TOWARDS CAUSAL CONCEPTS FOR EXPLAINING LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

The emergence of large-scale pretrained language models has posed unprecedented challenges in deriving explanations of why the model has made some predictions. Stemmed from the compositional nature of languages, spurious correlations have further undermined the trustworthiness of NLP systems. Thus, there exists an urgent demand for causal explanations to encourage fairness and transparency. To derive more causal, usable, and faithful explanations, we propose a complete framework for interpreting language models by deriving causal concepts. Specifically, we propose a post-hoc method that derives both high-level explanations in terms of concepts and surface-level local explanations from the pretrained model's hidden layer activations. To ensure causality, we optimize for a causal loss that maximizes the Average Treatment Effect (ATE), where we intervene on the concept-level as an innovative substitute to the traditional counterfactual interventions on the surface words. Moreover, we devise several causality evaluation metrics for explanations that can be universally applied. Extensive experiments on real and synthetic tasks demonstrate that our method achieves superior results on causality, usability and faithfulness compared to the baselines. Our codebase is available at `https://anonymous.4open.science/r/CausalConcept`.

1 INTRODUCTION

Over the past few years, NLP models have grown to be more large-scale, more complex, and more expensive to train. Despite their impressive performance in numerous tasks, understanding their decision processes remains difficult, which has become the need of the hour for high-stakes applications. One main challenge is that these models contain a large number of *spurious correlations* – features that are useful for training but not causal, which have become a serious threat (Feder et al., 2021; McCoy et al., 2019; Eisenstein, 2022). Thus, there is a growing and urgent need to interpret the black-box language models in a *causal* and faithful way.

Most of the currently popular NLP explainability methods only discover correlational information, such as induction-based methods in (Ling et al., 2017), explainability-aware architectures in (Rajani et al., 2019), or feature importance scores in (Croce et al., 2019). Outside the NLP domain, there are concept-based methods such as (Kim et al., 2018) that derives high-level concepts as explanations. However, one critical drawback is that these methods do not differentiate between *correlational* and *causal* information. As shown later in our experiments, the derived concept-level explanations have little effect on the final model outputs, which undermines the validity of the explanations.

To derive more causal explanations in NLP, there have been recent attempts utilizing counterfactuals, probing, and Causal Mediation Analysis (CMA). For example, many methods generate input counterfactuals (Wachter et al., 2017; Alvarez-Melis & Jaakkola, 2017). These methods derive local explanations to a specific input instance or word, while it is also desirable to have a global explanation, as it mimics the human reasoning process: humans typically reason globally at the concept level, drawing connections from similar examples and grouping them systematically (Tenenbaum, 1998). The probing methods train an external model to predict some desired properties from the latent representations of the pretrained model (Belinkov, 2022a; Conneau et al., 2018). However, a main limitation is that the probing model and original model are disconnected. Thus, it does not necessarily tell us whether the probed property is indeed involved in the original prediction tasks.

In this work, we propose a complete framework for explaining language models based on high-level concepts that are more causal by construction. We first propose *CausalConcept*, a method to derive both global concepts and its corresponding local explanations that result in high output changes. Thus, our method generates both forms of explanations that complement each other while conforming to the 'mindset' of the model. As a post-hoc approach, our method discovers latent features from the hidden activations as global concepts, providing a flexible representation of attributes within the data. To train the explanation model, we enforce a reconstruction loss such that explanations are faithful and informative. Crucially, to ensure that the generated explanations have a high impact on the output predictions, we propose a causal loss to maximize the corresponding output changes with respect to the concepts. Instead of generating input counterfactuals, we perturb on high-level concepts in the hidden activations, which correspond to features in the input distribution as ensured by an auto-encoding loss. Thus, our intervention can be seen as generation of *latent* counterfactuals. To get the corresponding instance-level explanations, we map the concepts back to the input via visualization strategies and token importance scores. We then propose causality metrics that stem from theoretical definitions of treatment effects in literature (Pearl, 2009). Finally, we construct reliable and extensive experiments that prove the causality, usability, and faithfulness of our method.

2 Related Work

As denoted by Feder et al. (2021), causality shows a promising path forward for NLP researchers, which can offer insights into the inner workings of the model. Most current methods attempt to causally explain an NLP model by generating *counterfactual* inputs. For example, Alvarez-Melis & Jaakkola (2017) use a Variational autoencoder (VAE) to generate counterfactuals and conduct causal analysis. Veitch et al. (2021) conduct stress tests by perturbing input words. Wu et al. (2021) construct a low-cost counterfactual generator for downstream applications. Apart from NLP, there have been attempts to generate counterfactual inputs using disentangled VAEs, such as (O'Shaughnessy et al., 2020), which has a similar motivation to our work, but is still confined on the input space. Such counterfactual explanations, however, require extra caution to hold rigorously in causality, as causal and correlational relationship exist among input features and we cannot explicitly obtain such casual graphs or correlations in practice. To overcome this challenge, we propose to perturb on the intermediate hidden layer, thus assuming the independence between latent concepts.

Another line of work uses probing and Causal Mediation Analysis (CMA) to explain black-box models. Probing (Conneau et al., 2018; Belinkov et al., 2020) methods train an external model - a *probe* - to predict some properties of interest from the latent representations. To further investigate causal effects of the features learned from probing, Elazar et al. (2021) assess the influence of a causal intervention by removing a feature. However, subsequent work Barrett et al. (2019) shows that such methods generalize poorly to unseen samples. Moreover, as Belinkov (2022b) points out, the disconnect between the probing model and the original model may result in the properties not being utilized in the original model's prediction task. CMA Pearl (2022) measures the change in an output following a counterfactual intervention in an intermediate variable, or mediator. The work of Vig et al. (2020) is an application of CMA in NLP, where gender bias is examined by changing pronouns in the input. We argue that both probing and CMA require human-constructed features (e.g., linguistic, gender features), which require expertise on the datasets and tasks. Thus, it might be beneficial for inexperienced users to develop unsupervised explanation features.

