

CAUSE: POST-HOC NATURAL LANGUAGE EXPLANATION OF MULTIMODAL CLASSIFIERS THROUGH CAUSAL ABSTRACTION

Anonymous authors

Paper under double-blind review

ABSTRACT

The increasing integration of AI models in critical areas, such as healthcare, finance, and security has raised concerns about their “black-box” nature, limiting trust and accountability. To ensure robust and trustworthy AI, interpretability is essential. In this paper, we propose **CAuSE (Causal Abstraction under Simulated Explanation)**, a novel framework for post-hoc explanation of multimodal classifiers. Unlike existing interpretability methods, such as Amnesic Probing and Integrated Gradients, CAuSE generates causally faithful natural language explanations of fine-tuned multimodal classifiers’ decisions. CAuSE integrates Interchange Intervention Training (IIT) within a Language Model (LM) based module to simulate the causal reasoning behind a classifier’s outputs. We introduce a novel metric *Counterfactual F1 score* to measure causal faithfulness and demonstrate that CAuSE achieves state-of-the-art performance on this metric. We also provide a rigorous theoretical underpinning for causal abstraction between two neural networks and implement this within our CAuSE framework. This ensures that CAuSE’s natural language explanations are not only simulations of the classifier’s behavior but also reflect its underlying causal processes. Our method is task-agnostic and achieves state-of-the-art results on benchmark multimodal classification datasets, such as e-SNLI-VE and Facebook Hateful Memes, offering a scalable, faithful solution for interpretability in multimodal classifiers.

1 INTRODUCTION

With the rise of Visual Language Models (VLMs), AI systems have evolved to handle multiple data types like images, text, and audio. Multimodal classifiers, central to this advancement, are crucial in applications such as healthcare, where they combine medical images and patient data to improve diagnostic accuracy for diseases like COVID-19 and Alzheimer’s (Baltrušaitis et al., 2017). Similarly, in autonomous driving, they enhance decision-making by integrating visual, LiDAR, and radar inputs (Xiao et al., 2022). These classifiers boost performance by leveraging diverse modalities, making them vital in real-world scenarios.

However, as multimodal classifiers grow in complexity, the need for interpretability becomes paramount. Current interpretability methods, such as Integrated Gradients (Sundararajan et al., 2017a), are designed to highlight explicit input features but fall short of capturing the implicit causal relationships that often drive the decisions of these models. While some techniques, like CausaLM (Feder et al., 2022) and Amnesic Probing (Elazar et al., 2021), aim to incorporate causal mechanisms for interpretability, they struggle with scalability. Other methods, such as Semantify (Bandyopadhyay et al., 2024), manage implicit concepts efficiently but are restricted to specific use cases and fail to generate comprehensive natural language explanations.

To address these limitations, large Visual Language Models (VLMs) have been utilized to generate natural language explanations for decisions made by visual-text multimodal classifiers. However, these models often inject their own biases and opinions, leading to explanations that are inconsistent or detached from the actual workings of the classifier (Agarwal et al., 2024). Recent studies (Madsen et al., 2024) have highlighted these faithfulness issues, revealing inconsistencies when models are further probed.

In this paper, we introduce **CAuSE (Causal Abstraction under Simulated Explanation)**, a novel framework designed to generate faithful natural language explanations for the decisions of a pre-trained classifier, offering post-hoc interpretability. CAuSE combines Interchange Intervention Training (Geiger et al., 2021a) with Language Model (LM)-based modules, ensuring that the generated explanations are both causally accurate and reflective of the classifier’s internal decision-making process. We introduce a new metric, the *Counterfactual F1 score*, to assess the causal faithfulness of explanations. CAuSE sets a new benchmark on this metric, achieving state-of-the-art performance. Through case studies, we showcase successful generations from our framework and conduct error analysis to identify common mistakes and their underlying causes.

Our framework is task-agnostic and demonstrates state-of-the-art performance on benchmark datasets, such as e-SNLI-VE (Do et al., 2021) and Facebook Hateful Memes (Kiela et al., 2021), providing robust, faithful explanations across diverse multimodal tasks. The codes are available at <https://anonymous.4open.science/r/CAuSE-5BD0>.

2 ARCHITECTURE

Our framework, CAuSE, generates faithful natural language explanations for decisions made by a pre-trained multimodal classifier (called the **post-hoc classifier**). As detailed in Section 3.2, CAuSE acts as a causal abstraction of the post-hoc classifier, ensuring its explanations are rooted in the actual decision-making process. This is supported by the high Counterfactual F1 scores CAuSE achieves compared to the other ablated components, as shown in Table 2. This section introduces the post-hoc classifier and provides a detailed description of the CAuSE framework, with a working diagram of both presented in Figure 1.

2.1 POST-HOC CLASSIFIER

The post-hoc classifier is assumed to be composed of a multimodal encoder E and a feed-forward neural network (FFN) C_1 .

Multimodal Encoder. The multimodal encoder E accepts as inputs the text ($t \in \mathbb{R}^{m-1}$) and image representation ($v \in \mathbb{R}^{m-1}$). The image and text representation are fused via either i) early-fusion or ii) late-fusion modules. The final multimodal representation is denoted as $c \in \mathbb{R}^{m-1}$, where $c = E(t; v)$.

This module serves as a plug-and-play replacement for any multimodal encoder, whether based on early-fusion or late-fusion. In our implementation for this paper, we use a late-fusion-based module, which consists of CLIP (Radford et al., 2021) and MFB (Yu et al., 2017), as commonly adopted in the literature (Bandyopadhyay et al., 2024).

Classifier C_1 . The classifier gets the multimodal representation c and via a chain of feed-forward neural nets, it gets transformed into a vector $z \in \mathbb{R}^{L-1}$, where L is the number of classes in the output label. A softmax function is used which converts logit z into a probability distribution $y_1 = \text{softmax}(z)$. Supposing the one-hot ground truth probability distribution is \hat{y}_1 , the cross-entropy loss which is used to optimize the post-hoc classifier is

$$L_{PH} = -[\hat{y}_1 \log(y_1)] \quad (1)$$

2.2 CAUSE

The CAuSE is composed of i) A language model (LM) called L_1 which reconstructs the input text. ii) Another LM L_2 which generates the explanation. L_2 is coupled with another classifier (C_2) which is trained to predict the outputs of the original classifier C_1 . *It is important to note that L_1 and L_2 share the same weights and are both implemented using a single GPT-2 small model with 350 million parameters, reducing memory consumption.*

Training the LMs. The LMs are trained using vanilla causal language modelling (CLM) loss. Specifically, the multimodal representation c is broken into two components c_0 and c_1 by passing them through two separate FFNs (F_0 and F_1) which bring their dimension to match with LM embedding dimension \mathbb{R}^{768-1} , such that $c_0 = F_0(c)$, and $c_1 = F_1(c)$.

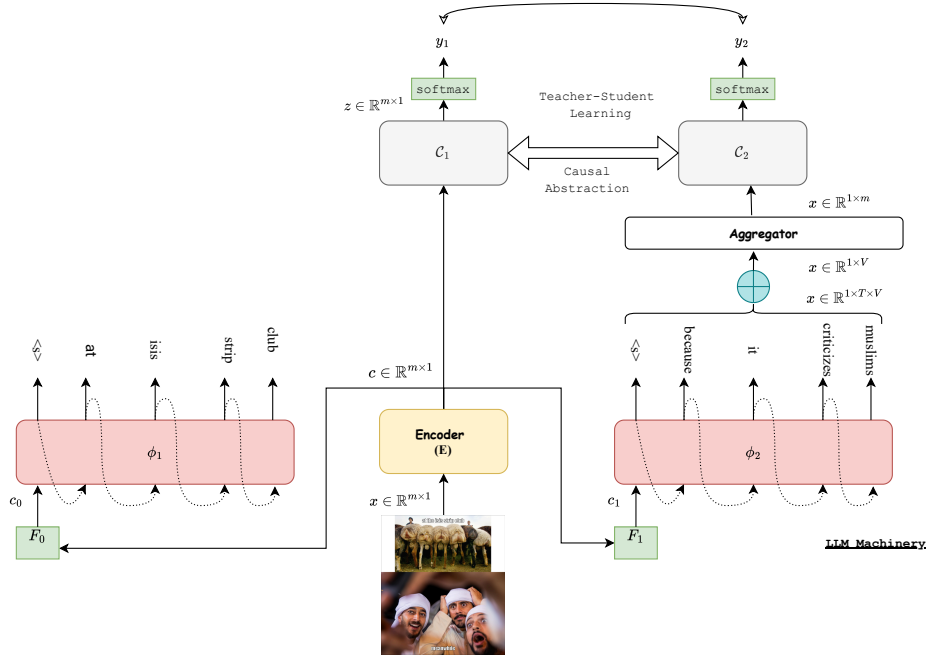


Figure 1: Diagram of our proposed framework CAUSE and the post-hoc classifier.

