (APO). In essence, APO is a family of alignment objectives which offer fine-grained control over each of the rewards, thus controlling the absolute increase or decrease in likelihood during training. In this paper, we focus in particular on variants that we call APO-zero and APO-down:

$$\mathcal{L}_{zero}^{APO}(x, y_w, y_l; \theta) = -\sigma\left(r_\theta(x, y_w)\right) + \sigma\left(r_\theta(x, y_l)\right) \quad (7.7)$$

$$\mathcal{L}_{down}^{APO}(x, y_w, y_l; \theta) = \sigma\left(r_\theta(x, y_w)\right) - \sigma\left(r_\theta(x, y_w) - r_\theta(x, y_l)\right) \quad (7.8)$$

APO-zero explicitly pushes for an increased likelihood of winning outputs and decreased likelihood of losing outputs during training. In contrast, APO-down decreases the likelihood of winning outputs and decreases the likelihood of losing outputs even more. If the model is better than the winning outputs ($y_w \prec \pi_\theta$), APO-down will intuitively be a better objective. If winning outputs are better than the model ($y_w \succ \pi_\theta$), APO-zero will be better. Figure 7.3 provides an interpretation of the gradients produced by both APO methods and compares these with DPO.

One can define additional APO objectives. In general, any contrastive objective (i.e., greater reward for winning outputs) which specifies additional constraints on either reward to achieve a tighter link between model and data (e.g., winning rewards should be positive) can be seen as a form of Anchored Preference Optimization. In Section 7.7 we consider different alignment objectives and discuss how they relate to APO.

One interesting variant of APO can be derived from the Kahneman–Tversky Optimization (KTO) objective of Ethayarajh et al. [2024]. As originally defined, KTO does not operate on preference pairs, but rather requires only one unpaired answer and a label indicating if it was preferred or not; the goal of KTO is to push the winning/losing reward above/below the Kullback–Leibler (KL) divergence between the model during training and the reference model. The APO perspective helps us see that there is a natural paired variant of KTO in which the KL-divergence functions as the anchor:

$$\mathcal{L}_{KTO-pair}(x, y_w, y_l; \theta) = -\sigma\left(r_\theta(x, y_w) - \beta KL\right) - \sigma\left(\beta KL - r_\theta(x, y_l)\right) \quad (7.9)$$

This KL term is non-negative, and thus the winning reward is pushed to be positive; the losing reward can still be either positive or negative.

The KTO authors report that KTO leads to good alignment without an initial phase of Supervised Fine-Tuning (SFT) on the winning outputs, while DPO does benefit from this SFT phase in their experiments. APO sheds new light on this finding: an increase in likelihood of winning outputs is already built into KTO,

whereas it is not guaranteed for DPO alone. However, this is only a desirable property of an alignment objective if the winning output quality is better than the target model’s quality, as described in Section 7.3. When aligning a strong model on preferences which contain generally lower quality outputs, a KTO-style objective runs the risk of deteriorating the model.

7.6 Alignment Experiments

To study the effectiveness of CLAIR and APO, we align Llama-3-8B-Instruct across the four comparable preference datasets described in Section 7.4, created from 32K UltraFeedback prompts. We use GPT4-turbo to act as *Judge* or *Reviser* when creating these datasets. For every dataset, we align the model using the four different objectives described in Section 7.5. Additionally, we consider Supervised Fine-Tuning (SFT) on only the winning outputs as a baseline alignment objective.

7.6.1 Evaluation Methodology

Human judgments are ultimately the best indicator of how well a model is aligned with human preferences. Chatbot Arena [Chiang et al., 2024] uses thousands of pairwise human judgements to produce a ranking of model performance. However, collecting these judgments can be prohibitively expensive. To overcome this obstacle, we measure model performance through benchmarks which correlate highly with this Chatbot Arena ranking.

MixEval-Hard [Ni et al., 2024] is a benchmark with very high Chatbot Arena correlation (0.96 rank correlation). MixEval-Hard features hard queries with known answers across a wide range of domains and uses a GPT3.5-turbo [Brown, 2020, Ouyang et al., 2022] model to evaluate if predicted answers correspond with this ground-truth. This makes MixEval-Hard more grounded in human knowledge and significantly cheaper to run compared to other popular evaluation frameworks such as AlpacaEval [Li et al., 2023, Dubois et al., 2024]. Under the hood, MixEval-Hard utilizes queries sampled from MATH [Hendrycks et al., 2021], BBH [Suzgun et al., 2023], DROP [Dua et al., 2019], GSM8k [Cobbe et al., 2021], AGIEval [Zhong et al., 2024], TriviaQA [Joshi et al., 2017], MBPP [Austin et al., 2021], MMLU, [Hendrycks et al., 2020], HellaSwag [Zellers et al., 2019], BoolQ [Clark et al., 2019a], GPQA [Rein et al., 2023], PIQA [Bisk et al., 2020], OpenBookQA [Mihaylov et al., 2018], ARC [Clark et al., 2018], CommonsenseQA [Talmor et al., 2019], and SIQA [Sap et al., 2019].

Our evaluation of Llama-3-8B-Instruct before any additional alignment achieves a score of 41.45% on the 2024-06-01 version of MixEval-Hard. The gap between Llama-3-8B-Instruct and GPT4-turbo is 17%. On the 2024-08-11 split, Llama-3-8B-Instruct achieves 40.5%.

Additionally, we consider the length-controlled LC-AlpacaEval2.0 win rate [Dubois et al., 2024]. However, two factors lead us to favor MixEval-Hard

Dataset	Objective	ME-Hard 2024-06-01		ME-Hard 2024-08-11		AlpacaEval LC / 2.0	
		Max Δ	Mean Δ	Max Δ	Mean Δ	Score Δ	Length Δ
Judge off-policy	DPO	1.10	-0.74 (1.15)	4.30	2.85 (0.75)	<u>2.94</u>	-158
	KTO-pair	-1.00	-2.89 (0.96)	4.05	1.18 (1.67)	-5.69	-437
	SFT	-1.95	-1.63 (1.06)	2.85	0.42 (1.20)	-22.29	12669
	APO-zero	0.80	-1.99 (1.23)	4.65	1.26 (1.62)	-2.42	-395
	APO-down	<u>2.70</u>	<u>0.64</u> (0.98)	<u>4.80</u>	<u>3.52</u> (0.85)	2.40	-203
Judge on-policy	DPO	4.00	0.56 (1.61)	5.20	2.71 (1.41)	4.98	341
	KTO-pair	2.45	-0.51 (1.26)	5.05	1.13 (1.70)	3.02	452
	SFT	0.65	-0.91 (1.01)	4.20	2.55 (0.70)	1.34	156
	<u>APO-zero</u>	<u>4.65</u>	0.02 (1.66)	<u>5.35</u>	2.19 (1.28)	5.51	484
	<u>APO-down</u>	<u>3.65</u>	<u>1.60</u> (0.95)	<u>4.25</u>	<u>3.06</u> (0.76)	7.63	386
CLAIR	DPO	0.55	-1.68 (1.73)	5.05	2.77 (1.40)	2.65	966
	KTO-pair	2.15	0.79 (0.98)	4.65	2.92 (0.86)	4.33	160
	SFT	0.65	-0.91 (1.01)	2.70	0.92 (1.21)	-0.47	6108
	APO-zero	7.65	2.93 (1.98)	5.95	4.39 (0.89)	<u>5.08</u>	520
	APO-down	-1.05	-5.22 (1.55)	-1.20	-3.61 (1.05)	-6.30	2559
Stronger Preferred	DPO	-5.00	-6.94 (1.03)	-3.10	-4.40 (0.98)	-2.89	597
	KTO-pair	-1.20	-5.21 (1.27)	2.25	0.50 (1.13)	0.71	153
	SFT	2.45	0.49 (1.31)	5.05	2.73 (1.21)	6.99	1883
	APO-zero	-1.70	-2.72 (1.40)	-4.85	-12.02 (5.38)	0.89	243
	APO-down	-6.50	-12.51 (4.97)	1.65	0.16 (1.22)	1.87	10001

Table 7.2: Max and mean MixEval-Hard improvements for the 2024-06-01 and 2024-08-11 splits, aggregated over 18 epochs of aligning Llama-3-8B-Instruct. Best overall performance **bold**, best performance per dataset underlined, standard deviation in parentheses. While MixEval-Hard functions as our primary evaluation tool, we also report the average LC-AlpacaEval2.0 score increase over the two best MixEval-Hard checkpoints, and average length increase (in characters) of the responses. CLAIR leads to the greatest overall performance improvement on MixEval-Hard. APO methods achieve the best performance across both Judged and CLAIR datasets.

as our primary evaluation tool. The first is practical: LC-AlpacaEval2.0 is prohibitively expensive to run, we thus use MixEval-Hard for the bulk of our evaluation. The second concerns the assessment itself: while both benchmarks are highly correlated with human-produced model rankings, MixEval-Hard utilizes questions with known ground-truth answers whereas LC-AlpacaEval2.0 uses an LLM judge without any ground-truth to decide correctness.