Outside the NLP domain, Harradon et al. (2018) attempt to intervene in an unsupervised way on the hidden space by constructing several even-spaced VAEs throughout a CNN, but it only trains with a reconstruction loss instead of explicitly optimizing for causality. Similarly utilizing unsupervised features, concept-based methods such as (Kim et al., 2018) have been a popular interpretability method as it derives user-friendly explanation units. Yeh et al. (2020) discover concepts in the intermediate layer with a classic bottleneck-shaped network. They further propose an adapted Shapley value metric to evaluate concept importance by quantifying how much each individual concept contributes to the final completeness score. Koh et al. (2020) learn high-level concepts and experiment with how the user could interact to edit concepts during test time. However, as we will see in our experiments (§5), because the existing concept-based methods do not differentiate between correlational and causal information, their performance on NLP tasks is problematic: the discovered concepts often have little impact on the final prediction of the model. Especially, on complex transformer models with stronger confounding effects brought by pretraining, their performances may further decrease.

3 METHODOLOGY

Setup and Symbols: We can view a pretrained neural model as a composite of two functions, divided at an intermediate layer: the first part $\phi(\cdot)$ maps the input text $\mathbf{x}$ to a hidden representation $\phi(\mathbf{x})$, and the second part $\psi(\cdot)$ maps $\phi(\mathbf{x})$ to classification probabilities $\psi(\phi(\mathbf{x}))$. Without loss of generality, we assume that, for an input data point $\mathbf{x}$, which consists of T tokens $[x_1, \ldots, x_T]$, $\phi(\mathbf{x})$ can be represented as a concatenation of $[\phi(x_1), \ldots, \phi(x_T)]$, where each $\phi(x_t) \in \mathbb{R}^d$ denotes a representation of an input token x_t. Depending on the model architecture, $\phi(x_t)$ can be encoded from a local receptive field as in convolutional nets or a global one as in Transformers (Vaswani et al., 2017). As the model being interpreted has been trained with seed initialization, $\phi(\mathbf{x})$ can be seen as a transformed version of input $\mathbf{x}$. Thus, we could extract n concepts $\mathcal{C} = \{\mathbf{c}_1, \ldots, \mathbf{c}_n\}$ as directions on the hidden space that represent different linguistic or semantic features of the textual input distribution. For a visualization of the model architecture, we refer readers to Appendix A.

3.1 INTRODUCING CAUSALITY

We argue that current unsupervised concept extraction approaches such as (Yeh et al., 2020) only finds hidden directions based on *correlational* information, instead of causal. This issue becomes **crucial when interpreting large pretrained models** which have become the norm in NLP as such they are also called the foundation models (Bommasani et al., 2021). This is because pretraining can bring in more task-irrelevant information, such as orthographic and grammatical information that is not informative, and may create biases and spurious correlations (McCoy et al., 2019; Tu et al., 2020; Gardner et al., 2021). Thus, in the experiment section (§5), we will observe that the discovered concepts based on current methods often have little to no impact on output predictions, especially when interpreting pretrained models like BERT (Devlin et al., 2018) and T5 (Raffel et al., 2020).

The failure cases can be explained with the causal graph in Fig. 1. A real-life analogy is that, while the hot weather (X) creates high demand for ice cream (E), it also produces intense UV light exposure (Z), thus causing more sunburns (Y). However, it is obvious that high ice cream sales (E) do not cause sunburns (Y). In the case for IMDB movie sentiment prediction, an input X (*e.g.,* a movie review) may contain both causal information Z to the model predictions, which is mostly task-relevant (*e.g.,* adjectives like 'awesome'), and extra information E (*e.g.,* movie name) which does not affect the model prediction Y (sentiment). In pretrained language models, the hidden activation space consists of both E and Z. Although only Z truly affects prediction Y, E and Z may also be correlated due to the confounding effects brought by X. However, a traditional concept mining model does not differentiate between them and considers both as valid, which is problematic as E is only correlational.

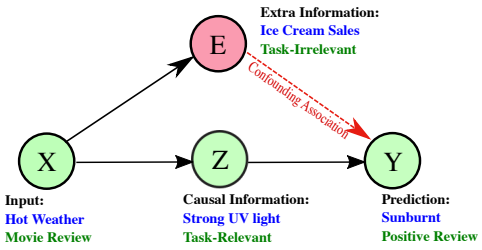

Figure 1: Causal graph illustration.

As a post-hoc explainability method, we are enforcing explanations to be more causal with respect to the model predictions, instead of to real-life scenarios. Thus, if a discovered feature is task-irrelevant in real life but misused by the model for predictions, it is deemed as a causal explanation because it captures why the model made predictions. Another important assumption is that, as the concept vectors exist on the space of $E \cup Z$, there is no causal relationships among the concepts $\{\mathbf{c}_1, \ldots, \mathbf{c}_n\}$.

In causal analysis, Individual Treatment Effect (ITE) and Average Treatment Effect (ATE) are defined to measure the effect of interventions in randomized experiments. Given a binary treatment variable T that indicates whether an intervention is performed, ATE and ITE are defined with the do-operation:

$$\text{ITE}(x) := \mathbb{E}[y|\mathbf{X} = x, \text{do}(T = 1)] - \mathbb{E}[y|\mathbf{X} = x, \text{do}(T = 0)]; \qquad \text{ATE} := \mathbb{E}[\text{ITE}(x)] \tag{1}$$

It is, however, difficult to calculate ITE and ATE in real life, as randomized controlled interventions (a requirement for the do-operation) are either expensive or impractical. Currently, most NLP explainability models intervene by producing input counterfactuals, such as changing the gender pronouns (Vig et al., 2020). Such word-level counterfactuals only cover a limited set of interventional space (such as synonyms and antonyms). To tackle this problem, we propose to generate counterfactuals in the latent representation (i.e., concept) space.

In our case, a concept $\mathbf{c}_i$ is discovered as a direction in the latent space, corresponding to a feature in the input distribution. If $\mathbf{c}_i$ is used by the model for prediction, the removal of $\mathbf{c}_i$ should negatively influence the model's prediction accuracy. Thus, we could define the treatment to be the removal of a specific feature by setting $\mathbf{c}_i = \mathbf{0}$. While another possible strategy is to add noise to a specific feature $\mathbf{c}_i$, such as $\mathbf{c}_i := \mathbf{c}_i + \epsilon$, with $\epsilon \sim N(\boldsymbol{\mu}, \sigma)$, we omit such interventions for two reasons. First, factors such as the noise type and its distribution would introduce bias into the distribution of concepts. Second, such interventions may not mimic real-life scenarios, as they introduce artificial noise which may not correspond to a plausible text. In contrast, simply omitting a concept closely resembles the real-life intervention of omitting a factor in the input, such as removing words related to a plot in the movie review. Similar ATE approximations are proposed in prior work (Goyal et al., 2019). Thus, we customize definitions of ITE for a concept $\mathbf{c}_i$ and ATE for a concept set $\mathcal{C}$ as follows.