Given c_0 , ϕ_1 reconstructs next word (x_i) for the i -th step via the following loss over a total of T^0 time-steps:

$$L_1 = \sum_{i=1}^{T^0} \log P_1(x_i | x_{i-1}) \quad \text{where } x_0 = c_0 \quad (2)$$

Similar equation is used to train ϕ_2

$$L_2 = \sum_{i=1}^T \log P_2(x_i | x_{i-1}) \quad \text{where } x_0 = c_1 \quad (3)$$

Aggregator A . The logits x_i retrieved from ϕ_2 has the dimension $\mathbb{R}^{1 \times T \times V}$, where V is the vocabulary size. These logits are first summed up along the time axis, which yields an intermediate vector x having dimension of $\mathbb{R}^{1 \times V}$. This is then passed through another FFN which converts into a dimension same as c , which is $\mathbb{R}^{m \times 1}$.

Classifier C_2 . The aggregated output having the same dimension as c is passed through a classifier C_2 architecturally identical to C_1 . C_2 is then trained to predict labels from C_1 ¹. y_1 is the output distribution from C_1 . Similarly, the probability distribution of C_2 is $y_2 = \text{softmax}(C_2(x))$, where $x = (A \oplus F_1)(c)$. We minimize the Cross-Entropy loss between outputs of C_2 and C_1 as:

$$L_C = -[y_1 \log(y_2)] \quad (4)$$

3 TRAINING METHODOLOGY

Training CAUSE involves two steps other than using L_C to align C_2 to C_1 . They are i) Linguistic Infusion, ii) Causal Intervention.

¹because we want to **mimic** C_2 using C_1 output.

162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215

3.1 LINGUISTIC INFUSION (LI)

We denote the input to the classifier C_1 as c , which is a multimodal encoding from the encoder. This captures the overall encoded representation of the multimodal source input. Through LI, we want to enrich c with input source $(t; v)$ such that the latter could possess enough source information. LI is performed because: We only use a projected version of c as the input token representation c_2 to C_2 . This essentially serves as a bottleneck and most of the source information is lost when input is given to the LLM.

Assuming $M = (t; v)$, in LI, the enrichment of c through source can be defined as the following constrained maximization problem following Plug and Play Language Model (PPLM)(Dathathri et al., 2020).

$$\hat{c} = \arg \max_c P(c|M) \quad \text{such that} \quad C_1(\hat{c}) = C_1(c) \tag{5}$$

Applying Bayes’ theorem, $P(c|M) \propto P(c)P(M|c)$. Subsequently, the optimization Equation 5 can be written as: $\hat{c} = \arg \max_c P(M|c)$.

To estimate $P(M|c)$, we use an autoencoder which tries to predict M from c . Formally, we try to estimate $P(d|c)$ by training an autoencoder which is trained to minimize a loss denoted by $L_{AE} = \|d - M\|_j$. This ensures d becomes as close to M as possible. Specifically, to find \hat{c} , we train the autoencoder first and then perform gradient descent of c along the loss. We use $\hat{c} = c - \gamma_c L_{AE}$ as the iterative update formula to get \hat{c} from c .

3.2 CAUSAL INTERVENTION

Causal Abstraction. In Geiger et al. (2021c), the authors introduced the concept of causal abstraction for neural models. They define a neural network, N_2 , as a causal abstraction of a higher-level causal model, N_1 , if the neural representations of N_2 exhibit the same causal properties as the corresponding high-level variables in N_1 . This alignment is achieved through the Interchange Intervention Training (IIT) objective.

A natural extension of this idea is to consider N_1 as a structurally identical neural network to N_2 and apply IIT between them, keeping N_1 frozen. This process ensures that N_2 becomes a causal abstraction of N_1 . In our framework, we replace N_1 with C_1 and N_2 with C_2 . Through IIT, we aim to ensure that the structurally identical classifier C_2 becomes a causal abstraction of C_1 .

Benefits of Causal Abstraction. The type of causal abstraction learned through IIT is referred to as *constructive abstraction* in the causality literature. This concept ensures a systematic correspondence between interventions on the neurons in N_1 and those in N_2 . Unlike a traditional teacher-student loss, which merely teaches the student to mimic the teacher’s output, causal abstraction ensures that the student model internally mirrors the teacher’s decision-making process. Through IIT, we guarantee that interventions on N_1 have corresponding effects on N_2 , meaning that N_2 operates in the same causal manner as N_1 .

We theoretically demonstrate that applying IIT can have significant implications if specific conditions are met. *Notably, when the weights of C_1 and C_2 remain the same throughout the IIT process:*

- The LLM machinery (i.e., A, C_2 along with F_1 , combined as $F(z) = (A \ C_2 \ F_1)(z)$) perfectly simulates the encoder, such that for any input x , $F(E(x)) = E(x)$. Hence, the output from the LLM machinery matches that of the encoder [proven in **Theorem 1**].
- Building on this result, under a specific set of assumptions, we further show that the LLM machinery, together with C_2 (referred to as the “*explainer*”), forms a causal abstraction of the encoder and C_1 (the “*post-hoc classifier*”) [proven in **Theorem 2**].

Teacher-student objective. Figure 2 illustrates the training process for C_2 . A sample input, consisting of both an image and a text from the dataset, is passed through the encoder. The encoder produces an output c , represented as a 3-dimensional vector, which is then fed into C_1 . Assuming the weights in the first layer are all set to one, the activation of the i_1 -th neuron (as shown in the diagram) would be calculated as $1 \cdot 0.1 + 1 \cdot 0.2 + 1 \cdot 0.3 = 0.6$. The final activation is then computed as $y_1 = 3 \cdot 0.6 + 2 \cdot 0.6 = 3$.