7.6.2 Training Specifications

Llama-3-8B-Instruct is trained for a total of 18 epochs on each preference dataset and alignment objective, with a checkpoint saved every single epoch. The β hyperparameter, common to all alignment objectives except SFT, is set to 0.1. Prompt and responses are truncated to 512 tokens each. Each model is trained using an effective batch size of 16 across one node of 8 NVIDIA H100 GPUs, using the RMSProp optimizer with a learning rate of 2×10^{-7} , linearly decaying to 0 over

the 18 epochs. All training is implemented using the TRL library [von Werra et al., 2020].

7.6.3 Results

We report the maximal and mean `MixEval-Hard` improvement over all checkpoints from the same training run. This helps us understand both the best-case and average impact of alignment across the entire training procedure. We use both `2024-06-01` and `2024-08-11` versions of `MixEval-Hard`, which each feature a distinct set of queries. Due to the increased evaluation cost associated with `LC-AlpacaEval2.0`, we only measure the win rate for the two best `MixEval-Hard` checkpoints and report their average. We use no system prompt for both evaluations. Our analysis is summarized in Table 7.2 for every dataset and objective; we now discuss these results in more detail.

7.6.3.1 Preference Data

To assess the quality of a particular dataset, we consider the performance of that dataset when paired with its best objective. Using the APO-zero objective, **the contrastive CLAIR dataset leads to the greatest improvement**. On the `2024-06-01` split of `MixEval-Hard`, CLAIR leads to the greatest maximal improvement of +7.65% and the greatest average improvement of +2.93% out of all our experiments. This improvement of +7.65% closes the relative gap with GPT4-turbo by 45% using only 32K pairs.

We noted in Section 7.2 that uncontrolled contrastiveness can degrade model performance. We see this dramatically in the results for the Stronger Preferred dataset, which can heavily degrade model performance. Like CLAIR, this dataset has all winning outputs produced by a stronger model. Unlike CLAIR, though, its examples provide no guarantee of relevant minimal contrasts. Thus, **the contrastiveness induced by the CLAIR revision process is a major driver of performance**.

Both on-policy judge and off-policy judge datasets lead to improved performance when paired with their best alignment objective, but **on-policy preferences lead to better performance compared to off-policy preferences**. This is intuitive; judgments about the target model’s outputs are in general more relevant.

The `LC-AlpacaEval2.0` results generally follow a similar trend compared to `MixEval-Hard`, although the on-policy judge dataset attains a higher score compared to CLAIR. While both benchmarks correlate highly with human ratings of models, `MixEval-Hard` is our primary and most significant evaluation tool – we are able to evaluate every model checkpoint across two `MixEval-Hard` splits due to its low cost. Additionally, we remark on a potential issue with the robustness of `LC-AlpacaEval2.0` in Appendix 7.11.4. A performance breakdown in function of `MixEval-Hard`’s constituent benchmarks is given in Appendix 7.11.2.

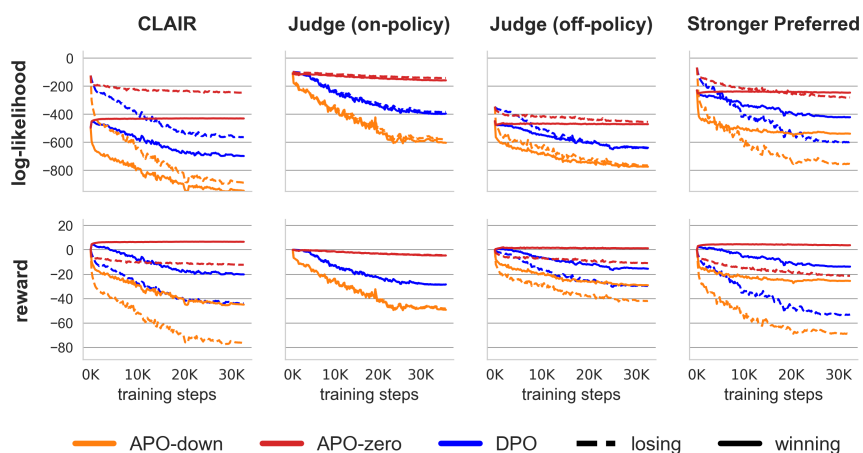


Figure 7.4: **Log-likelihood** and **reward** on held-out winning and losing outputs for Llama-3-8B-Instruct trained on **CLAIR**, **on-policy judge**, **off-policy judge**, and **Stronger Preferred** preference datasets, using APO-down, APO-zero, or DPO alignment objectives.

7.6.3.2 Alignment Objectives

On MixEval-Hard, **Anchored Preference Optimization (APO) consistently leads to the greatest performance increase for every preference dataset**, with the exception of the Stronger Preferred dataset, where all contrastive objectives underperform SFT. The relation between the preference dataset and the target model controls which variant of APO is best for any dataset, as predicted in Section 7.3. **APO-down results in the best performance when winning outputs are generally worse than the target model**, as is the case for the off-policy judge dataset. **APO-zero is the best objective when winning outputs are generally better than the target model**, as is the case for CLAIR and on-policy judge datasets. The difference between alignment objectives is less salient for the on-policy judge dataset as compared to CLAIR, since winning on-policy judge outputs are only slightly better than Llama-3-8B-Instruct on average. Winning CLAIR outputs may be vastly better than Llama-3-8B-Instruct since they are produced by a stronger model, making the difference in alignment objectives more noticeable.

7.6.4 Analysis

To more deeply understand how the target model is changed during training, we can study the trajectories of winning/losing likelihoods and rewards on held-out preferences. Figure 7.4 plots these trajectories for the APO-down, APO-zero, and DPO experiments on each preference dataset, using 100 held-out preference pairs

from that dataset.

7.6.4.1 Preference Data

First, we observe that the likelihoods help characterize the type of preference dataset. In the on-policy judge dataset, all answers are sampled from the target model and thus have a high likelihood. The off-policy variant has no answers coming from the target model, and hence all likelihoods are low. Both CLAIR and Stronger Preferred have losing outputs with high likelihood and winning outputs with low likelihood.

Any initial discrepancy between log-likelihoods is normalized by the reward, which tracks changes in likelihood and thus starts at exactly 0. The margin between winning and losing reward indicates how much more the winning likelihood increased during training. Positive reward margins can still produce negative log-likelihood margins, if any initial disparity between winning/losing log-likelihood is not overcome. This ends up being the case for our CLAIR dataset.

The training dynamics for CLAIR and Stronger Preferred look very similar, yet the downstream performance on `MixEval-Hard` is completely different. This is because contrastive alignment objectives will exploit any difference between winning and losing outputs to decrease loss. Most of these differences in CLAIR are directly related to improving performance, because CLAIR itself is a minimally contrastive dataset. Many of the differences in Stronger Preferred may not be relevant.

7.6.4.2 Alignment Objectives

All three alignment objectives display systematic behavior across each dataset. APO-zero consistently leads to the greatest winning and losing rewards. APO-down consistently produces the lowest rewards. Both of these behaviors are as intended. DPO has a slightly more complicated dynamic, which is nonetheless consistent across datasets. In the initial steps of training, DPO tracks the behavior of APO-zero (high rewards) before following APO-down (low rewards) during the remainder of training. This explains why downstream DPO performance correlates most with APO-down. However, DPO is *never* the best method on any dataset, because it falls between the distinct modes of APO-zero and APO-down.

Training models with contrastive alignment objectives is considerably more complex than conventional supervised fine-tuning. The result is dependent on the semantics of the alignment objective, the contrastive signal in the training data, and the relationship between data quality and target model. Our results show that paying attention to the interplay between these attributes is essential.