$$\text{ITE}(\mathbf{x})_i = \mathbb{E}[y|\mathbf{X} = \mathbf{x}, \mathbf{c}_i = \mathbf{0}] - \mathbb{E}[y|\mathbf{X} = \mathbf{x}, \mathbf{c}_i = \mathbf{c}_i]; \qquad \text{ATE} = \mathop{\mathbb{E}}_{\mathbf{c}_i \in \mathcal{C}}[\text{ITE}(\mathbf{x})_i] \tag{2}$$

3.2 Generation of Concept Vectors

To encode hidden activations $\phi(x_t) \in \mathbb{R}^d$ into n concepts $\mathcal{C} = \{\mathbf{c}_1, \ldots, \mathbf{c}_n\}$, we first initialize the concepts uniformly as $\mathbf{c}_i \sim \mathcal{U}(-0.5, 0.5) \in \mathbb{R}^d$. Then, similar to concept extraction models (Kim et al., 2018), we approximate the distribution over the concepts by encoding activations into concept probabilities as $p_{\mathcal{C}}(x_t) = [p_c^1(x_t), \ldots, p_c^n(x_t)]$. For each concept $\mathbf{c}_i$, the similarity is calculated as $p_c^i(x_t) = \text{TH}((\phi(x_t)^\top \mathbf{c}_i), \beta)$, where TH is a threshold function that forces all inputs smaller than β to be 0.[1] To get the concept distribution for the entire sequence $\mathbf{x}$, we concatenate the token-level distributions: $p_{\mathcal{C}}(\mathbf{x}) = [p_{\mathcal{C}}(x_1), \ldots, p_{\mathcal{C}}(x_T)] \in \mathbb{R}^{T \times n}$. In the special case of the last layer in BERT, $\phi(\mathbf{x})$ consists of only the [CLS] representation, which is used to construct $p_{\mathcal{C}}(\mathbf{x})$. For T5, $\phi(\mathbf{x})$ represents the decoder state at the final layer which is used to predict the first token of the sequence.

Next, we attempt to reconstruct the original hidden activations $\phi(\mathbf{x})$ from $p_{\mathcal{C}}(\mathbf{x})$ with a 2-layer perceptron g_θ such that $g_\theta(p_{\mathcal{C}}(\mathbf{x})) \approx \phi(\mathbf{x})$. The simple 2-layer perceptron gives enough parameters for reconstruction, while not involving too much complexity that may introduce further confounding. To train the bottleneck-shaped perceptron in an end-to-end way, we optimize the following losses:

• **Reconstruction loss:** To faithfully recover the original DNN model's predictions, we optimize a surrogate loss with cross-entropy (CE) defined as:

$$\mathcal{L}_{\text{rec}}(\theta, \mathcal{C}) = \text{CE}\Big(\psi\big(\phi(\mathbf{x})\big), \psi\big(g_\theta(p_{\mathcal{C}}(\mathbf{x}))\big)\Big) = -\sum_{b \in \mathcal{B}} \psi\big(\phi(\mathbf{x})\big)_b \log\big(\psi(g_\theta(p_{\mathcal{C}}(\mathbf{x})))_b\big) \tag{3}$$

where $\mathcal{B}$ is the set of class labels and $\psi(.)_b$ denotes the prediction score corresponding to label b. While we allow for general loss of information through concept distributions, we aim to ensure that the information crucial for prediction is preserved by reconstructing the same label distributions.

• **Regularization loss:** To ensure that the concepts derived are more user-friendly, we regularize them such that each concept vector corresponds to actual examples and the concepts are distinct from each other. This can be achieved by maximizing the similarity of a concept $\mathbf{c}_i$ to the actual tokens in its top-N neighborhood $\mathcal{R}_i$ (measured in the activation space) and also minimizing the similarity between the concepts themselves (Yeh et al., 2020). Formally,

$$\mathcal{L}_{\text{reg}}(\mathcal{C}) = -\lambda_1 \frac{\sum_{i=1}^{n} \sum_{x_t \in \mathcal{R}_i} \mathbf{c}_i^\top \phi(x_t)}{nN} + \lambda_2 \frac{\sum_{i_1 \neq i_2} \mathbf{c}_{i_1}^\top \mathbf{c}_{i_2}}{n(n-1)} \tag{4}$$

The assumptions are: (*i*) if the concepts are sufficient, we will be able to maximize the information overlap between concept vectors and hidden activations; (*ii*) as they exist in the same activation space, the concepts are independent factors by assumption (as they do not influence each other).

• **Auto-Encoding loss:** To learn the distribution of $\phi(\mathbf{x}) \in \mathbb{R}^d$ by using our surrogate model, in which the discovered concepts $p_{\mathcal{C}}(\mathbf{x})$ could faithfully reconstruct $\phi(\mathbf{x})$ by serving as latent features, we force the reconstructed embeddings to be close to the input embeddings. For this, we define the following mean-squared error (MSE) loss:

$$\mathcal{L}_{\text{enc}}(\theta, \mathcal{C}) = \text{MSE}\Big(\phi(\mathbf{x}), g_\theta(p_{\mathcal{C}}(\mathbf{x}))\Big) = \frac{1}{d}||\phi(\mathbf{x}) - g_\theta(p_{\mathcal{C}}(\mathbf{x})||_2^2 \tag{5}$$

[1] We perform normalization to ensure that $\phi(x_t)$ and $\mathbf{c}_i$ are unit vectors, and use softmax such that $p_c^i(x_t)$ are probabilities.