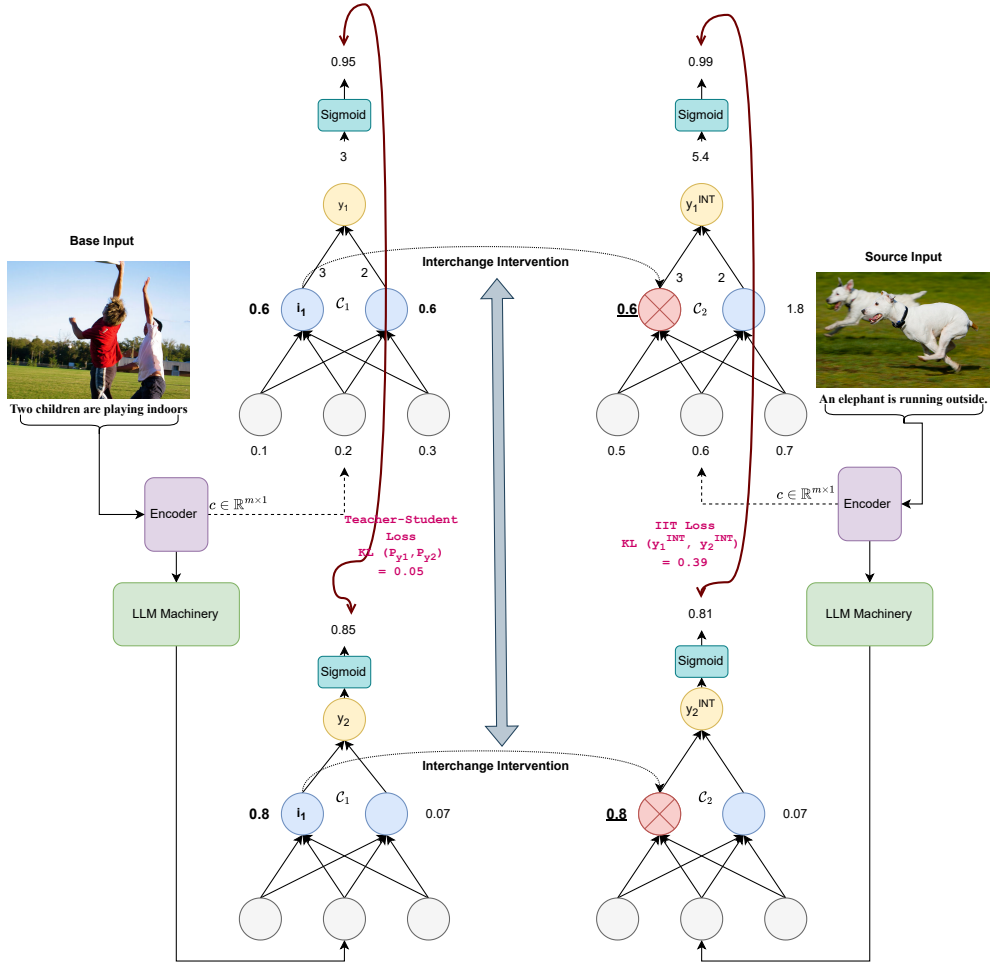


Figure 2: Causal Abstraction is enabled by IIT objective. Along with the teacher-student training objective, IIT poses as indispensable for C_2 to be a causal abstraction of C_1 .

Simultaneously, the output c is passed through the LLM machinery, which generates an activation that is forwarded to C_2 , producing an activation denoted as y_2 . To ensure C_2 mirrors the behavior of C_1 , we calculate the final loss using the KL divergence between their outputs:

$$L_{TS} = KL(P_{y_1} P_{y_2}) \quad (6)$$

where $P_{y_1} = [(y_1); 1 \quad (y_1)]$ and $P_{y_2} = [(y_2); 1 \quad (y_2)]$. This approach can be generalized to handle multiple outputs by applying the softmax function.

IIT objective. The Interchange Intervention (II) process is depicted in Figure 2. A neuron is randomly selected from C_1 (denoted as i_1), and the II is applied. For a given source input, let $c = [0.5; 0.6; 0.7]$ (shown on the right-hand side). The II process ensures that the value of neuron i_1 is replaced with its original value, 0.6, which was obtained when the base input was processed. The final value after this intervention, referred to as the “intervened output,” is represented as y_1^{INT} for C_1 .

The same operation is carried out for C_2 , and the resulting “intervened output” is denoted as y_2^{INT} . Following the methodology of Geiger et al. (2021c), to ensure that C_2 becomes a causal abstraction of C_1 , we minimize the IIT loss between the two outputs:

$$L_{IIT} = KL(P_{y_1^{INT}} P_{y_2^{INT}}) \quad (7)$$

CAuSE Loss Function. The final loss used to train CAuSE (i.e. L_{CAuSE}) is defined as a sum of all individual loss terms.

$$L_{CAuSE} = L_{C_1} + L_{C_2} + L_{IIT} + L_{TS} + L_C + kW_{C_1} + W_{C_2}k_F \quad (8)$$

where $kW_{C_1} + W_{C_2}k_F$ denotes the frobenius norm between the weights of C_1 and C_2 respectively. This term ensures that weights of C_1 and C_2 remain the same during training.

Counterfactual F1 score. We hypothesize that if the explainer becomes a causal abstraction of the post-hoc classifier, it should still mimic the classifier under counterfactual input. To evaluate this, we introduce the counterfactual F1 (c-F1) score. Our empirical analysis shows that using only teacher-student training results in poor performance on counterfactual input, as reflected by a low c-F1 score. However, when combined with IIT, the explainer achieves a robust c-F1 score. Algorithm 1 details c-F1 calculation, and Table 2 compares methods based on their c-F1 scores.

3.2.1 CALCULATING COUNTERFACTUAL F1 SCORE

Suppose $x \in T$ is a data-point from test set. As posed in Feder et al. (2022), the corresponding counterfactual input x^0 for the post-hoc classifier would satisfy the following:

$$x^0 = \arg \min_{x^0 \in T} d(x; x^0) \quad \text{such that} \quad C_1(x) \neq C_1(x^0) \quad (9)$$

d is any kind of distance metric (e.g. manhattan, euclidean etc) between these data points. $C_1(z)$ denotes the output class from C_1 for any input z .

Subsequently, any counterfactual for x can be expressed as: $x^0 = x + \delta$, where $\delta = x^0 - x$ is the perturbation between normal and counterfactual input. Note that $E(x^0)$ could not be a good counterfactual input for the LLM machinery, as $x^0 \in T$ and high simulation performance between C_2 and C_1 means C_2 could easily find label of x^0 . Therefore, we resort to the following three constraints while designing a counterfactual input z^0 for the LLM Machinery: i) z^0 should be a counterfactual for C_1 , as our task is to measure how many counterfactuals for C_1 are also counterfactual for C_2 . ii) z^0 should not be representation of any data-point from T , iii) It should be a transformation of the original data-point x and its perturbation δ .

We assume z^0 has the following generic form (satisfying ii. and iii.), $z^0 = z + T(\delta)$, where $z = E(x)$ is an input to the LLM machinery. So, $z^0 = E(x) + T(\delta)$. Note that to ensure $T(\delta)$ is an invertible function of δ (satisfying iii.), we use an autoencoder which maps δ to $T(\delta)$ and then back to δ . Finally, to satisfy the first constraint, we ensure the following holds true:

$$C_1(E(x) + T(\delta)) = C_1(E(x + \delta)) \quad (10)$$

Note that this can be enforced by standard KL divergence loss between C_1 and C_2 .

Algorithm 1: Counterfactual F1 Score for C_1 and C_2

Input: Data-point $x \in T$

Function CounterFactual(x):

```

 $x^0 = \arg \min_{x^0 \in T} d(x; x^0)$  s.t.  $C_1(x) \neq C_1(x^0)$ ;
 $\delta = x^0 - x$ ; // Compute the perturbation
 $z = E(x)$ ; // Encode the original input
 $T(\delta) = f(\delta)$  where  $g(f(\delta)) = \delta$ ; // Transform the perturbation
 $z^0 = z + T(\delta)$ ;
return  $z^0, x^0$ 

```

Procedure Calculate Counterfactual F1 score

```

ZList = [];
XList = [];
while  $T \neq \emptyset$  do
    Sample  $x \in T$ ; // Draw a new data point
     $z^0, x^0 = \text{CounterFactual}(x)$ ;
    Ensure:  $C_1(z^0) = C_1(E(x^0))$ ; // constraint i.
    ZList =  $[fC_2(z^0)g]$ ; // Append  $C_2(z^0)$  to the list
    XList =  $[fC_1(x^0)g]$ ; // Append  $C_1(x^0)$  to the list
     $T = T - \{x\}$ ;
return  $F_1 = \text{score}(XList; ZList)$ 

```

Table 1: Ablation studies. L_{MSE} refers to an MSE loss between c and x , such that $F(E(x)) = E(x)$. B-1, B-2, B-3, B- refers to Bleu scores with various n gram precisions.