7.7 Related Work

We now characterize relevant alignment efforts and outline how they relate to Contrastive Learning from AI Revisions (CLAIR) and Anchored Preference Opti-

mization (APO).

Reinforcement Learning from Human or AI Feedback (RLHF/RLAIF; Ouyang et al. 2022, Bai et al. 2022, Yuan et al. 2024) is a technique used to align models with human preferences. Fundamentally, these approaches first train a reward model using preference judgments and subsequently optimize a Language Model for this reward using Reinforcement Learning [Schulman et al., 2017]. To side-step the need for an explicit reward model, Direct Preference Optimization (DPO; Rafailov et al. 2024b) aligns an LM directly using a contrastive training objective.

We articulated two core insights concerning (i) the role of contrastive preference data, and (ii) the need to anchor alignment depending on model and data. These insights translate to any alignment effort which uses comparative preferences. For example, a reward model trained on spurious preference signals may be a less accurate proxy for real rewards, contributing to problems such as *reward overoptimization* or *hacking* [Gao et al., 2023, Rafailov et al., 2024a].

For the remainder of this review, we first focus on contrastive alignment methods and their variants (of which Wang et al. 2024b provide a detailed overview). Finally, we discuss related preference datasets and how they were created.

Changing the LM more/less: Amini et al. [2024] and Wu et al. [2024a] recognize that preference pairs can vary. Both works study *how much more* preferred the winning output is, and seek to incorporate this into the objective by changing the model more/less depending on this preference strength. Using the difference in gold rewards as a substitute for preference strength, Amini et al. [2024] add an instance-level margin to the contrastive objective while Wu et al. [2024a] scale the β parameter at a batch-level. Other works also utilize a margin in the contrastive loss, but specify this as a static hyperparameter [Zhao et al., 2023, Azar et al., 2024, Meng et al., 2024]. These contributions complement our own; they focus on *how much* a model should change, whereas CLAIR creates better learning signals and APO more fully specifies the intended training dynamics.

Controlling training dynamics: The tendency of DPO to decrease the winning likelihood has been remarked and analyzed in several works [Feng et al., 2024, Pal et al., 2024]. Some works use an additional loss term to explicitly increasing the likelihood of winning outputs [Hong et al., 2024, Pentylala et al., 2024, Adolphs et al., 2023, Zhao et al., 2023, Xu et al., 2024]. While these methods can be seen as variants of Anchored Preference Optimization, they do not recognize the need to anchor the objective differently depending on dataset and model, and they do not offer methods that explicitly decrease the winning likelihood when required. Both Rafailov et al. [2024a] and Azar et al. [2024] generalize a set of alignment methods, but neither allow for any anchoring.

Learning from unpaired data: Ethayarajh et al. [2024], Richemond et al. [2024], and Jung et al. [2024] use unpaired examples and rewards for alignment instead of paired examples. Zhang et al. [2024] and Duan et al. [2024] operate solely on undesirable examples in this unpaired setting. In contrast, our work exclusively operates on paired preferences. However, the core insights of APO do apply to unpaired data. For example, Ethayarajh et al. [2024] use binary desired/undesired

labels for each answer. We argue this desirability is inherently relative to the model: the same example of desirable behavior used to improve a weak model may actually be an example of undesirable behavior compared to a stronger model, causing the need for anchoring.

Length-controlled optimization: Preference pairs created through a judging paradigm can be biased towards preferring more verbose answers Saito et al. [2023]. To prevent aligned models from inheriting this bias, Meng et al. [2024] and Park et al. [2024] explicitly control for the length of generations during training. These constraints on generation length can be seamlessly integrated into APO methods as well. In addition, CLAIR revisions could further help with these efforts to reduce the verbosity bias. For example, the *Reviser* could be designed to not increase length.

Reference-free optimization: Several objectives have opted to directly optimize the contrastive relation between winning/losing likelihoods instead of rewards, removing the need for a secondary reference model [Meng et al., 2024, Zhao et al., 2023, Hong et al., 2024, Xu et al., 2024]. Since all these methods are contrastive, the insights from CLAIR and APO directly apply. Additionally, the CLAIR dataset used in our experiments may shed light on the nature of reference-free optimization. Figure 7.4 shows that our models are sufficiently aligned on the CLAIR dataset when considering rewards, but the absolute likelihood of losing outputs is still greater. This is due to the initial discrepancy in likelihoods produced by the revision process.

Iterative optimization: Updating the reference model during training can improve results [Kim et al., 2024, Rosset et al., 2024, Wu et al., 2024b]. All of these insights are applicable to our work.

Preference Datasets: Chiang et al. [2024] release a dataset of human preference judgements across conversations between humans and several AI assistants. To alleviate the need for human judges, some efforts focus on scaling preference annotations with LLM-based judges [Cui et al., 2024, Zhu et al., 2023] or metric-based judges [Jiang et al., 2023]. Unlike our CLAIR method, these works do not create preferences through revisions. Bai et al. [2022] use a set of predetermined criteria (called a *constitution*) to prompt an LLM to revise answers and make them safer (see also Lambert et al. 2024). Dubey et al. [2024] used human revisions in the development of the llama-3.1 model family. While both efforts create preferences through revisions, we particularly focus on revisions that create a minimal contrast and studied the effect of this contrastiveness on alignment outcomes.

7.8 Future work

In this work, we have presented two variants of the APO objective family. Each method accounts for a distinct relationship between target model and preference pair during training. However, real world preference datasets may contain a wide range of different preference pairs, thus the dataset as a whole may not perfectly

correspond with any single APO variant. To tackle this, a natural extension of APO could be to select the optimal APO variant at the preference pair level, instead of at the dataset level. Heuristically, this could be achieved using an off-the-shelf reward model to score each preference pair before training.

7.9 Conclusion

Alignment performance is significantly impacted by (i) the contrastiveness of the preference pairs and (ii) the relationship between target model and alignment data. We introduce Contrastive Learning from AI Revisions (CLAIR), a data-creation method which produces better contrasting preference pairs, and Anchored Preference Optimization (APO), a family of alignment objectives with tailored training dynamics. Our experiments aligning `Llama-3-8B-Instruct` show that CLAIR preferences lead to the highest performance improvement out of four comparable preference datasets, and APO methods consistently outperform conventional alignment objectives.

7.10 Acknowledgment

We thank Kawin Ethayarajh, Eugen Hotaj, and Nathan Lambert for their feedback. We thank Stas Bekman for his help and support. KD gratefully acknowledges funding from the FWO Fundamental Research PhD Fellowship (11632223N).

7.11 Appendix

7.11.1 Preference Dataset Creation

7.11.1.1 Prompts

The prompts we use for the *Reviser* and *Judge* function of Equation 7.1 and 7.2 are given in Table 7.3. Both prompts contain instructions to prefer more clear, more correct, and more engaging outputs. The *Reviser* prompt creates a preference pair by minimally revising and improving an output according to these preferences. Instead, the *Judge* prompt selects a more preferred output given two candidate answers.

7.11.1.2 Preference Pair Filtering

We reject revisions or judgments if the LLM failed to follow formatting guidelines specified in the revising or judging prompt. Additionally, we reject revisions if they altered the length of the original output too much; we found this mainly happens when the LLM misunderstands the revision prompt. Starting from the same 32K instructions sampled from `UltraFeedback`, this procedure creates

Type	Prompt
<i>Reviser</i>	<p>You are a teacher and your task is to minimally improve a student’s answer. I will give you a <code>{{task}}</code> and a <code>{{student_solution}}</code>. Your job is to revise the <code>{{student_solution}}</code> such that it is clearer, more correct, and more engaging. Copy all non-corrected parts of the student’s answer. Do not allude to the <code>{{corrected_student_solution}}</code> being a revision or a correction in your final solution. <code>\n\n{{task}}: <instruction x> \n\n{{student_solution}}: <losing output y_l> \n\n—————\n\nLet’s first think step by step with a <code>{{teacher_reasoning}}</code> to decide how to improve the <code>{{student_solution}}</code>, then give the <code>{{corrected_student_solution}}</code>. Mention the <code>{{teacher_reasoning}}</code> and <code>{{corrected_student_solution}}</code> identifiers to structure your answer. <code>\n\n</code></code></p>
<i>Judge</i>	<p>You are a teacher and your task is to pick the best student’s answer. The best answer is the most clear, most correct, and most engaging answer. I will give you a <code>{{task}}</code> and <code>{{student_solution_1}}</code> and <code>{{student_solution_2}}</code>. Your final answer must contain [1] if <code>{{student_solution_1}}</code> was best, else [2]. <code>\n\n{{task}}: <instruction x> \n\n{{student_solution_1}}: <first output y_1> \n\n{{student_solution_2}}: <second output y_2> \n\n—————\n\nLet’s first think step by step with a <code>{{teacher_reasoning}}</code> to decide which solution is better, and then answer [1] or [2]. <code>\n\n</code></code></p>

Table 7.3: Prompt templates used for creating preference triples (x, y_l, y_w) with the *Reviser* and *Judge* function of Equation 7.1 and 7.2. The variables in the prompt template are **bolded** and bracketed. Both prompts target clear, correct, and engaging outputs. The *Reviser* prompt instructs that a losing output y_l should be minimally improved to create the winning output y_w . Instead, the *Judge* prompt picks the winning/losing output out of two candidates y_1 & y_2 . Both prompts also instruct a model to produce a reasoning before revising or judging.