• **Causality loss:** To disentangle concept directions that are more causal, we design a loss that forces the discovered concepts to have a greater influence on the final prediction. Following Eq. 2, our intuition is that a less causal concept should have a treatment effect close to 0. Therefore, we optimize an approximation of the ATE through the following causality loss.

$$\mathcal{L}_{\text{cau}}(\theta, \mathcal{C}) = -\sum_{\mathbf{x}_j \in \mathcal{D}} \sum_{\mathbf{c}_i \in \mathcal{S}} \left| \psi\Big(g_\theta(p_{\mathcal{C}}(\mathbf{x}_j) | \mathbf{c}_i = \mathbf{0})\Big) - \psi\Big(g_\theta(p_{\mathcal{C}}(\mathbf{x}_j) | \mathbf{c}_i = \mathbf{c}_i)\Big) \right| \approx -|\text{ATE}| \tag{6}$$

Here, $\mathcal{S}$ denotes a set of concepts to remove, $\mathcal{S} \subseteq \mathcal{C}$, which can be selected in several ways: perturbing on all concepts ($\mathcal{S} = \mathcal{C}$), selecting randomly from $\mathcal{C}$, or selecting the most similar concept with the input $\mathcal{S} = \{\mathbf{c}_i : i = \arg\max_i p_c^i(\mathbf{x}_j)\}$. Through experiments, we have found that selecting randomly yields the best performance. As we perturb on all inputs $\mathbf{x}_j \in \mathcal{D}$, the dataset $\mathcal{D}$ will serve both as the treatment group and the nontreatment group, ensuring that no divergence is present. As the $|\psi(\cdot|\mathbf{c}_i = \mathbf{0}) - \psi(\cdot|\mathbf{c}_i = \mathbf{c}_i)|$ term in Eq. 6 approximates the effect that the a random $\mathbf{c}_i$ has on a data point $\mathbf{x}_j$, averaging over $\mathcal{S}$ then serves as an approximate of the ITE over $\mathcal{C}$. Therefore, the sum over $\mathcal{D}$ will approximate its expectation, which resembles the ATE. Therefore, minimizing the designed causality loss is a close approximation to maximizing the expected ATE of concepts on the final predictions. Intuitively, this loss encourages the concepts to incorporate directions that result in more significant changes in the output predictions.

Total loss: Finally, the overall loss function that we minimize becomes:

$$\mathcal{L}(\theta, \mathcal{C}) = \mathcal{L}_{\text{rec}}(\theta, \mathcal{C}) + \mathcal{L}_{\text{reg}}(\mathcal{C}) + \lambda_e \mathcal{L}_{\text{enc}}(\theta, \mathcal{C}) + \lambda_c \mathcal{L}_{\text{cau}}(\theta, \mathcal{C}) \tag{7}$$

where λ_e, λ_c are the weights for the auto-encoding loss and the causal loss respectively. In practice, we only turn on the causal loss after a certain number of epochs (usually half of the overall number of epochs) to make sure that the surrogate model first learns to faithfully reconstruct from the set of concepts before optimizing for the causal ones. This is because learning the two conflicting objectives at once will usually result in low accuracy. We also note that some contextual information is still needed to maximize the accurate reconstruction of hidden activations $\phi(\mathbf{x})$. Thus, the causality loss is enforced on all concepts except the last one $\mathbf{c}_n$, which is used as a 'context concept'. During model inference, the last (noncausal) concept is unused.

• **Post-processing**: While the number of concepts n is user-selected, as in many topic models, it is an inherent flaw as it requires a certain level of domain expertise. For example, in a movie review dataset with only 2 output classes, if an unfamiliar user sets n to 200, the model will naturally discover many noisy concepts and only a few useful ones. To ensure that the noisy concepts are eliminated, we post-process the concepts and filter out the unused ones (with a change in ATE close to 0). Thus, a more desirable number of concepts is returned even if the user provides an overestimate of n. In our experiments, we see that, after filtering, the model always achieves a better or same prediction-reconstruction performance as before. However, even with this post-processing, specifying too large a number of concepts can still be dangerous as it harms the concept model's training process.

3.3 Mapping Concepts Back to Word Tokens

As language model explanations require mapping concepts to the discrete input tokens, when the receptive field of a concept is larger than token-level, we employ several techniques to interpret them. For explaining BERT, when we use the last layer (for which [CLS] representation is used), we employ the transformer visualization method proposed in (Chefer et al., 2021) to map back from the [CLS] activation concepts to input tokens. Specifically, Chefer et al. (2021) visualizes classifications with a combination of layer-wise propogration (LRP), gradient backpropagation, and layer aggregation with rollout. As a result, for each sample $\mathbf{x}$ and concept $\mathbf{c}_i$, we will go from having only one concept similarity score $p_c^i(\mathbf{x})$ to having a list of token importance scores $s_1(\mathbf{c}_i), \ldots, s_T(\mathbf{c}_i)$. For the intermediate layers of BERT, we simply use the corresponding tokens as the representations (*i.e.,* $\phi(x_t)$ and its corresponding $p_c^i(x_t)$) are already at the token level. For CNNs, we employ the GradCam (Selvaraju et al., 2017), which rolls out the gradients to produce scores for each token.

4 EXPERIMENT SETTINGS

4.1 DATASETS AND CLASSIFICATION MODELS

We mainly test the effectiveness of our method with two standard text classification datasets: IMDB (Maas et al., 2011) and AG-news (Zhang et al., 2015). The IMDB dataset consists of movie reviews labeled with positive or negative sentiment. The AG-news dataset consists of news articles categorized with 4 topics. Table 6 in Appendix C gives a summary of the datasets. We explain three classification models: (*i*) a 6-layer transformer encoder trained from scratch, (*ii*) a pre-trained BERT with finetuning, (*iii*) a pre-trained T5 (Raffel et al., 2020) with finetuning.

4.2 EVALUATION MEASURES

We evaluate the explanation methods based on three important aspects as described below.

• **Causality:** Causality is an important consideration to evaluate explainability methods, especially where spurious correlations are strong. Doshi-Velez & Kim (2017) state: "Causality implies that the predicted change in output due to a perturbation will occur in the real system". Thus, we propose measures to quantitatively assess whether a high-level concept directly affects the final predictions. Following the definitions introduced in §3.1, we define the individual causal effect (ICE_i) for a concept $\mathbf{c}_i$ and the **average causal effect (ACE)** for a concept set $\mathcal{C}$ as:

$$\text{ICE}_i := \sum_{\mathbf{x}_j \in \mathcal{D}_{\text{test}}} |\psi\big(g_\theta(p_\mathcal{C}(\mathbf{x}_j))\big) - \psi\big(g_\theta(p_{\mathcal{C}\setminus\{i\}}(\mathbf{x}_j))\big)|; \qquad \text{ACE} = \frac{1}{|\mathcal{C}|}\sum_{\mathbf{c}_i \in \mathcal{C}} \text{ICE}_i \tag{8}$$