		F1	B-1	B-2	B-3	B-4	BertScore
Hateful Meme	L_2	97.29	0.65	0.53	0.47	0.39	0.971
	$L_1 + L_2$	98.44	0.65	0.53	0.47	0.39	0.971
	$L_1 + L_2 + L_{MSE}$	98.55	0.64	0.53	0.46	0.39	0.971
	$L_1 + L_2 + L_C$	98.33	0.64	0.53	0.46	0.38	0.971
	L_{CAUSE}	98.09	0.64	0.51	0.44	0.36	0.969
e-SNLI-VE	L_2	94.66	0.39	0.27	0.19	0.15	0.905
	$L_1 + L_2$	94.08	0.39	0.27	0.19	0.15	0.905
	$L_1 + L_2 + L_{MSE}$	94.39	0.39	0.27	0.20	0.15	0.905
	$L_1 + L_2 + L_C$	94.94	0.38	0.27	0.20	0.15	0.905
	L_{CAUSE}	91.96	0.39	0.27	0.20	0.15	0.904

Table 2: In addition to the Counterfactual F1 score, we also report the number of comprehensible generations ($\#gen$), as many outputs from CAUSE tend to be gibberish when counterfactual input is provided. To provide a more holistic evaluation of CAUSE’s performance on counterfactual inputs, we compute the harmonic mean (HM) of the F1 score and $\#gen$, capturing both accuracy and the quality of generated explanations.

	Hateful Meme			e-SNLI-VE		
	F1	# gen.	HM	F1	# gen.	HM
$L_1 + L_2$	55.02	17	32.98	93.81	167	28.35
$L_1 + L_2 + L_{MSE}$	33.33	2	3.976	85.94	322	46.84
$L_1 + L_2 + L_C$	53.78	91	15.56	73.48	850	78.82
L_{CAUSE}	75.81	755	75.61	85.24	986	91.43

4 RESULTS AND ANALYSIS

4.1 AUTOMATIC EVALUATION

The proposed system is evaluated across two verticals: i) Mimicking capability of the *explainer* when compared to *post-hoc classifier*, and ii) performance under counterfactual input. The automatic evaluation metric used to evaluate CAUSE performance can be grouped into two categories, i) **Faithfulness**: This is measured by the obtained F1 score measured between the predicted class by the LLM machinery (or C_2) and the predicted class by the post-hoc classifier C_1 . The predicted class obtained from the LLM machinery is extracted either from the prediction of C_2 or from C_1 classifier head. ii) **Plausibility**: This is measured as the BLEU scorePapineni et al. (2002) and BERTScoreZhang et al. (2020) between the generated explanation and the ground truth explanation from the test set.

Baselines. *To the best of our knowledge, ours is the first approach that generates faithful natural language explanations directly from a classifier’s hidden state.* Nonetheless, we compare our method with several Visual Language Model (VLM) baselines as there are no existing techniques for this task in the literature. Specifically, we use zero-shot and few-shot ($k = 2$ or 3) prompting with i) PaLiGemma(Beyer et al., 2024), ii) LLaVA(Liu et al., 2023), to simulate the predicted class from a given classifier (C_1), based on previous input-output examples². Since it is challenging to simulate a model’s behaviour without access to its hidden activations, few-shot prompting often performs similarly or even worse than zero-shot prompting. The faithfulness of the explanations, as measured by the F1 score, is inconsistent and random (below 50% for the Hateful Memes dataset and below 33% for e-SNLI-VE), as shown in Table 3 The fine-tuned models (shown through FT suffix) perform the best, where the F1 score reaches close to 70%.

²The specific prompting used are shown in the Appendix C

Table 3: Various VLM-based baselines. FT as a suffix denotes finetuned model. Note that LLaVA has 7B and PaLiGemma has 3.5B parameters respectively.

Dataset	Baselines	F1	B-1	B-2	B-3	B-4	BertScore
Hateful Meme	<i>LLaVA-0-shot</i>	58.44	0.09	0.01	0.01	0.01	0.889
	<i>LLaVA-2-shot</i>	46.55	0.12	0.02	0.01	0.01	0.864
	<i>PaLiGemma -FT</i>	72.33	0.41	0.27	0.15	0.09	0.891
	<i>LLaVA-FT</i>	72.38	0.40	0.27	0.17	0.13	0.894
e-SNLI-VE	<i>LLaVA-0-shot</i>	33.12	0.22	0.07	0.03	0.02	0.876
	<i>LLaVA-3-shot</i>	35.77	0.22	0.07	0.03	0.01	0.869
	<i>PaLiGemma -FT</i>	64.90	0.19	0.04	0.01	0.01	0.866
	<i>LLaVA-FT</i>	64.29	0.22	0.08	0.03	0.02	0.859

Table 4: Case studies: A few example where our model succeeds. **Pred:** Explanation generated from the model, **GT:** Ground truth explanation.

Image Path	Pred	GT	y_1	y_2
489134459.jpg	A woman is a female. Just because she is sitting on a curb, it means she is outside..	A boy and a girl are two kids. The front of a house is located outside..	E	E
5631556013.jpg	A man is performing on the street in front of a group of people..	man jumping from someone	E	E
12507.png	it promotes negative stereotypes about people who are Muslim and suggests that all Muslims are violent or dangerous	it promotes harmful stereotypes about Muslims, suggesting that they are violent and intolerant.	O	O
91462.png	it promotes racism, specifically by implying that white people are superior to other people.	it promotes harmful stereotypes about black women.	O	O

4.1.1 ABLATION STUDIES

What is the use of various loss function other than L_1 and L_2 loss? As seen from Table 1, it can be posed as a valid question. Indeed, when using our proposed method which uses L_{IIT} and other losses seem to achieve slightly lower F1 score (indicating slightly lower faithfulness) and slightly lower BLEU score / BERTScore (indicating slightly lower plausibility). Note that this difference is very small and it is compensated by very high counterfactual F1 score as shown in Table 2 obtained by our method.

Why is IIT required? As can be seen from Table 1 and Table 2, good mimicking performance under normal condition does not always entail good performance when posed with counterfactual input. IIT ensures causal abstraction between C_2 and C_1 theoretically and this is also being verified empirically by the high counterfactual F1 score obtained by our method which uses IIT.

Is L_1 necessary? L_1 which is used to train the LLM (θ_1) which reconstructs the content is required, because that shows better mimicking performance (at least in Hateful meme dataset) coupled with L_2 than using L_2 alone. This can be attributed to the joint training objective which ensures that C possesses enough input information to aid in explanation generation by the second LLM θ_2 .

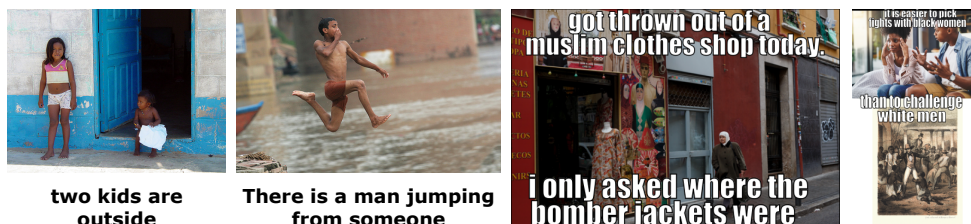


Figure 3: Examples corresponding to Table 4

Table 5: Error Analysis: These cases demonstrate four kinds of error cases that is prevalent among our proposed framework CAuSE.

Image Path	Gen	GT	y	\hat{y}
7046014201.jpg	Construction work necessitates working outdoor.	A juggler is juggling clubs at an outdoor plaza.	E	E
2731298834.jpg	A dog that is jumping into the water will be wet.	swimming is perform in a water.	E	E
151215569.jpg	A man is pulling on the street so he is outdoors.	A young blond girl describes a child and a man describes and adult.	E	E
59260.png	it suggests that white people are superior to other people, which is not accurate	it promotes anti-Semitism and hatred towards Jewish people.	O	O



Figure 4: Memes pertaining to Error Analysis shown in Table 5

4.2 QUALITATIVE STUDIES

4.2.1 CASE STUDIES

In Table 4, we present four successful examples from the e-SNLI-VE and Facebook Hateful Memes datasets (two from each). The first two examples are from e-SNLI-VE, while the latter two are from the Hateful Memes dataset. In the e-SNLI-VE examples, CAuSE produces semantically accurate explanations while correctly predicting the class as “Entailment.” A noticeable pattern emerges from these successful cases: CAuSE tends to perform well when the class-level information can be explicitly inferred from the combination of the image and text. Specifically, for e-SNLI-VE, when CAuSE generates accurate explanations, the hypothesis often functions like a caption for the image premise, which aids in classification.