MixEval-Hard split	# query	score	+ CLAIR	+ Judge (on-policy)	+ Judge (off-policy)	+ Stronger Preferred
Overall score	988	41.45	49.10	46.10	44.15	43.90
TriviaQA	267	34.30	49.20	42.40	43.70	39.80
MMLU	231	43.70	39.00	42.00	36.80	34.60
DROP	167	50.20	58.70	64.30	64.90	58.90
AGIEval	71	31.00	38.00	38.00	39.40	38.00
HellaSwag	61	29.50	37.70	26.20	29.50	27.90
CommonsenseQA	50	60.00	72.00	60.00	48.00	58.00
BoolQ	37	40.50	45.90	32.40	21.60	27.00
GSM8k	22	60.00	80.00	69.50	63.20	84.10
SIQA	20	45.00	50.00	40.00	15.00	40.00
MATH	16	47.50	63.70	51.30	58.80	73.10
BBH	16	51.30	68.80	57.50	60.60	66.90
OpenBookQA	8	62.50	62.50	50.00	62.50	75.00
GPQA	8	12.50	25.00	25.00	25.00	37.50
PIQA	8	50.00	62.50	62.50	62.50	75.00
ARC	4	0.00	0.00	0.00	0.00	0.00
MBPP	2	0.00	0.00	0.00	0.00	0.00
Objective used:		/	APO-zero	APO-zero	APO-down	SFT

Table 7.4: Breakdown of MixEval-Hard performance (version 2024-06-01) in function of which dataset the queries originate from. Analysis given for Llama-3-8B-Instruct and our best models on the CLAIR, Judge (on-policy), Judge (off-policy), and Stronger Preferred datasets. While individual splits may not always indicate the best model (particularly when the amount of queries is low), the overall score correlates highly with human judgments about model performance (Chatbot Arena Elo; Chiang et al. 2024). MixEval-Hard uses a GPT3.5-turbo model to rate if a response to a query agrees with a known gold-truth response.

29K CLAIR pairs, 29K Stronger Preferred pairs, 29K off-policy Judge pairs, and 32k on-policy Judge pairs. We adapted the code by Williams [2023] to efficiently query closed-source LLMs in parallel over API.

7.11.2 MixEval-Hard Performance Breakdown

MixEval-Hard features queries from a wide range of established benchmarks, as outlined in Section 7.6.1. Previously, we reported on the overall MixEval-Hard performance. Table 7.4 breaks down this overall performance in function of these different benchmarks. While MixEval-Hard often incorporates only a few queries from any given benchmark, the overall performance correlates highly with human judgements.

		ME-Hard 2024-06-01		ME-Hard 2024-08-11		
Dataset	Loss	Max Δ	Mean Δ	Max Δ	Mean Δ	Train Time
Judge off-policy	KTO	<u>2.10</u>	<u>-2.70</u> (1.67)	<u>4.75</u>	<u>1.31</u> (1.61)	19h 18m 10s
	APO †	-0.40	-3.67 (1.68)	4.35	0.66 (1.44)	12h 32m 58s
Judge on-policy	KTO	3.50	1.28 (1.11)	4.85	2.70 (1.35)	19h 40m 10s
	APO	4.35	<u>1.31</u> (1.44)	<u>5.60</u>	<u>3.92</u> (0.99)	13h 49m 55s
CLAIR	KTO	<u>3.75</u>	1.47 (1.39)	5.80	4.12 (1.09)	17h 33m 24s
	APO	1.40	-1.49 (1.77)	3.20	1.13 (1.21)	12h 31m 03s
Stronger Preferred	KTO	-3.25	-4.73 (1.01)	0.30	-1.18 (0.75)	19h 07m 29s
	APO	<u>-2.70</u>	<u>-4.57</u> (1.32)	<u>2.95</u>	<u>0.50</u> (1.25)	12h 38m 49s

Table 7.5: Max and mean MixEval-Hard improvements for the 2024-06-01 and 2024-08-11 splits, aggregated over 18 epochs of aligning Llama-3-8B-Instruct. Best overall performance **bold**, best performance per dataset underlined, standard deviation in parentheses. KTO is the best unpaired loss given the off-policy Judge and CLAIR datasets, while APO performs better when given the on-policy Judge and Stronger Preferred datasets. KTO can take 60% longer to train for the same configuration. †In this table, we use the APO-zero-unpaired loss for all APO experiments.

7.11.3 Unpaired APO

In this work, we designed datasets and alignment objectives for paired preferences (output $y_l \prec y_w$ for input x). The original KTO objective [Ethayarajh et al., 2024] was designed to operate on desirability data (output y for input x was desirable or not), which does not use such paired preferences. We consider an unpaired variant of our APO-zero loss, called APO-zero-unpaired, which resembles the KTO objective but which fixes the KL term to zero. Table 7.5 compares KTO with APO-zero-unpaired, keeping everything else comparable with our main results in Table 7.2. To turn our paired datasets into unpaired datasets, we turn each datapoint consisting of two outputs into two datapoints with one output.

There is no clear winner between KTO and APO-zero-unpaired across the board. Within each dataset however, there always is a clear winner. This reflects the main findings of our work, different alignment objectives have distinct semantics, and different datasets require different semantics. APO-zero-unpaired consistently trains faster, due to not calculating the KL term. In some cases, the KTO objective can take 60% longer to train.

7.11.4 How well does AlpacaEval control for lengths?

GPT4 as a judge is known to favor more verbose responses, which can artificially inflate AlpacaEval win rates for verbose models [Dubois et al., 2024]. To coun-

teract this bias, Dubois et al. [2024] estimate a length-controlled `AlpacaEval` win rate, which we report on in Table 7.2. Specifically, the authors adopt a causal inference framework to answer the question "What would the `AlpacaEval` metric be, if the outputs of all models had the same length as those of the baseline?" [Dubois et al., 2024].

In order to meaningfully apply causal inference, a few key assumptions need to be met. The *Positivity* assumption [Hernán and Robins, 2006] states that, when estimating the effect of a treatment, there are at least some subjects which receive the treatment for all covariates. Intuitively, the Positivity assumption applied to the length-control question states that you need to observe at least some long and some short responses for every model in order to accurately estimate how the response length influences the model’s win rate.

The `AlpacaEval` framework does not check if this Positivity assumption is met, potentially giving bad estimates for the length-controlled win rates in some settings. If a certain model consistently generates responses longer than those of the baseline, it is impossible to accurately estimate how good the responses would be if they were as long as the baseline.

This may give us insights into some of our length-controlled `AlpacaEval` win rates. For example, the SFT result on the Stronger Preferred dataset in Table 7.2 seems disproportionately high in comparison to the `MixEval-Hard` results for that same experiment. This model is considerably more verbose than `Llama-3-8B-Instruct`, as evident from the large response length increase associated with this experiment (+ 1883 characters on average). It is possible the Positivity constraint was not met for this experiment, causing the length-controlled framework of `AlpacaEval` to provide inaccurate estimates.

While a more thorough study of length-controlled win rate is out of scope for this work, one potential avenue towards a more robust length-controlled win rate would be to specifically prompt models to generate shorter or longer answers if the Positivity constraint is not met.