Similarly, we include the accuracy change with and without a specific concept $\mathbf{c}_i$ to measure causality. The intuition is that, if a concept $\mathbf{c}_i$ is a cruicial factor used by the model to make predictions, omitting it will result in high accuracy changes. Denoting $\text{Acc}(\mathcal{C}) = \text{Accuracy}(\psi\big(g_\theta(p_\mathcal{C}(\mathbf{x}))\big), \phi(\psi(\mathbf{x}))$:

$$\Delta\text{Acc}_i = |\text{Acc}(\mathcal{C}) - \text{Acc}(\mathcal{C} \setminus \{i\})|; \qquad \Delta\text{Acc} \quad = \frac{1}{|\mathcal{C}|}\sum_{\mathbf{c}_i \in \mathcal{C}} \Delta\text{Acc}_i \tag{9}$$

Such evaluation measures can be globally applied to other explainability methods with feasible do-interventions. A higher ACE and ΔAcc represent a higher change in model prediction, thus a more impactful set of concepts. To measure causal effects for an individual token x_t, we also devise a "causal score" metric. We take the top-3 most similar concepts $\mathcal{C}_{\text{top}}$ to the input $\mathbf{x}$ using the normalized similarity score $p_\mathcal{C}(\mathbf{x})$. For each concept $\mathbf{c}_i \in \mathcal{C}_{\text{top}}$, the transformer visualization method (§3.3) produces normalized token importance scores $\{s_1(\mathbf{c}_i), \ldots, s_\text{T}(\mathbf{c}_i)\}$. The **causal importance** score for a token x_t is defined as: $\text{CI}(x_t) = \sum_{c_i \in \mathcal{C}_{\text{top}}} p_c^i(\mathbf{x}) s_t(\mathbf{c}_i)$.

• **Usability:** Proposed in (Doshi-Velez & Kim, 2017), an important desiderata of explainability is to make sure that it provides usable information that assists users to accomplish a task. With the causal concepts being more reliable, we expect that end-users can better understand the model's reasoning process, which can be useful for debugging and fairness. We include visualizations and human studies to test it qualitatively.

• **Faithfulness:** Faithfulness evaluates whether our surrogate model can accurately mimic the original model's prediction process. In other words, we want to make sure that our captured concept probabilities $p_\mathcal{C}(\mathbf{x})$ can recover the original model's predictions $\psi\big(\phi(\mathbf{x})\big)$. We report the **recovering accuracy** for the set of concepts $\mathcal{C}$:

$$\text{RAcc} = \frac{1}{|\mathcal{D}_{\text{test}}|}\sum_{\mathbf{x}_j \in \mathcal{D}_{\text{test}}} \mathbb{1}\Big(\psi\big(\phi(\mathbf{x}_j)\big) = \psi\big(g_\theta(p_\mathcal{C}(\mathbf{x}_j))\big)\Big)$$

As we view the original deep network as a composite function: $f = \phi \circ \psi$, we can derive a concept-based explanation method in two corresponding steps: (*i*) generate high-level concepts (low-dimensional subspaces) that correspond to vectors in the hidden space $\phi(\mathbf{x})$, and (*ii*) from the hidden activations, map the concepts back to input tokens by propagating relevance scores to facilitate instance-level explanation. See Appendix A for a visualization of the method.

Table 1: Faithfulness (RAcc↑) and causality (ACE↑, ΔAcc ↑) evaluation of different text classification methods.

Dataset	Model	Cls.Acc	Metric	β-TCVAE	K-means	PCA	ConceptSHAP	CausalConcept
IMDB	Transformer	81.74%	ACE	0.037	0.047	0.001	0.031	**0.150**
			ΔAcc	1.24%	2.59%	0.01%	1.30%	**11.06%**
			RAcc	52.08%	83.64%	85.18%	84.36%	**88.78%**
	BERT	89.14%	ACE	0.057	0.038	0.002	0.050	**0.104**
			ΔAcc	4.10%	1.56%	0.02%	0.06%	**9.47%**
			RAcc	93.86%	**98.69%**	96.68%	95.84%	94.53%
	T5	72.98%	ACE	0.000	0.025	0.000	0.000	**0.094**
			ΔAcc	0.00%	1.06%	0.02%	20.21%	**38.34%**
			RAcc	0.00%	75.85%	98.86%	60.20%	**99.50%**
AG	Transformer	88.33%	ACE	**0.049**	0.044	0.000	0.000	0.045
			ΔAcc	6.62%	0.07%	0.03%	0.00%	**7.12%**
			RAcc	98.90%	98.16%	**99.99%**	73.01%	99.50%
	BERT	93.75%	ACE	0.044	0.028	0.001	0.025	**0.058**
			ΔAcc	5.32%	7.15%	0.01%	4.44%	**10.54%**
			RAcc	92.30%	86.83%	99.79%	93.46%	**99.90%**
	T5	94.30%	ACE	0.000	0.011	0.000	0.000	**0.054**
			ΔAcc	0.00%	1.49%	0.01%	0.00%	**52.20%**
			RAcc	0.00%	24.87%	97.38%	0.00%	**99.46%**

Table 2: Generated concept keywords with ACE from AG-News dataset, BERT model.

Method	ACE	Keywords
ConceptSHAP	0.000	one, two, gt, new, cl, lt, first, world, mo, last, b, san, tuesday, soccer, time,nhl, Australia, red, bryant
ConceptSHAP	0.000	first, new, red, world, Yankees, Australia, giants, nl, as, two, one, ga, last, b, u, tuesday, quo, men
CausalConcept	0.108	update, us, fed, wal, op, u, stocks, oil, dollar, delta, hr, ex, Wednesday, world, percent, crude
CausalConcept	0.151	red, NBA, football, Yankees, sports, NFL, team, baseball, olympic, league, game, season, coach

4.3 Baselines and Hyperparameters

As fair comparisons to our method, we can only consider unsupervised feature discovery algorithms. Thus, we use conceptSHAP (Yeh et al., 2020) as a baseline. To compare to VAE methods, we include the disentanglement VAE (β-TCVAE) by Chen et al. (2018). Moreover, we also include comparisons to popular non-parametric clustering techniques, including PCA and k-means to discover directions on the hidden space.

The full list of hyperparameters used for training the CausalConcept model can be found in Appendix C. Briefly, we recommend using loss coefficients: regularizer similarity loss $\lambda_1 = 0.1$, regularizer orthogonality loss $\lambda_2 = 0.5$, auto-encoding loss $\lambda_e = 1$, and causal loss $\lambda_c \in [1, 3]$, where λ_c (causal coefficient) depends on the level of confounding within the dataset. Perturbation is performed on the most similar concept to the input. All experiments are conducted on the penultimate layer with 10 concepts. The hyperparameters are chosen as an optimal default through grid search. To make the comparison fair, PCA, K-means, and β-TCVAE also use 10 dimensions to encode.