For the Hateful Memes examples, CAuSE also generates correct explanations. In these cases, the image and the embedded text are semantically aligned rather than contradictory (i.e., where the image-text mismatch is used to evoke negative sentiment). In such instances, CAuSE effectively provides explanations and correctly predicts the appropriate output class.

4.2.2 ERROR ANALYSIS

We selected four examples from the e-SNLI-VE and Hateful Memes datasets to highlight common types of errors made by CAuSE (in Table 5). These errors can be categorized into three main types:

Lack of representation capability: In the first example, the hypothesis reads, “A juggler is performing outdoors,” and the premise is entailed, as confirmed by the ground truth explanation: “A juggler is juggling clubs at an outdoor plaza.” However, CAuSE incorrectly generates the explanation: “Construction work necessitates working outdoors,” confusing the act of juggling with construction work. This error likely stems from insufficient information in the initial representation, C , used by CAuSE.

Lack of object-level representation: The post-hoc classifier relies on unimodal representations from the CLIP architecture, which lacks fine-grained object-level details, compared to models like Faster R-CNNRen et al. (2016). In the second example, instead of recognizing a “dog,” CAuSE should have identified “a woman and children” for a more accurate representation.

The third example illustrates both issues: lack of object-level representation and general representation capability. These limitations prevent CAuSE from correctly describing the relationship between “a young blonde girl,” “an adult,” and “a man pulling outdoors.”

Implicit semantic category: In the fourth example, although CAuSE correctly predicts the output class as offensive, it does so for the wrong reasons. Even a human might struggle to recognize the

implicit anti-Semitism in this meme, as neither the image nor the text explicitly convey the historical context of the Holocaust, where six million Jews were killed. Without this prior knowledge, CAuSE cannot fully comprehend the offence.

5 RELATED WORK

Interpretability. Interpretability is crucial for building trust in AI systems within human society. Techniques like LIME, SHAP and RISE (Ribeiro et al., 2016; Lundberg & Lee, 2017; Petsiuk et al., 2018) explain classifier predictions by providing feature-level explanations for local interpretability. Although model-agnostic, these methods lack global interpretability, which is addressed by GALE van der Linden et al. (2019), where local explanations are aggregated into a global model understanding. Approaches like SmoothGrad Smilkov et al. (2017) and Integrated Gradients (Sundararajan et al., 2017b) utilize input gradients for model explanation, while CAM Zhou et al. (2015) highlights critical pixels for decision making in visual classification. Counterfactual generations (Chang et al., 2019; Mothilal et al., 2020; Goyal et al., 2019) also offer insights into the inner working of the model by revealing decision boundaries. However, most of these methods often overlook implicit features behind model decisions and lack natural language explanations. To address these limitations, we propose a novel framework for classifier explanations which generates both *faithful* and *plausible* natural language outputs.

Causal Interpretability. Causal interpretability refers to the ability to explain a model’s decisions by identifying the cause-effect relationships between input features and the model’s output. Feder et al. (2022) demonstrated how incorporating causal reasoning in NLP tasks can improve model predictions and enhance interpretability by going beyond simple correlations between input features and outputs. Further works by Geiger et al. (2021b); Vig et al. (2020); Meng et al. (2023) have focused on causal abstraction and causal mediation analysis, helping to create causally faithful models and identify both direct and indirect causal factors behind certain model behaviors. In addition to generating counterfactuals, testing models on counterfactual inputs is another critical aspect of understanding model behavior. Since creating exact counterfactuals is challenging, Abraham et al. (2022); Calderon et al. (2022), recent research has focused on approximations Geiger et al. (2021b) or counterfactual representations Feder et al. (2021); Elazar et al. (2021); Ravfogel et al. (2021). Our proposed counterfactual metric is inspired by these counterfactual representations. Moreover, most of the existing works focuses on single modality (e.g., text or vision) Feder et al. (2021); Goyal et al. (2020). In contrast, the natural language causal explanation provided by our framework is model-agnostic, task-agnostic, and capable of handling multimodal inputs.

6 CONCLUSION AND FUTURE WORK

In this paper, we presented CAuSE (Causal Abstraction under Simulated Explanation), a novel framework for generating causally faithful natural language explanations for multimodal classifiers. By integrating *Interchange Intervention Training* (IIT) with a Language Model (LM) based module, CAuSE addresses the limitations of existing interpretability methods, ensuring explanations are directly tied to the classifier’s causal reasoning. Our new Counterfactual F1 score highlights CAuSE’s state-of-the-art performance on datasets like e-SNLI-VE and Facebook Hateful Memes.

While CAuSE demonstrates robust task-agnostic performance, future work will focus on enhancing fine-grained object-level representations and extending the framework to temporal data, such as video and audio. Additionally, we aim to explore how *self-supervised* learning and deeper integration of implicit cultural knowledge can further improve the framework’s scalability and contextual understanding in real-world applications.

ETHICS STATEMENT

The datasets used in this study are publicly available. The explanations for hateful memes were generated from publicly accessible meme data, and we adhered to copyright regulations to prevent any infringement. Furthermore, our research received approval from the Institutional Review Board (IRB). Since the hateful meme dataset includes content that may be offensive, we recommend that readers approach it with discretion.

540 REPRODUCIBILITY STATEMENT

541
542 To ensure reproducibility, we consistently use a random seed of 42 across all experiments. The code
543 is available at <https://anonymous.4open.science/r/CAUSE-5BD0>, and model outputs
544 will be shared upon paper acceptance. These outputs can be cross-verified with the results generated
545 from the provided code. Our method is theoretically sound, supported by the proof of the proposed
546 theorem and proposition outlined in Appendix A, with all underlying assumptions clearly stated and
547 justified.

548
549 REFERENCES

- 550
551 Eldar David Abraham, Karel D’Oosterlinck, Amir Feder, Yair Ori Gat, Atticus Geiger, Christopher
552 Potts, Roi Reichart, and Zhengxuan Wu. Cebab: Estimating the causal effects of real-world
553 concepts on nlp model behavior, 2022.
- 554 Chirag Agarwal, Sree Harsha Tanneru, and Himabindu Lakkaraju. Faithfulness vs. plausibility: On
555 the (un)reliability of explanations from large language models, 2024.
- 556 Tadas Baltrušaitis, Chaitanya Ahuja, and Louis-Philippe Morency. Multimodal machine learning:
557 A survey and taxonomy, 2017. URL <https://arxiv.org/abs/1705.09406>.
- 558 Diblyanayan Bandyopadhyay, Asmit Ganguly, Baban Gain, and Asif Ekbal. Semantify: Unveil-
559 ing memes with robust interpretability beyond input attribution. In Kate Larson (ed.), *Proceed-*
560 *ings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24*, pp.
561 6189–6197. International Joint Conferences on Artificial Intelligence Organization, 8 2024. doi:
562 10.24963/ijcai.2024/684. URL <https://doi.org/10.24963/ijcai.2024/684>. Main
563 Track.
- 564 Lucas Beyer, Andreas Steiner, André Susano Pinto, Alexander Kolesnikov, Xiao Wang, Daniel Salz,
565 Maxim Neumann, Ibrahim Alabdulmohsin, Michael Tschannen, Emanuele Bugliarello, Thomas
566 Unterthiner, Daniel Keysers, Skanda Koppula, Fangyu Liu, Adam Grycner, Alexey Gritsenko,
567 Neil Houlsby, Manoj Kumar, Keran Rong, Julian Eisenschlos, Rishabh Kabra, Matthias Bauer,
568 Matko Bošnjak, Xi Chen, Matthias Minderer, Paul Voigtlaender, Ioana Bica, Ivana Balazevic,
569 Joan Puigcerver, Pinelopi Papalampidi, Olivier Henaff, Xi Xiong, Radu Soricut, Jeremiah Harm-
570 sen, and Xiaohua Zhai. Paligemma: A versatile 3b vlm for transfer, 2024. URL <https://arxiv.org/abs/2407.07726>.
- 571 Nitay Calderon, Eyal Ben-David, Amir Feder, and Roi Reichart. DoCoGen: Domain counter-
572 factual generation for low resource domain adaptation. In Smaranda Muresan, Preslav Nakov,
573 and Aline Villavicencio (eds.), *Proceedings of the 60th Annual Meeting of the Association for*
574 *Computational Linguistics (Volume 1: Long Papers)*, pp. 7727–7746, Dublin, Ireland, May
575 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.533. URL
576 <https://aclanthology.org/2022.acl-long.533>.
- 577 Chun-Hao Chang, Elliot Creager, Anna Goldenberg, and David Duvenaud. Explaining image clas-
578 sifiers by counterfactual generation, 2019. URL <https://arxiv.org/abs/1807.08024>.
- 579 Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosin-
580 ski, and Rosanne Liu. Plug and play language models: A simple approach to controlled text
581 generation, 2020. URL <https://arxiv.org/abs/1912.02164>.
- 582 Virginie Do, Oana-Maria Camburu, Zeynep Akata, and Thomas Lukasiewicz. e-snli-ve: Corrected
583 visual-textual entailment with natural language explanations, 2021. URL <https://arxiv.org/abs/2004.03744>.
- 584 Yanai Elazar, Shauli Ravfogel, Alon Jacovi, and Yoav Goldberg. Amnesic probing: Behavioral
585 explanation with amnesic counterfactuals, 2021.
- 586 Amir Feder, Nadav Oved, Uri Shalit, and Roi Reichart. CausaLM: Causal model explanation
587 through counterfactual language models. *Computational Linguistics*, 47(2):333–386, June 2021.
588 doi: 10.1162/coli.a.00404. URL <https://aclanthology.org/2021.cl-2.13>.