References

- Eldar David Abraham, Karel D’Oosterlinck, Amir Feder, Yair Ori Gat, Atticus Geiger, Christopher Potts, Roi Reichart, and Zhengxuan Wu. CEBaB: Estimating the causal effects of real-world concepts on NLP model behavior. *Advances in Neural Information Processing Systems*, 2022. URL <https://arxiv.org/abs/2205.14140>.
- Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.
- Leonard Adolphs, Tianyu Gao, Jing Xu, Kurt Shuster, Sainbayar Sukhbaatar, and Jason Weston. The CRINGE loss: Learning what language not to model. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8854–8874, 2023.
- Afra Amini, Tim Vieira, and Ryan Cotterell. Direct preference optimization with an offset. *arXiv preprint arXiv:2402.10571*, 2024.
- Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete Problems in AI Safety. *arXiv:1606.06565 [cs]*, 2016. URL <http://arxiv.org/abs/1606.06565>.
- Isaac Asimov. *I, robot*, volume 1. Spectra, 2004.
- Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, et al. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021.
- Mohammad Gheshlaghi Azar, Zhaohan Daniel Guo, Bilal Piot, Remi Munos, Mark Rowland, Michal Valko, and Daniele Calandriello. A general theoretical paradigm to understand learning from human preferences. In *International Conference on Artificial Intelligence and Statistics*, pages 4447–4455. PMLR, 2024.
- Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. Constitutional AI: Harmlessness from AI feedback. *arXiv preprint arXiv:2212.08073*, 2022.

- Pangbo Ban, Yifan Jiang, Tianran Liu, and Shane Steinert-Threlkeld. Testing pre-trained language models' understanding of distributivity via causal mediation analysis. arXiv:2209.04761, 2022. URL <https://arxiv.org/abs/2209.04761>.
- Jasmijn Bastings, Yonatan Belinkov, Emmanuel Dupoux, Mario Giulianelli, Dieuwke Hupkes, Yuval Pinter, and Hassan Sajjad. Proceedings of the fourth BlackboxNLP workshop on analyzing and interpreting neural networks for NLP. In *Proceedings of the Fourth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, 2021.
- Leonard Bereska and Efstratios Gavves. Mechanistic interpretability for ai safety—a review. *arXiv preprint arXiv:2404.14082*, 2024.
- Steven Bills, Nick Cammarata, Dan Mossing, Henk Tillman, Leo Gao, Gabriel Goh, Ilya Sutskever, Jan Leike, Jeff Wu, and William Saunders. Language models can explain neurons in language models. URL <https://openaipublic.blob.core.windows.net/neuron-explainer/paper/index.html>. (Date accessed: 14.05. 2023), 2, 2023.
- Alexander Binder, Grégoire Montavon, Sebastian Lapuschkin, Klaus-Robert Müller, and Wojciech Samek. Layer-wise relevance propagation for neural networks with local renormalization layers. In *International Conference on Artificial Neural Networks*, pages 63–71. Springer, 2016.
- Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. PIQA: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 7432–7439, 2020.
- Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.
- Nick Bostrom. *Superintelligence: Paths, Dangers, Strategies*. Oxford University Press, Oxford, 2014. ISBN 9780199678112. An AI, designed to manage production in a factory, is given the final goal of maximizing the manufacturing of paperclips, and proceeds by converting first the Earth and then increasingly large chunks of the observable universe into paperclips.
- Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nicholas L. Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Alex Tamkin, Karina Nguyen, Brayden McLean, Josiah E. Burke, Tristan Hume, Shan Carter, Tom Henighan, and Chris Olah. Towards monosemanticity: Decomposing language models with dictionary learning, 2023. URL <https://transformer-circuits.pub/2023/monosemantic-features>. Accessed: 2024-12-07.

-
- Tom B Brown. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.
- Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language Models are Few-Shot Learners. *arXiv:2005.14165 [cs]*, July 2020. URL <http://arxiv.org/abs/2005.14165>. arXiv: 2005.14165 version: 4.
- Nitay Calderon, Eyal Ben-David, Amir Feder, and Roi Reichart. Docogen: Domain counterfactual generation for low resource domain adaptation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7727–7746, 2022.
- Center for AI Safety. About us, 2024. URL <https://www.safe.ai/about>. Accessed: 2024-12-07.
- Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, Banghua Zhu, Hao Zhang, Michael Jordan, Joseph E Gonzalez, et al. Chatbot arena: An open platform for evaluating LLMs by human preference. In *Forty-first International Conference on Machine Learning*, 2024.
- Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 4302–4310, 2017.
- Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53, 2024.
- Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2924–2936, 2019a.
- Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. What does BERT look at? An analysis of BERT’s attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks*

for NLP, pages 276–286, Florence, Italy, August 2019b. Association for Computational Linguistics. URL <https://www.aclweb.org/anthology/W19-4828>.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try ARC, the AI2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Alexis Conneau, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. What you can cram into a single $\&\&! \#^*$ vector: Probing sentence embeddings for linguistic properties. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2126–2136, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1198. URL <https://www.aclweb.org/anthology/P18-1198>.

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Bingxiang He, Wei Zhu, Yuan Ni, Guotong Xie, Ruobing Xie, Yankai Lin, et al. ULTRAFEEDBACK: Boosting language models with scaled AI feedback. In *Forty-first International Conference on Machine Learning*, 2024.

Nicola De Cao, Leon Schmid, Dieuwke Hupkes, and Ivan Titov. Sparse interventions in language models with differentiable masking. arxiv:2112.06837, 2021. URL <https://arxiv.org/abs/2112.06837>.

René Descartes. *The philosophical writings of Descartes*, volume 2. Cambridge University Press, 1984.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL <https://www.aclweb.org/anthology/N19-1423>.

Jay DeYoung, Sarthak Jain, Nazneen Fatema Rajani, Eric Lehman, Caiming Xiong, Richard Socher, and Byron C. Wallace. ERASER: A benchmark to evaluate rationalized NLP models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4443–4458, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.408. URL <https://aclanthology.org/2020.acl-main.408>.

-
- Doe v. GitHub, 2022. Doe 1 et al. v. github, inc., et al. Class action complaint, 2022. URL <https://githubcopilotlitigation.com>. Case No. 4:22-cv-06823-KAW, U.S. District Court for the Northern District of California, filed November 3, 2022.
- Karel D’Oosterlinck, Winnie Xu, Chris Develder, Thomas Demeester, Amanpreet Singh, Christopher Potts, Douwe Kiela, and Shikib Mehri. Anchored preference optimization and contrastive revisions: Addressing underspecification in alignment. *arXiv preprint arXiv:2408.06266*, 2024.
- Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2368–2378, 2019.
- Shitong Duan, Xiaoyuan Yi, Peng Zhang, Tun Lu, Xing Xie, and Ning Gu. Negating negatives: Alignment without human positive samples via distributional dispreference optimization. *arXiv preprint arXiv:2403.03419*, 2024.
- Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.
- Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B Hashimoto. Length-controlled alpacaeval: A simple way to debias automatic evaluators. *arXiv preprint arXiv:2404.04475*, 2024.
- Upol Ehsan, Q Vera Liao, Michael Muller, Mark O Riedl, and Justin D Weisz. Expanding explainability: Towards social transparency in AI systems. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, 2021. URL <https://dl.acm.org/doi/pdf/10.1145/3411764.3445188>.
- Jacob Eisenstein, Chirag Nagpal, Alekh Agarwal, Ahmad Beirami, Alex D’Amour, DJ Dvijotham, Adam Fisch, Katherine Heller, Stephen Pfohl, Deepak Ramachandran, et al. Helping or herding? reward model ensembles mitigate but do not eliminate reward hacking. *arXiv preprint arXiv:2312.09244*, 2023.
- Yanai Elazar, Shauli Ravfogel, Alon Jacovi, and Yoav Goldberg. Amnesic probing: Behavioral explanation with amnesic counterfactuals. *Transactions of the Association for Computational Linguistics*, 9:160–175, 2021.
- Tyna Eloundou, Sam Manning, Pamela Mishkin, and Daniel Rock. Gpts are gpts: An early look at the labor market impact potential of large language models. *arXiv preprint arXiv:2303.10130*, 2023.