5 Results and Analysis

To first provide a sanity check for our method, we conduct a toy experiment with a synthetic graphic dataset where the level of confounding can be controlled with ground truth concepts. Appendix B gives details of the experiment. The results show that our method discovers concepts that align with human understanding and consistently outperforms the baseline by deriving more causal features. As confounding levels in the dataset increase, the performance gap also widens.

5.1 Results on text classification datasets

The experiment results on text classification datasets are presented in Table 1. Concepts discovered by the baseline methods lead to tiny changes in prediction outputs, which undermine their reliability. On the contrary, our method is able to derive concepts that induce a much higher change in predictions (ACE, ΔAcc), while maintaining high accuracy (RAcc). On all models, especially pretrained BERT and T5, CausalConcept induces a larger prediction change than all the baseline methods, while maintaining faithfulness. This observation consolidates our intuition that pretrained complex language models with more confounding correlations can benefit more from causality. As a seq2seq text generation model, the pretrained T5 is especially hard to learn for the surrogate models, as the

Method	Visualization
ConceptSHAP	dream team leads spain 44 - 42 at halftime athens, greece - as expected, the u.s. men's basketball team had its hands full in a quarterfinal game against spain on thursday...
CausalConcept	dream team leads spain 44 - 42 at halftime athens, greece - as expected, the u.s. men's basketball team had its hands full in a quarterfinal game against spain on thursday ...

Figure 2: Qualitative comparison from AG-News dataset: "World" news misclassified as "Sports" by BERT.

output vocabulary has a class size of 32,128. For better calculation of ACE, we simplify outputs by filtering to only the classification classes (e.g., words "Positive", "Negative" for IMDB) and summing all other vocab probabilities as "Other". Some models collapse completely in this case. CausalConcept, however, excels in maintaining both faithfulness and a large effect on final predictions.

To qualitatively examine the discovered concepts by our method, we take an example of BERT on AG-News. In Table 2, 2 out of 10 concepts from both CausalConcept and the baseline are shown as examples. For ConceptSHAP, both concepts picked had low ACE, corresponding to less changes in output predictions. Although vaguely hinting at the category "Sports", they consist of words that are less indicative, such as "one", "two", and "new". When looking at CausalConcept discovered topics, the first talks about "World", especially in the global finance topic. The second clearly points to the American sports leagues, indicating category "Sports". Thus, instead of merely pointing to class information, the concepts discovered contain more information that aligns with human understandable concepts. The ACE score shown here is also consistent with what humans perceive as important and causal words to the classification, thus indicating the metric's validity. Moreover, in Appendix F.1, we show more concepts discovered in former layers of BERT, which shows that the concepts are not only separable in semantic meanings, but also syntactical information (such as nouns and adjectives).

The **usability** of our method could be visualized with the examples in Fig. 2, which shows the same failure case (labeled as "World" news but misclassified as "Sports") highlighted with the top concept discovered. ConceptSHAP discovers a top concept related to the keywords "leads", "as expected", or "on thursday", which are not informative as to why the model classified this input as Sports news. On the contrary, CausalConcept could precisely point out why it wrongly predicted "Sports": BERT is looking at keywords such as "dream team", "game", and country names. Such examples show the potential of our CausalConcept being used in understanding the model's failure processes, which we further investigate in §5.3 with a carefully designed human study.

5.2 Ablation Study

To ensure that the designated 4 objectives behave as expected, we conduct ablation studies for BERT on AG-News and report the results in Table 3. As observed, eliminating prediction loss leads to a large decrease in RAcc, resulting in an unfaithful model. Thus, even though the model leads to large accuracy changes, the results cannot be trusted. Without auto-encoding loss or regularizer loss, the model has lower performances both in faithfulness and causality. Without causality loss, RAcc is the highest, indicating an accurate reconstruction of the original predictions. However, the discovered set of concepts results in low output changes. Finally, our CausalConcept method discovers a set of concepts that both generate high output changes and maintain a good level of faithfulness.

Table 3: Ablation on BERT for IMDB with faithfulness (RAcc) and causality (ACE, ΔAcc) evaluation.

Method	RAcc ↑	ACE ↑	ΔAcc ↑
No Auto-Encoding Loss	93.46%	0.028	6.11%
No Prediction Loss	68.00%	0.035	**17.41%**
No Regularizer Loss	95.76%	0.041	6.23%
No Causality Loss	**99.92%**	0.029	2.95%
CausalConcept	99.90%	**0.058**	10.54%

5.3 Human study

To validate that CausalConcept can identify words that are more causal than the baseline ConceptSHAP, we design the following human study setup: 100 randomly selected examples from AG's testset are shown, where each example consists of the text input and the model's prediction. The annotator is asked to select up to three most causal words for the predicted label. We collect annotations from 4 different annotators proficient in English in order to obtain a diverse set of causal keywords. We consider the keywords selected by the annotators to be the ground-truth, and calculate the average Causal Importance score (CI) (§4.2) for all unique words (superset) selected by the annotators, with

CausalConcept	google shares, once devalued, just may be winners after all wall street, which forced google, the internet search engine, to sharply lower the price of its shares in its initial public offering in august, has decided that the company is worth a lot more today than it was then.		
Human 1	devalued, shares, price;	**Human 2**	devalued, shares, google

Figure 3: Example sentence in the human study: "Business" news correctly classified.

the baseline and our method trained on the penultimate layer of BERT. Details about how the human study is conducted can be found in Appendix E.

Table 4: Human study for causality/usability evaluation.

	# Examples	Cohen κ	ConceptSHAP (CI)	Our (CI)
Total	100	0.41	0.41	**0.83**

Table 4 shows the results. The CI produced by our model for the superset of causal keywords selected by both the annotators is 2 times higher than the baseline, demonstrating that our model is looking at the right (causal) tokens. We find that the annotators have a Cohen's Kappa agreement of 0.41, which is considered as moderate agreement (Landis & Koch, 1977). This shows that even though the annotators prefer slightly different keywords as causal, our model assigns a higher score to keywords that the annotators find causal compared to the baseline. Such ability to identify causal tokens gives CausalConcept the potential to be used for debugging applications. For example, Fig. 3 shows a correctly classified "Business" news article. The human annotators, with a small disagreement, provide 4 unique keywords, which are all highlighted by the CausalConcept method. This indicates that the method can cover the causal keywords preferred by all annotators.