- 594 Amir Feder, Katherine A. Keith, Emaad Manzoor, Reid Pryzant, Dhanya Sridhar, Zach Wood-
595 Doughty, Jacob Eisenstein, Justin Grimmer, Roi Reichart, Margaret E. Roberts, Brandon M. Stewart,
596 Victor Veitch, and Diyi Yang. Causal inference in natural language processing: Estimation,
597 prediction, interpretation and beyond. *Transactions of the Association for Computational Linguistics*,
598 10:1138–1158, 2022. doi: 10.1162/tacl.a.00511. URL <https://aclanthology.org/2022.tacl-1.66>.
- 600 Atticus Geiger, Hanson Lu, Thomas Icard, and Christopher Potts. Causal abstractions of neural
601 networks. *CoRR*, abs/2106.02997, 2021a. URL <https://arxiv.org/abs/2106.02997>.
- 602 Atticus Geiger, Hanson Lu, Thomas Icard, and Christopher Potts. Causal abstractions of neural
603 networks, 2021b.
- 605 Atticus Geiger, Hanson Lu, Thomas Icard, and Christopher Potts. Causal abstractions of neural
606 networks, 2021c. URL <https://arxiv.org/abs/2106.02997>.
- 607 Yash Goyal, Ziyang Wu, Jan Ernst, Dhruv Batra, Devi Parikh, and Stefan Lee. Counterfactual visual
608 explanations, 2019. URL <https://arxiv.org/abs/1904.07451>.
- 609 Yash Goyal, Amir Feder, Uri Shalit, and Been Kim. Explaining classifiers with causal concept effect
610 (cace), 2020.
- 612 Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang,
613 and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021. URL <https://arxiv.org/abs/2106.09685>.
- 614 Douwe Kiela, Hamed Firooz, Aravind Mohan, Vedanuj Goswami, Amanpreet Singh, Pratik Ring-
615 shia, and Davide Testuggine. The hateful memes challenge: Detecting hate speech in multimodal
616 memes, 2021.
- 617 Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning, 2023. URL
618 <https://arxiv.org/abs/2304.08485>.
- 621 Scott Lundberg and Su-In Lee. A unified approach to interpreting model predictions, 2017. URL
622 <https://arxiv.org/abs/1705.07874>.
- 623 Andreas Madsen, Sarath Chandar, and Siva Reddy. Are self-explanations from large language mod-
624 els faithful?, 2024.
- 626 Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual
627 associations in gpt, 2023.
- 628 Ramaravind K. Mothilal, Amit Sharma, and Chenhao Tan. Explaining machine learning classifiers
629 through diverse counterfactual explanations. In *Proceedings of the 2020 Conference on Fair-
630 ness, Accountability, and Transparency*, FAT* '20. ACM, January 2020. doi: 10.1145/3351095.
631 3372850. URL <http://dx.doi.org/10.1145/3351095.3372850>.
- 632 Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic
633 evaluation of machine translation. In Pierre Isabelle, Eugene Charniak, and Dekang Lin (eds.),
634 *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pp.
635 311–318, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics.
636 doi: 10.3115/1073083.1073135. URL <https://aclanthology.org/P02-1040>.
- 637 Vitali Petsiuk, Abir Das, and Kate Saenko. Rise: Randomized input sampling for explanation of
638 black-box models, 2018. URL <https://arxiv.org/abs/1806.07421>.
- 640 Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agar-
641 wal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya
642 Sutskever. Learning transferable visual models from natural language supervision, 2021.
- 643 Shauli Ravfogel, Grusha Prasad, Tal Linzen, and Yoav Goldberg. Counterfactual interventions reveal
644 the causal effect of relative clause representations on agreement prediction. In Arianna Bisazza
645 and Omri Abend (eds.), *Proceedings of the 25th Conference on Computational Natural Language
646 Learning*, pp. 194–209, Online, November 2021. Association for Computational Linguistics. doi:
647 10.18653/v1/2021.conll-1.15. URL <https://aclanthology.org/2021.conll-1.15>.

- 648 Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object
649 detection with region proposal networks, 2016.
- 650 Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should i trust you?": Explaining
651 the predictions of any classifier, 2016. URL <https://arxiv.org/abs/1602.04938>.
- 652 Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. Smoothgrad:
653 removing noise by adding noise, 2017. URL <https://arxiv.org/abs/1706.03825>.
- 654 Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks, 2017a.
- 655 Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks, 2017b.
656 URL <https://arxiv.org/abs/1703.01365>.
- 657 Ilse van der Linden, Hinda Haned, and Evangelos Kanoulas. Global aggregations of local explana-
658 tions for black box models, 2019. URL <https://arxiv.org/abs/1907.03039>.
- 659 Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Simas Sakenis, Jason
660 Huang, Yaron Singer, and Stuart Shieber. Causal mediation analysis for interpreting neural nlp:
661 The case of gender bias, 2020.
- 662 Yi Xiao, Felipe Codevilla, Akhil Gurrum, Onay Urfalioglu, and Antonio M. Lopez. Multimodal
663 end-to-end autonomous driving. *IEEE Transactions on Intelligent Transportation Systems*, 23
664 (1):537–547, January 2022. ISSN 1558-0016. doi: 10.1109/tits.2020.3013234. URL <http://dx.doi.org/10.1109/TITS.2020.3013234>.
- 665 Zhou Yu, Jun Yu, Jianping Fan, and Dacheng Tao. Multi-modal factorized bilinear pooling with
666 co-attention learning for visual question answering, 2017. URL <https://arxiv.org/abs/1708.01471>.
- 667 Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. Bertscore: Evaluat-
668 ing text generation with bert, 2020. URL <https://arxiv.org/abs/1904.09675>.
- 669 Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep fea-
670 tures for discriminative localization, 2015. URL <https://arxiv.org/abs/1512.04150>.

678 A THEOREMS

679 During our training process, we have implemented IIT (Interchange Intervention Training) along
680 with an additional constraint that the weights of C_1 and C_2 become the same during training, which
681 is ensured by the frobenius norm term being used as a part of L_{CAUSE}

682 **Theorem 1.** Under the above stated conditions, when C_2 becomes a causal abstraction of C_1 and
683 their weights become the same, $E(x) = F(E(x))$.