- Kawin Ethayarajh, Yejin Choi, and Swabha Swayamdipta. Understanding dataset difficulty with \mathcal{V} -usable information. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 5988–6008. PMLR, 17–23 Jul 2022. URL <https://proceedings.mlr.press/v162/ethayarajh22a.html>.
- Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. KTO: Model alignment as prospect theoretic optimization. *arXiv preprint arXiv:2402.01306*, 2024.
- Amir Feder, Katherine A Keith, Emaad Manzoor, Reid Pryzant, Dhanya Sridhar, Zach Wood-Doughty, Jacob Eisenstein, Justin Grimmer, Roi Reichart, Margaret E Roberts, et al. Causal inference in natural language processing: Estimation, prediction, interpretation and beyond. *arXiv preprint arXiv:2109.00725*, 2021a.
- Amir Feder, Nadav Oved, Uri Shalit, and Roi Reichart. CausaLM: Causal model explanation through counterfactual language models. *Computational Linguistics*, 47(2):333–386, June 2021b. doi: 10.1162/coli_a_00404. URL <https://aclanthology.org/2021.cl-2.13>.
- Duanyu Feng, Bowen Qin, Chen Huang, Zheng Zhang, and Wenqiang Lei. Towards analyzing and understanding the limitations of DPO: A theoretical perspective. *arXiv preprint arXiv:2404.04626*, 2024.
- Matthew Finlayson, Aaron Mueller, Sebastian Gehrmann, Stuart Shieber, Tal Linzen, and Yonatan 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)*, pages 1828–1843, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.144. URL <https://aclanthology.org/2021.acl-long.144>.
- Iason Gabriel. Artificial intelligence, values, and alignment. *Minds and machines*, 30(3):411–437, 2020.
- Deep Ganguli, Liane Lovitt, Jackson Kernion, Amanda Askell, Yuntao Bai, Saurav Kadavath, Ben Mann, Ethan Perez, Nicholas Schiefer, Kamal Ndousse, et al. Red teaming language models to reduce harms: Methods, scaling behaviors, and lessons learned. *arXiv preprint arXiv:2209.07858*, 2022.
- Leo Gao, John Schulman, and Jacob Hilton. Scaling laws for reward model overoptimization. In *International Conference on Machine Learning*, pages 10835–10866. PMLR, 2023.

-
- Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Simas Sakenis, Jason Huang, Yaron Singer, and Stuart Shieber. Causal Mediation Analysis for Interpreting Neural NLP: The Case of Gender Bias. *arXiv:2004.12265 [cs]*, November 2020. URL <http://arxiv.org/abs/2004.12265>. arXiv: 2004.12265.
- Atticus Geiger, Kyle Richardson, and Christopher Potts. Neural natural language inference models partially embed theories of lexical entailment and negation. In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 163–173, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.blackboxnlp-1.16. URL <https://aclanthology.org/2020.blackboxnlp-1.16>.
- Atticus Geiger, Hanson Lu, Thomas Icard, and Christopher Potts. Causal abstractions of neural networks. *Advances in Neural Information Processing Systems*, 34, 2021.
- Atticus Geiger, Zhengxuan Wu, Hanson Lu, Josh Rozner, Elisa Kreiss, Thomas Icard, Noah Goodman, and Christopher Potts. Inducing causal structure for interpretable neural networks. In *International Conference on Machine Learning*, 2022. URL <https://proceedings.mlr.press/v162/geiger22a.html>.
- Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020.
- Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.
- Bryce Goodman and Seth Flaxman. European Union regulations on algorithmic decision-making and a "right to explanation". *AI Magazine*, 38(3):50–57, October 2017. ISSN 2371-9621, 0738-4602. doi: 10.1609/aimag.v38i3.2741. URL <http://arxiv.org/abs/1606.08813>. arXiv: 1606.08813.
- Yash Goyal, Ziyang Wu, Jan Ernst, Dhruv Batra, Devi Parikh, and Stefan Lee. Counterfactual visual explanations. In *International Conference on Machine Learning*, 2019. URL <http://proceedings.mlr.press/v97/goyal19a.html>.
- Yash Goyal, Amir Feder, Uri Shalit, and Been Kim. Explaining Classifiers with Causal Concept Effect (CaCE). *arXiv:1907.07165 [cs, stat]*, February 2020. URL <http://arxiv.org/abs/1907.07165>. arXiv: 1907.07165.
- Dirk Groeneveld, Iz Beltagy, Pete Walsh, Akshita Bhagia, Rodney Kinney, Oyvind Tafjord, Ananya Harsh Jha, Hamish Ivison, Ian Magnusson, Yizhong Wang, et al. Olmo: Accelerating the science of language models. *arXiv preprint arXiv:2402.00838*, 2024.

- Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, et al. A survey on llm-as-a-judge. *arXiv preprint arXiv:2411.15594*, 2024.
- Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Gian-notti, and Dino Pedreschi. A survey of methods for explaining black box models. *ACM computing surveys (CSUR)*, 51(5):1–42, 2018.
- Moritz Hardt, Eric Price, Eric Price, and Nati Srebro. Equality of Opportunity in Supervised Learning. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL <https://proceedings.neurips.cc/paper/2016/file/9d2682367c3935defcblf9e247a97c0d-Paper.pdf>.
- Peter Hase and Mohit Bansal. Evaluating explainable AI: Which algorithmic explanations help users predict model behavior? *arXiv preprint arXiv:2005.01831*, 2020.
- Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understand-ing. In *International Conference on Learning Representations*, 2020.
- Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.
- Miguel A Hernán and James M Robins. Estimating causal effects from epidemi-ological data. *Journal of Epidemiology & Community Health*, 60(7):578–586, 2006.
- Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. Distilling the knowledge in a neural network. *NeurIPS Deep Learning and Representation Learning Workshop*, 2015. URL <https://arxiv.org/abs/1503.02531>.
- Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Com-putation*, 1997. URL <https://ieeexplore.ieee.org/abstract/document/6795963>.
- Paul W Holland. Statistics and causal inference. *Journal of the American statistical Association*, 1986. URL <https://www.tandfonline.com/doi/abs/10.1080/01621459.1986.10478354>.
- Jiwoo Hong, Noah Lee, and James Thorne. Reference-free monolithic preference optimization with odds ratio. *arXiv preprint arXiv:2403.07691*, 2024.

-
- Sara Hooker, Dumitru Erhan, Pieter-Jan Kindermans, and Been Kim. A benchmark for interpretability methods in deep neural networks. *Advances in neural information processing systems*, 32, 2019.
- Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. Xtreme: A massively multilingual multi-task benchmark for evaluating cross-lingual generalisation. In *International Conference on Machine Learning*, pages 4411–4421. PMLR, 2020.
- Jing Huang, Atticus Geiger, Karel D’Oosterlinck, Zhengxuan Wu, and Christopher Potts. Rigorously assessing natural language explanations of neurons. In *Proceedings of the 6th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 317–331, 2023.
- Ahmed Hussein, Mohamed Medhat Gaber, Eyad Elyan, and Chrisina Jayne. Imitation learning: A survey of learning methods. *ACM Computing Surveys (CSUR)*, 50(2):1–35, 2017.
- Guido W Imbens. Matching methods in practice: Three examples. *Journal of Human Resources*, 50(2):373–419, 2015.
- Alon Jacovi and Yoav Goldberg. Towards faithfully interpretable NLP systems: How should we define and evaluate faithfulness? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4198–4205, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.386. URL <https://aclanthology.org/2020.acl-main.386>.
- Maurice Jakesch, Megan French, Xiao Ma, Jeffrey T Hancock, and Mor Naaman. AI-mediated communication: How the perception that profile text was written by AI affects trustworthiness. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, 2019. URL <https://dl.acm.org/doi/abs/10.1145/3290605.3300469>.
- Dongfu Jiang, Xiang Ren, and Bill Yuchen Lin. LLM-Blender: Ensembling large language models with pairwise ranking and generative fusion. In *The 61st Annual Meeting Of The Association For Computational Linguistics*, 2023.
- Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, 2017.
- Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, H erve J egou, and Tomas Mikolov. Fasttext.zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*, 2016.