5.4 Hyper-parameter comparisons

We perform further studies on the two most important hyperparameters: the layer(s) to interpret and number of concepts. We conduct text experiments and evaluate in terms of both causal effect and concept quality. In this section, we summarize our main findings and refer the readers to Appendix F for details about the experiments and results (charts and wordcloud visuals).

For **layer-wise comparisons**, we experiment on the 3rd, 6th, 9th, and 12th layer respectively, all with 10 concepts. In terms of causal effects, the intermediate layers (3, 6 and 9) have a higher ACE (around 0.150), while the penultimate layer (12) has a lower ACE (around 0.058). This is because the concepts discovered at the penultimate layer are sentence-level (using the [CLS] token), while the intermediate layer concepts are token-level. Thus, the sentence-level concepts have less fine-grained control. In terms of topic quality, the later layers tend to discover more coherent concepts, where each concept mostly corresponds to one class label. The beginning layers, on the contrary, tend to discover concepts that are more abstract with mixed class labels. The earlier layers can also discover lexical concepts, such as concepts with only nouns or adjectives. Similarly, Dalvi et al. (2021) find that BERT finds more lexical information in the earlier layers. This interesting observation could lead to future studies in investigating how information flows through different layers in BERT.

For **number of concepts**, we experiment with 3, 5, 10, 50, and 100 concepts on the penultimate layer. We find that a concept number close to the number of output classes usually gives higher prediction changes, while increasing the number results in higher recovering accuracy. When the number of concepts becomes larger, concepts usually become more coherent, although the performance will decrease, as too large a number of concepts introduces more noise into the training process.

6 Conclusions

We have proposed a complete framework to derive impactful concepts that explain a black-box language model's decisions. Our framework addresses 3 important challenges in NLP explainability: (*i*) **Causality:** it derives concepts that generate high output changes and minimize confounding explanations through a causal loss objective. (*ii*) **Counterfactuals:** it proposes an innovative substitute to the traditional input counterfactual. By producing latent counterfactuals that are designed to remove features within input texts, we avoid the input space search. The concern of interpreting the hidden activations is also addressed by incorporating visualization methods (*iii*) **User-friendly:** as demonstrated with visualizations and human studies, CausalConcept leads to human-friendly explanations in NLP tasks that contain high-level text attributes and semantically meaningful concepts.

7 Ethical Considerations and Broader Impacts

CausalConcept demonstrates the potential to play an important role in practical scenarios such as debugging and transparency. As AI ethics have become a major concern in real-life applications, such explanations can help users better identify bias and promote fairness. As a future venue to our work, we believe that our framework will set a good foundation for future research on causal NLP explainability methods, especially those that hope to derive human-friendly explanations. As for potential concerns, CausalConcept only encourages causality in post-hoc model explanations and should serve as an assistive tool instead of being accepted as ground-truth. Thus, to improve it further, a similar causal objective could be used to address spurious correlations during training. It also has the potential of being carried over to other domains, such as vision or tabular tasks. The high-level attributes in the hidden space can also be used in downstream applications to provide better controllability for the users.

References

David Alvarez-Melis and Tommi S Jaakkola. A causal framework for explaining the predictions of black-box sequence-to-sequence models. *arXiv preprint arXiv:1707.01943*, 2017.

Maria Barrett, Yova Kementchedjhieva, Yanai Elazar, Desmond Elliott, and Anders Søgaard. Adversarial removal of demographic attributes revisited. 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)*, pp. 6330–6335, 2019.

Yonatan Belinkov. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219, March 2022a. doi: 10.1162/coli_a_00422. URL https://aclanthology.org/2022.cl-1.7.

Yonatan Belinkov. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219, 2022b.

Yonatan Belinkov, Nadir Durrani, Fahim Dalvi, Hassan Sajjad, and James Glass. On the linguistic representational power of neural machine translation models. *Computational Linguistics*, 46 (1):1–52, 2020. doi: 10.1162/coli_a_00367. URL https://aclanthology.org/2020.cl-1.1.

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.

Hila Chefer, Shir Gur, and Lior Wolf. Transformer interpretability beyond attention visualization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 782–791, 2021.

Ricky TQ Chen, Xuechen Li, Roger B Grosse, and David K Duvenaud. Isolating sources of disentanglement in variational autoencoders. *Advances in neural information processing systems*, 31, 2018.

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. *arXiv preprint arXiv:1805.01070*, 2018.

Danilo Croce, Daniele Rossini, and Roberto Basili. Auditing deep learning processes through kernel-based explanatory models. 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)*, pp. 4037–4046, 2019.

Fahim Dalvi, Abdul Rafae Khan, Firoj Alam, Nadir Durrani, Jia Xu, and Hassan Sajjad. Discovering latent concepts learned in bert. In *International Conference on Learning Representations*, 2021.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Finale Doshi-Velez and Been Kim. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*, 2017.

Jacob Eisenstein. Uninformative input features and counterfactual invariance: Two perspectives on spurious correlations in natural language. In *NAACL*, 2022.

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.

Amir Feder, Katherine A. Keith, Emaad Manzoor, Reid Pryzant, Dhanya Sridhar, Zach Wood-Doughty, Jacob Eisenstein, Justin Grimmer, Roi Reichart, Margaret E. Roberts, Brandon M. Stewart, Victor Veitch, and Diyi Yang. Causal inference in natural language processing: Estimation, prediction, interpretation and beyond. *CoRR*, abs/2109.00725, 2021. URL https://arxiv.org/abs/2109.00725.

Matt Gardner, William Merrill, Jesse Dodge, Matthew E Peters, Alexis Ross, Sameer Singh, and Noah Smith. Competency problems: On finding and removing artifacts in language data. *arXiv preprint arXiv:2104.08646*, 2021.

Yash Goyal, Amir Feder, Uri Shalit, and Been Kim. Explaining classifiers with causal concept effect (cace). *arXiv preprint arXiv:1907.07165*, 2019.

Michael Harradon, Jeff Druce, and Brian Ruttenberg. Causal learning and explanation of deep neural networks via autoencoded activations. *arXiv preprint arXiv:1802.00541*, 2018.

Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, et al. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *International conference on machine learning*, pp. 2668–2677. PMLR, 2018.

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*, pp. 5338–5348. PMLR, 2020.

J Richard Landis and Gary G Koch. The measurement of observer agreement for categorical data. *biometrics*, pp. 159–174, 1977.

Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. Program induction by rationale generation: Learning to solve and explain algebraic word problems. *arXiv preprint arXiv:1705.04146*, 2017.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P11-1015.