684 *Proof.* Without loss of generality, we have considered only two input neurons for both C_1 and C_2 .
685 Under IIT, the following always holds between two output neurons (we assume that a) there exists
686 one intermediate intervened neuron and b) source s and base b inputs are provided as input):

$$\begin{aligned}
 & softmax(y_1) = softmax(y_2) \\
 \Rightarrow & softmax(w_{1i}(E(s)_i); w_{2i}(E(b)_i)) = softmax(w_{1i}^\delta F(E(s)_i); w_{2i}^\delta F(E(b)_i)); \delta i = 1; 2 \\
 & \text{Since } w = w^\delta \\
 \Rightarrow & \frac{\exp(\sum_i w_{1i} E(b)_i)}{\exp(\sum_i w_{1i} E(b)_i) + \exp(\sum_i w_{2i} E(s)_i)} = \frac{\exp(\sum_i w_{1i} F(E(b))_i)}{\exp(\sum_i w_{1i} F(E(b))_i) + \exp(\sum_i w_{2i} F(E(s))_i)} \\
 & \delta i = 1; 2 \\
 & (11)
 \end{aligned}$$

687 Let us assume $E(b) \notin F(E(b))$ so there $\exists j$ such that $E(b)_j \notin F(E(b))_j$. Let us assume,

$$\begin{aligned}
 E[b] &= [p; q] \\
 F[E[b]] &= [p; q] \\
 & (12)
 \end{aligned}$$

We pick $s, b \in D_E \subseteq D_E$ so $s = b$ when $s_i = b_i$. Here D_E refers to the data on which the encoder is being trained. Finally, from Equation 11,

$$w_{21}[E(s)_2 - F(E(s))_2] = \log(\exp(w_{12}E(b)_1 + w_{22}E(b)_2) + \exp(w_{11}E(s)_1 + w_{21}E(s)_2)) - \log(\exp(w_{12}F(E(b))_1 + w_{22}F(E(b))_2) + \exp(w_{11}F(E(s))_1 + w_{21}F(E(s))_2)) \quad (13)$$

Here, we name all the variables to for better readability:

$$w_{11} = \quad w_{12} = \quad w_{21} = \quad w_{22} = \quad (14)$$

From these, considering $s = b$ we can rewrite the equation 13 for two output neurons as:

$$q(1) = \underbrace{\log(\exp(p + q) + \exp(p + q))}_{P_1} - \underbrace{\log(\exp(p + q) + \exp(p + q))}_{P_2} \quad (15)$$

$$q(1) = P_1 - P_2 \quad (16)$$

This means, under IIT if :

$$q(1) = P_1 - P_2 + k_1 \quad (17)$$

and

$$q(1) = P_1 - P_2 + k_2 \quad (18)$$

then , $k_1 = k_2 = 0$

Now, we impose pairwise, equality of weights $w_{11} = w_{12}$ and $w_{21} = w_{22}$. Using this condition, the individual equations will become:

$$q(1) = q(1) + k_1 \quad (19)$$

$$q(1) = q(1) + k_2 \quad (20)$$

$$\Rightarrow k_1 = k_2 = 0 \quad (21)$$

The above condition where we consider pairwise equality of weights is a degenerate case. In this situation, every input node has the same weightage as it is passed to the deeper layers. This is an edge case rarely seen in real training scenarios.

We obtain the following values of k when the degenerate case is not considered:

$$k_1 = \log(\exp(p + q) + \exp(p + q)) - \log(\exp(p + q) + \exp(p + q)) + q(1) \quad (22)$$

$$k_2 = \log(\exp(p + q) + \exp(p + q)) - \log(\exp(p + q) + \exp(p + q)) + q(1) \quad (23)$$

The above values of $k_i \neq 0 \quad \forall i = 1, 2$. However, this is a contradiction since, this violates the property of (IIT). This means our initial assumption of $F(E(b)) \neq E(b)$ is wrong. This proves that $F(E(b)) = E(b)$.

Also, it is noteworthy that in the equations 19 $k_1 = k_2 = 0$ only when $\beta = 1$. This again validates our claim that $E(b) = [p; q] = F(E(b))$ \square

Definition. LLM machinery F coupled with the classifier C_2 is called *Explinator*, while the encoder with the classifier C_1 is called the *Post-hoc classifier*. We also assume the following, there exists a function ϕ which maps a set of variables (V_E) in E to a set of variables (V_F) in F , such that $\phi: V_E \rightarrow V_F$. We also assume that we intervene in a neuron $i_e \in V_E$, such that a mapped neuron ($i_e \in V_F$) is also intervened. Under this intervention schema, the intervened outputs of E and F are denoted as $F^{INT}(E(s); E(b))$ and $E^{INT}(s; b)$. The following lemma shows their relation.

Lemma 1. If C_1 and C_2 become identical (their weights are equal and they are causal abstraction of each other), $F^{INT}(E(s); E(b)) = E^{INT}(s; b)$, meaning F is a causal abstraction of E .

Proof. By Theorem 1, we know that if C_1 and C_2 become identical, then $F(E(x)) = E(x)$. This entails F acts as a perfect autoencoder considering F only accepts input from E . The source (s) and base (b) equivalent input for F would be $E(s)$ and $E(b)$, respectively. When supplied with $E(s)$ and $E(b)$ and an interchange intervention is performed in F , the F being a perfect autoencoder will try to reconstruct $E(s)$ but due to intervention with $E(b)$, the output will also contain a part of $E(b)$. F being a linear function of $E(x)$, we can write:

$$F^{INT}(E(s); E(b)) = f_1(E(s); E(b); w_F)E(s) + f_2(E(s); E(b); w_F)E(b) \quad (24)$$

$$= \alpha_1 E(s) + \alpha_2 E(b) \quad (25)$$

$$[\text{as } F^{INT}(x) = F(x) = x \text{ and by previous argument}] \quad (26)$$

Note that $E^{INT}(s; b) = E(s; b)$ (denoting a function of $(s; b)$) and as E is a linear function of $s; b$, by similar argument:

$$E^{INT}(s; b) = E(s; b) = g_1(s; b; w_E)E^{INT}(s) + g_2(s; b; w_E)E^{INT}(b) \quad (27)$$

$$= \beta_1 E(s) + \beta_2 E(b) \quad [\text{as } E^{INT}(x) = E(x)] \quad (28)$$

If the contribution of $E(s)$ towards its reconstruction by F , as quantified by α_1 and its equivalent contribution (through s) towards the intervention in E are the same then $\alpha_1 = \beta_1$. Physically this means s is as important to E as $E(s)$ is important to F for any equivalent b and $E(b)$. This is satisfied trivially when E and F give equal importance to any data point x and its transformed version $E(x)$, which is exactly ensured in training by the fact that $E(x) = F(E(x))$. This further implies $\frac{\alpha_1}{\beta_1} = 1$, and $\frac{\alpha_2}{\beta_2} = 1$. We observe, $F^{INT}(E(s); E(b)) = E^{INT}(s; b)$.

□

Theorem 2: If C_1 and C_2 become identical (their weights are equal and they are causal abstraction of each other), the *Explinator* becomes a causal abstraction of the *Post-hoc classifier*.

Proof. From lemma 1, we showed F is a causal abstraction of E . For any intervention performed between E and F , their outputs are equal, which are being fed to C_1 and C_2 respectively. As for the same input, C_1 and C_2 outputs will match, the final output from the explinator and post hoc classifier will also match. If any intervention is performed between C_1 and C_2 , their output will also match because they were trained to be causal abstraction of each other.

So in summary, for pairwise interchange intervention between E and F or C_1 and C_2 , the final output from post-hoc classifier and explinator will match. This is the definition for causal abstraction. Therefore, the explinator becomes a causal abstraction of the post-hoc classifier. □

B SOME GENERIC THEORETICAL RESULTS

Notation: Assume two identical neural nets N_1 and N_2 . Their weights are w and \hat{w} . These two neural nets are trained on two different datasets: D_1 and D_2 . We denote activation at an arbitrary layer's neuron as i_n , where the subscript i denotes the NNs. We also assume $k = \frac{i_2}{i_1}$. The following lemma shows a relation.