- John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with alphafold. *nature*, 596(7873):583–589, 2021.
- Seungjae Jung, Gunsoo Han, Daniel Wontae Nam, and Kyoung-Woon On. Binary classifier optimization for large language model alignment. *arXiv preprint arXiv:2404.04656*, 2024.
- Douwe Kiela, Max Bartolo, Yixin Nie, Divyansh Kaushik, Atticus Geiger, Zhengxuan Wu, Bertie Vidgen, Grusha Prasad, Amanpreet Singh, Pratik Ringshia, Zhiyi Ma, Tristan Thrush, Sebastian Riedel, Zeerak Waseem, Pontus Stenertorp, Robin Jia, Mohit Bansal, Christopher Potts, and Adina Williams. Dynabench: Rethinking benchmarking in NLP. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4110–4124, Online, June 2021. Association for Computational Linguistics. URL <https://www.aclweb.org/anthology/2021.naacl-main.324>.
- Been Kim. *Interactive and Interpretable Machine Learning Models for Human Machine Collaboration*. PhD thesis, Massachusetts Institute of Technology, 2015. URL <https://dspace.mit.edu/handle/1721.1/98680>.
- Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, and Rory Sayres. Interpretability Beyond Feature Attribution: Quantitative Testing with Concept Activation Vectors (TCAV). In *International Conference on Machine Learning*, pages 2668–2677. PMLR, July 2018. URL <http://proceedings.mlr.press/v80/kim18d.html>. ISSN: 2640-3498.
- Dahyun Kim, Yungi Kim, Wonho Song, Hyeonwoo Kim, Yunsu Kim, Sanghoon Kim, and Chanjun Park. sDPO: Don’t use your data all at once. *arXiv preprint arXiv:2403.19270*, 2024.
- Jon Kleinberg, Sendhil Mullainathan, and Manish Raghavan. Inherent trade-offs in the fair determination of risk scores. In *8th Innovations in Theoretical Computer Science Conference*, 2017. URL <http://drops.dagstuhl.de/opus/volltexte/2017/8156>.
- Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. In *International Conference on Machine Learning*, 2020. URL <https://proceedings.mlr.press/v119/koh20a.html>.
- Sören R Künzel, Jasjeet S Sekhon, Peter J Bickel, and Bin Yu. Metalearners for estimating heterogeneous treatment effects using machine learning. *Proceedings of the national academy of sciences*, 116(10):4156–4165, 2019.

-
- Nathan Lambert, Hailey Schoelkopf, Aaron Gokaslan, Luca Soldaini, Valentina Pyatkin, and Louis Castricato. Self-directed synthetic dialogues and revisions technical report. *arXiv preprint arXiv:2407.18421*, 2024.
- Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.
- Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. AlpacaEval: An automatic evaluator of instruction-following models. https://github.com/tatsu-lab/alpaca_eval, 5 2023.
- Yujia Li, David Choi, Junyoung Chung, Nate Kushman, Julian Schrittwieser, Rémi Leblond, Tom Eccles, James Keeling, Felix Gimeno, Agustin Dal Lago, et al. Competition-level code generation with alphacode. *Science*, 378(6624):1092–1097, 2022.
- Zachary C. Lipton. The mythos of model interpretability. *Communications of the ACM*, 61(10):36–43, September 2018. ISSN 0001-0782. doi: 10.1145/3233231. URL <https://doi.org/10.1145/3233231>.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv:1907.11692 [cs]*, July 2019. URL <http://arxiv.org/abs/1907.11692>. arXiv: 1907.11692.
- Charles Lovering and Ellie Pavlick. Unit testing for concepts in neural networks. *arXiv preprint arXiv:2208.10244*, 2022.
- Scott M. Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS’17*, pages 4768–4777, Red Hook, NY, USA, December 2017. Curran Associates Inc. ISBN 978-1-5108-6096-4.
- Thang Luong, Hieu Pham, and Christopher D Manning. Effective approaches to attention-based neural machine translation. In *EMNLP*, 2015.
- Christopher D. Manning, Kevin Clark, John Hewitt, Urvashi Khandelwal, and Omer Levy. Emergent linguistic structure in artificial neural networks trained by self-supervision. *Proceedings of the National Academy of Sciences*, 117(48):30046–30054, 2020. ISSN 0027-8424. doi: 10.1073/pnas.1907367117. URL <https://www.pnas.org/content/117/48/30046>.
- Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. A survey on bias and fairness in machine learning. *ACM Computing Surveys (CSUR)*, 2021. URL <https://dl.acm.org/doi/abs/10.1145/3457607>.

Yu Meng, Mengzhou Xia, and Danqi Chen. SimPO: Simple preference optimization with a reference-free reward. *arXiv preprint arXiv:2405.14734*, 2024.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2381–2391, 2018.

Azalia Mirhoseini, Anna Goldie, Mustafa Yazgan, Joe Wenjie Jiang, Ebrahim Songhori, Shen Wang, Young-Joon Lee, Eric Johnson, Omkar Pathak, Azade Nazi, et al. A graph placement methodology for fast chip design. *Nature*, 594 (7862):207–212, 2021.

Christoph Molnar. *Interpretable machine learning*. Lulu. com, 2020.

Andrew Ng et al. Sparse autoencoder. *CS294A Lecture notes*, 72(2011):1–19, 2011.

Jinjie Ni, Fuzhao Xue, Xiang Yue, Yuntian Deng, Mahir Shah, Kabir Jain, Graham Neubig, and Yang You. MixEval: Deriving wisdom of the crowd from LLM benchmark mixtures. *arXiv preprint arXiv:2406.06565*, 2024.

Clemens Otte. Safe and Interpretable Machine Learning: A Methodological Review. In Christian Moewes and Andreas Nürnberger, editors, *Computational Intelligence in Intelligent Data Analysis*, Studies in Computational Intelligence, pages 111–122, Berlin, Heidelberg, 2013. Springer. ISBN 978-3-642-32378-2. doi: 10.1007/978-3-642-32378-2_8.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

Arka Pal, Deep Karkhanis, Samuel Dooley, Manley Roberts, Siddartha Naidu, and Colin White. Smaug: Fixing failure modes of preference optimisation with DPO-positive. *arXiv preprint arXiv:2402.13228*, 2024.

Ryan Park, Rafael Rafailov, Stefano Ermon, and Chelsea Finn. Disentangling length from quality in direct preference optimization. *arXiv preprint arXiv:2403.19159*, 2024.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, 2019.

J. Pearl. *Causality*. Causality: Models, Reasoning, and Inference. Cambridge University Press, 2009a. ISBN 978-0-521-89560-6. URL <https://books.google.co.il/books?id=f4nuexsNVZIC>.

-
- Judea Pearl. Causal diagrams for empirical research. *Biometrika*, 82(4):669–688, 1995.
- Judea Pearl. *Causality*. Cambridge university press, 2009b.
- Judea Pearl. The limitations of opaque learning machines. *Possible minds: twenty-five ways of looking at AI*, pages 13–19, 2019.
- Jeffrey Pennington, Richard Socher, and Christopher Manning. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014. URL <https://aclanthology.org/D14-1162>.
- Shiva Kumar Pentylala, Zhichao Wang, Bin Bi, Kiran Ramnath, Xiang-Bo Mao, Regunathan Radhakrishnan, Sitaram Asur, et al. PAFT: A parallel training paradigm for effective LLM fine-tuning. *arXiv preprint arXiv:2406.17923*, 2024.
- Danish Pruthi, Rachit Bansal, Bhuwan Dhingra, Livio Baldini Soares, Michael Collins, Zachary C Lipton, Graham Neubig, and William W Cohen. Evaluating explanations: How much do explanations from the teacher aid students? *Transactions of the Association for Computational Linguistics*, 10:359–375, 2022.
- Alec Radford and Karthik Narasimhan. Improving language understanding by generative pre-training. 2018. URL <https://api.semanticscholar.org/CorpusID:49313245>.
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.
- Rafael Rafailov, Yaswanth Chittetu, Ryan Park, Harshit Sikchi, Joey Hejna, Bradley Knox, Chelsea Finn, and Scott Niekum. Scaling laws for reward model overoptimization in direct alignment algorithms. *arXiv preprint arXiv:2406.02900*, 2024a.
- Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36, 2024b.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*, 2019.
- Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International conference on machine learning*, pages 8821–8831. Pmlr, 2021.