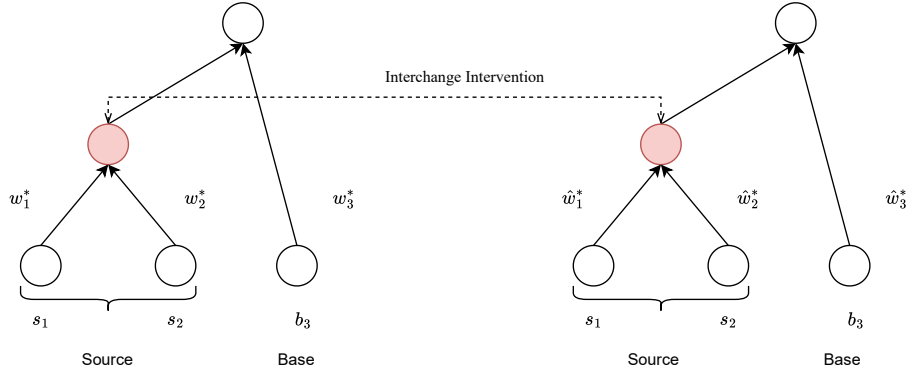


Figure 5: Structure of N_1 and N_2 are assumed to be the same. The red neuron denotes the intervened neuron.

Lemma 2. For N_1 and N_2 , after convergence, $k = f(D_1; D_2; x)$, where x denotes the network input.

Proof. After training is complete, assume the optimal weights are w_1 and w_2 . Naturally, $w_1 = w_1(D_1)$ and $w_2 = w_2(D_2)$. k , being a ratio of the activation of two neural nets, will depend on their inputs, and converged weights. Therefore, $k = f(w_2; w_1; x) = f(D_1; D_2; x)$. \square

Lemma 3. If outputs under interchange intervention are equal (as satisfied by IIT training objective), then k must be of the form $f(D_{IIT}; s; b)$ to ensure N_2 and N_1 are the causal abstraction of each other.

Proof. Suppose, the neural networks converge to a state where both N_1 and N_2 have the parameter values w and \hat{w} . Refer to Figure 5.

If the outputs are equal after intervening on V_1 , and identically V_2 neuron of N_1 and N_2 respectively, then:

$$\underbrace{w_1 s_1 + w_2 s_2}_{i_1} + w_3 b_3 = \underbrace{\hat{w}_1 s_1 + \hat{w}_2 s_2}_{k i_1} + \hat{w}_3 b_3 \quad (29)$$

Assume $k = f(D_{IIT}; s; b)$. From Equation 29,

$$f(D_{IIT}; s; b) = 1 \frac{(w; \hat{w}; s; b)}{i_1} \quad (30)$$

We know IIT optimizes the weights such that two neural nets become causal abstraction of each other. This is done essentially by confounding on D_{IIT} , where $s; b \perp\!\!\!\perp D_{IIT}$ and $D_{IIT} \perp\!\!\!\perp w$ and $D_{IIT} \perp\!\!\!\perp \hat{w}$.³ This argument necessitates the RHS of Equation 30 depends only on D_{IIT} , s , and b . For LHS to be equal, it also must depend on these parameters, facilitating k as a spurious variable.

The equation itself is the necessary as well as the sufficient condition (i.e. the definition) for causal abstraction which is satisfied when $k = f(D_{IIT}; s; b)$. Thus i) Equality of output of two neural nets under interchange intervention and ii) $k = f(D_{IIT}; s; b)$ together pose as a *necessary and sufficient condition* for causal abstraction between N_1 and N_2

\square

Note: Although this is shown for the above neural network having specific architecture, this holds true regardless of the architecture, as the functional form of RHS and LHS must match.

³ $\perp\!\!\!\perp$ denotes the causal arrow, i.e. by optimizing on D_{IIT} , we obtain both w^* and \hat{w}^*

We assume E and F are encoder and LLM machinery respectively having weights of w_E and w_F . The encoder and LLM machinery are followed by C_1 and C_2 respectively having weights w and \hat{w} . Let us assume w_E is the optimized weight of the encoder when it is fine-tuned with $x \sim D_E$. Further, assume we have done IIT on C_1 and C_2 keeping the encoder frozen.

Following would be the dependency of various weights: i) $w_E = f_1(D_E)$, ii) $w_F = f_2(w_E; D_{IIT}; w; \hat{w})$, iii) $w = f_3(D_{IIT})$ and iv) $\hat{w} = f_4(D_{IIT})$. Without loss of generality we can assume $D_{IIT} = D_E$, as both $s; b$ and x are being sampled from the same dataset. The functional dependencies then boil down to the fact that all the weights are a function of D_{IIT} .

Being a closely trained system with only one dataset $D_E = D_{IIT}$, and from lemma 1, the most generalized version linking the intervened output from F and E will be $F^{INT}(E(s; b)) = (D_{IIT}; s; b)E^{INT}(s; b)$.

Upon the assumption that E and F are two variables (i.e. neurons, composed of all other neurons inside E and F) inside C_1 and C_2 respectively, their intervened output depends only on $(D_{IIT}; s; b)$. Also, E and F are assumed to be inside C_1 and C_2 respectively would mean for any input $(s; b)$, their intervened outputs remain the same. Both of these satisfy the necessary and sufficient requirements for causal abstraction as per Lemma 2. This is complementary to Theorem 2 and its proof, shown by assuming these strong conditions.

C PROMPTS

0-shot Meme Dataset Prompt

Prompt: Is this image offensive? If it is offensive, give a single-line explanation, otherwise simply state that it is 'not offensive'.



Few-shot Meme Dataset Prompt

Prompt 1: Is this meme offensive? Answer briefly. Give 1 line explanation only if it is offensive.



Assistant: This meme is offensive. \neq Explanation goes here.

Prompt 2: Is this image offensive? Answer briefly. Give 1 line explanation only if it is offensive.



Assistant: This meme is not offensive.

Prompt 3: Is this image offensive? Answer briefly. Give 1 line explanation only if it is offensive.



918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971

0-shot SNLI VE Prompt

Prompt: Answer with 'entailment', 'contradiction', or 'neutral' if the hypothesis that *[Insert hypothesis here]* follows the image, contradicts it, or is neutral to it. Also, give a 1-line explanation for your answer.

`images/sample_image_snli.png`

Few-shot SNLI VE Prompt

Prompt 1: Answer with 'entailment', 'contradiction', or 'neutral' if the hypothesis that *[Insert **entailment** hypothesis]* follows the image, contradicts it, or is neutral to it. Also, give a 1-line explanation for your answer.

`images/sample_image_snli_0.png`

Assistant: Entailment. *f*Explanation goes here.

Prompt 2: Answer with 'entailment', 'contradiction', or 'neutral' if the hypothesis that *[Insert **contradiction** hypothesis]* follows the image, contradicts it, or is neutral to it. Also, give a 1-line explanation for your answer.

`images/sample_image_snli_1.png`

Assistant: Contradiction. *f*Explanation goes here.

Prompt 3: Answer with 'entailment', 'contradiction', or 'neutral' if the hypothesis that *[Insert **neutral** hypothesis]* follows the image, contradicts it, or is neutral to it. Also, give a 1-line explanation for your answer.

`images/sample_image_snli_2.png`

Assistant: Neutral. *f*Explanation goes here.

D DATASET AND EXPERIMENTATION

The e-SNLI-VE dataset includes human-annotated explanations for both text and images. For offensive memes, we generated explanations using the Jurassic-1⁴ language model through zero-shot prompting, as detailed in Appendix Section C via its relevant prompts. In this context, the LLM-generated explanations serve as the ground truth. The experiments were conducted on a Kaggle kernel with PyTorch version 2.1.2 and a single P-100 GPU, with a random seed of 42 maintained for all runs. Additionally, baseline VLM models were implemented using PEFT⁵ and LoRA (Hu et al., 2021). The code is available anonymously for review.

Table 6: Train-test splits for e-SNLI-VE and Hateful Memes datasets.

Dataset	Train Split	Test Split
e-SNLI-VE	9000	1000
Hateful Memes	6997	1000

⁴<https://www.ai21.com/blog/announcing-ai21-studio-and-jurassic-1>

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