- Shauli Ravfogel, Yanai Elazar, Hila Gonen, Michael Twiton, and Yoav Goldberg. Null It Out: Guarding Protected Attributes by Iterative Nullspace Projection. *arXiv:2004.07667 [cs]*, April 2020. URL <http://arxiv.org/abs/2004.07667>. arXiv: 2004.07667.
- David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. GPQA: A graduate-level google-proof q&a benchmark. *arXiv preprint arXiv:2311.12022*, 2023.
- Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1135–1144, San Francisco California USA, August 2016. ACM. ISBN 978-1-4503-4232-2. doi: 10.1145/2939672.2939778. URL <https://dl.acm.org/doi/10.1145/2939672.2939778>.
- Pierre Harvey Richemond, Yunhao Tang, Daniel Guo, Daniele Calandriello, Mohammad Gheshlaghi Azar, Rafael Rafailov, Bernardo Avila Pires, Eugene Tarassov, Lucas Spangher, Will Ellsworth, et al. Offline regularised reinforcement learning for large language models alignment. *arXiv preprint arXiv:2405.19107*, 2024.
- Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.
- Corby Rosset, Ching-An Cheng, Arindam Mitra, Michael Santacroce, Ahmed Awadallah, and Tengyang Xie. Direct nash optimization: Teaching language models to self-improve with general preferences. *arXiv preprint arXiv:2404.03715*, 2024.
- D. Rubin. Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of Educational Psychology*, 66:688–701, 1974.
- David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.
- Keita Saito, Akifumi Wachi, Koki Wataoka, and Youhei Akimoto. Verbosity bias in preference labeling by large language models. In *NeurIPS 2023 Workshop on Instruction Tuning and Instruction Following*, 2023.
- Wojciech Samek, Grégoire Montavon, Sebastian Lapuschkin, Christopher J Anders, and Klaus-Robert Müller. Explaining deep neural networks and beyond: A review of methods and applications. *Proceedings of the IEEE*, 109(3):247–278, 2021.
- Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. Social IQa: Commonsense reasoning about social interactions. 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), pages 4463–4473, 2019.

- Naomi Saphra and Adam Lopez. Understanding learning dynamics of language models with SVCCA. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3257–3267, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1329. URL <https://www.aclweb.org/anthology/N19-1329>.
- John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.
- Lloyd Shapley. A value for n-person games. In H. W. Kuhn and A. W. Tucker, editors, *Contributions to the Theory of Games II*, pages 307–317. Princeton University Press, 1953.
- Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *International conference on machine learning*, pages 3145–3153. PMLR, 2017.
- Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, et al. Make-a-video: Text-to-video generation without text-video data. In *The Eleventh International Conference on Learning Representations*.
- Paul Soulos, R. Thomas McCoy, Tal Linzen, and Paul Smolensky. Discovering the compositional structure of vector representations with role learning networks. In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 238–254, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.blackboxnlp-1.23. URL <https://aclanthology.org/2020.blackboxnlp-1.23>.
- Jost Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net. *CoRR*, 12 2014.
- Chi Sun, Luyao Huang, and Xipeng Qiu. Utilizing BERT for aspect-based sentiment analysis via constructing auxiliary sentence. *arXiv preprint arXiv:1903.09588*, 2019.
- Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML’17, page 3319–3328. JMLR.org, 2017.
- Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems-Volume 2*, pages 3104–3112, 2014.

- Richard Sutton. The bitter lesson. *Incomplete Ideas (blog)*, 13(1):38, 2019.
- Richard S Sutton and Andrew G Barto. Reinforcement learning: An introduction. *Robotica*, 17(2):229–235, 1999.
- Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc Le, Ed Chi, Denny Zhou, et al. Challenging BIG-bench tasks and whether chain-of-thought can solve them. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13003–13051, 2023.
- Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158, 2019.
- Ian Tenney, Dipanjan Das, and Ellie Pavlick. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, Florence, Italy, July 2019. Association for Computational Linguistics. URL <https://www.aclweb.org/anthology/P19-1452>.
- Ian Tenney, James Wexler, Jasmijn Bastings, Tolga Bolukbasi, Andy Coenen, Sebastian Gehrmann, Ellen Jiang, Mahima Pushkarna, Carey Radebaugh, Emily Reif, and Ann Yuan. 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*, pages 107–118, Online, October 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-demos.15. URL <https://aclanthology.org/2020.emnlp-demos.15>.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems 30.*, 2017. URL <http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf>.
- Sahil Verma, John Dickerson, and Keegan Hines. Counterfactual explanations for machine learning: A review. , 2020. URL <https://arxiv.org/abs/2010.10596>.
- Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart M. Shieber. Investigating gender bias in language models using causal mediation analysis. In Hugo Larochelle, Marc’Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural*

Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual, 2020. URL <https://proceedings.neurips.cc/paper/2020/hash/92650b2e92217715fe312e6fa7b90d82-Abstract.html>.

Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan Lambert, and Shengyi Huang. TRL: Transformer reinforcement learning. <https://github.com/huggingface/trl>, 2020.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium, 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-5446. URL <http://aclweb.org/anthology/W18-5446>.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. Superglue: A stickier benchmark for general-purpose language understanding systems. *Advances in neural information processing systems*, 32, 2019.

Rose E Wang, Ana T Ribeiro, Carly D Robinson, Susanna Loeb, and Dora Demszky. Tutor copilot: A human-ai approach for scaling real-time expertise. *arXiv preprint arXiv:2410.03017*, 2024a.

Zhichao Wang, Bin Bi, Shiva Kumar Pentylala, Kiran Ramnath, Sougata Chaudhuri, Shubham Mehrotra, Xiang-Bo Mao, Sitaram Asur, et al. A comprehensive survey of LLM alignment techniques: RLHF, RLAI, PPO, DPO and more. *arXiv preprint arXiv:2407.16216*, 2024b.

Becca Williams. Parallel process GPT. https://github.com/tiny-rawr/parallel_process_gpt, 2023.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Huggingface’s transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*, 2019.

Junkang Wu, Yuexiang Xie, Zhengyi Yang, Jiancan Wu, Jinyang Gao, Bolin Ding, Xiang Wang, and Xiangnan He. β -DPO: Direct preference optimization with dynamic β . *arXiv preprint arXiv:2407.08639*, 2024a.

Tongshuang Wu, Marco Tulio Ribeiro, Jeffrey Heer, and Daniel Weld. Polyjuice: Generating counterfactuals for explaining, evaluating, and improving models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, 2021.

- Yue Wu, Zhiqing Sun, Huizhuo Yuan, Kaixuan Ji, Yiming Yang, and Quanquan Gu. Self-play preference optimization for language model alignment. *arXiv preprint arXiv:2405.00675*, 2024b.
- Zhengxuan Wu, Karel D’Oosterlinck, Atticus Geiger, Amir Zur, and Christopher Potts. Causal proxy models for concept-based model explanations. In *International conference on machine learning*, pages 37313–37334. PMLR, 2023.
- Haoran Xu, Amr Sharaf, Yunmo Chen, Weiting Tan, Lingfeng Shen, Benjamin Van Durme, Kenton Murray, and Young Jin Kim. Contrastive preference optimization: Pushing the boundaries of LLM performance in machine translation. In *Forty-first International Conference on Machine Learning*, 2024.
- Bayya Yegnanarayana. *Artificial neural networks*. PHI Learning Pvt. Ltd., 2009.
- Chih-Kuan Yeh, Been Kim, Serkan Arik, Chun-Liang Li, Tomas Pfister, and Pradeep Ravikumar. On completeness-aware concept-based explanations in deep neural networks. *Advances in Neural Information Processing Systems*, 33:20554–20565, 2020.
- Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Xian Li, Sainbayar Sukhbaatar, Jing Xu, and Jason E Weston. Self-rewarding language models. In *Forty-first International Conference on Machine Learning*, 2024.
- Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 818–833, Cham, 2014. Springer International Publishing.
- Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hel-laSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, 2019.
- Ruiqi Zhang, Licong Lin, Yu Bai, and Song Mei. Negative preference optimization: From catastrophic collapse to effective unlearning. *arXiv preprint arXiv:2404.05868*, 2024.
- Yao Zhao, Rishabh Joshi, Tianqi Liu, Misha Khalman, Mohammad Saleh, and Peter J Liu. Slic-hf: Sequence likelihood calibration with human feedback. *arXiv preprint arXiv:2305.10425*, 2023.
- Wanjun Zhong, Ruixiang Cui, Yiduo Guo, Yaobo Liang, Shuai Lu, Yanlin Wang, Amin Saied, Weizhu Chen, and Nan Duan. AGIEval: A human-centric benchmark for evaluating foundation models. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 2299–2314, 2024.
- Banghua Zhu, Evan Frick, Tianhao Wu, Hanlin Zhu, and Jiantao Jiao. Starling-7B: Improving LLM helpfulness & harmlessness with RLAIIF, November 2023.