Tom McCoy, Ellie Pavlick, and Tal Linzen. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 3428–3448, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1334. URL https://aclanthology.org/P19-1334.

Matthew O'Shaughnessy, Gregory Canal, Marissa Connor, Christopher Rozell, and Mark Davenport. Generative causal explanations of black-box classifiers. *Advances in Neural Information Processing Systems*, 33:5453–5467, 2020.

Judea Pearl. *Causality: Models, Reasoning and Inference*. Cambridge University Press, 2nd edition, 2009.

Judea Pearl. Direct and indirect effects. In *Probabilistic and Causal Inference: The Works of Judea Pearl*, pp. 373–392. 2022.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67, 2020.

Nazneen Fatema Rajani, Bryan McCann, Caiming Xiong, and Richard Socher. Explain yourself! leveraging language models for commonsense reasoning. *arXiv preprint arXiv:1906.02361*, 2019.

Michael Röder, Andreas Both, and Alexander Hinneburg. Exploring the space of topic coherence measures. In *Proceedings of the eighth ACM international conference on Web search and data mining*, pp. 399–408, 2015.

Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pp. 618–626, 2017.

Joshua Tenenbaum. Bayesian modeling of human concept learning. In M. Kearns, S. Solla, and D. Cohn (eds.), *Advances in Neural Information Processing Systems*, volume 11. MIT Press, 1998. URL https://proceedings.neurips.cc/paper/1998/file/d010396ca8abf6ead8cacc2c2f2f26c7-Paper.pdf.

Lifu Tu, Garima Lalwani, Spandana Gella, and He He. An empirical study on robustness to spurious correlations using pre-trained language models. *Transactions of the Association for Computational Linguistics*, 8:621–633, 2020.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Victor Veitch, Alexander D'Amour, Steve Yadlowsky, and Jacob Eisenstein. Counterfactual invariance to spurious correlations in text classification. *Advances in Neural Information Processing Systems*, 34, 2021.

Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart Shieber. Investigating gender bias in language models using causal mediation analysis. *Advances in Neural Information Processing Systems*, 33:12388–12401, 2020.

Sandra Wachter, Brent Mittelstadt, and Chris Russell. Counterfactual explanations without opening the black box: Automated decisions and the gdpr. *Harv. JL & Tech.*, 31:841, 2017.

Tongshuang Wu, Marco Tulio Ribeiro, Jeffrey Heer, and Daniel S Weld. Polyjuice: Generating counterfactuals for explaining, evaluating, and improving models. *arXiv preprint arXiv:2101.00288*, 2021.

Chih-Kuan Yeh, Been Kim, Sercan 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.

Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *NIPS*, 2015.

Appendix for "Explaining Language Models with Causal Concepts"

A Overall method visualization

Figure 4 shows the overall visualization of the concept generation process. The neural network f is divided at the intermediate layer into two parts: ϕ and ψ. The **black-arrow** path shows the original neural network prediction process: $y = f(\mathbf{x}) = \psi(\phi(\mathbf{x}))$, where $\mathbf{x}$ is the input and y is the classification output.

To generate concepts at the intermediate layer, instead of feeding $\phi(\mathbf{x})$ directly into ψ, we first pass it through a concept network: Firstly, $\phi(\mathbf{x})$ is condensed into concept probabilities $p_{\mathcal{C}}(\mathbf{x})$ by multiplying the normalized activations $\phi(\mathbf{x})$ with normalized concept vectors $\mathcal{C} = \{\mathbf{c}_1, \ldots, \mathbf{c}_n\}$ and going through the threshold (TH) function. Then, a 2-layer perceptron g_θ is used to reconstruct the original activation: $\phi(\mathbf{x}) \approx g_\theta(p_{\mathcal{C}}(\mathbf{x}))$. The reconstruction is then passed into ψ to get the prediction $y' = \psi(g_\theta(p_{\mathcal{C}}(\mathbf{x})))$. To train the network, we use reconstruction loss, regularizer loss, and causality loss.

The **green path** indicates the mapping back process from concept probabilities $p_{\mathcal{C}}(\mathbf{x})$ to input tokens in $\mathbf{x} = [x_1, \ldots, x_t]$. We use the transformer visualization approach (Chefer et al., 2021) and Grad-CAM (Selvaraju et al., 2017), which rely on the gradients generated from the red path.

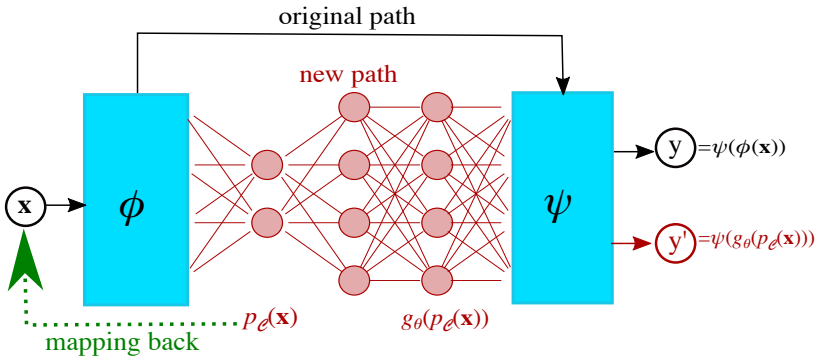

Figure 4: The overall concept generation process.

B Toy example

We conduct experiments on a synthetic (toy) image dataset with ground truth concepts in order to test the validity of our method and confirm the claim that higher confounding effects within the dataset lead to more correlational explanations, thus calling for a more causal explainability approach. Specifically, We extend the toy dataset design of Yeh et al. (2020) to make it more realistic by inserting spurious correlations.

B.1 Data generation

As a synthetic setup, at most 15 shapes are randomly scattered on a blank canvas at random locations with random color selections (as noise). For each image sample x_j, $\boldsymbol{z}^j_{\{1:15\}}$ are binary variables of whether or not a shape is present in x_j with each $\boldsymbol{z}^j_s$ sampling from a Bernoulli distribution with probability 0.5. Then, a 15-class target $\boldsymbol{y_j}$ is constructed with respect to whether the first 5 shapes ($\boldsymbol{z}^j_{\{1:5\}}$) are present or not with human-designed rules. For example, $\boldsymbol{y_1} =\sim (\boldsymbol{z_1} \cdot \boldsymbol{z_3}) + \boldsymbol{z_4}$. A total of $60,000$ examples are generated as the toy dataset using a seed of 0.

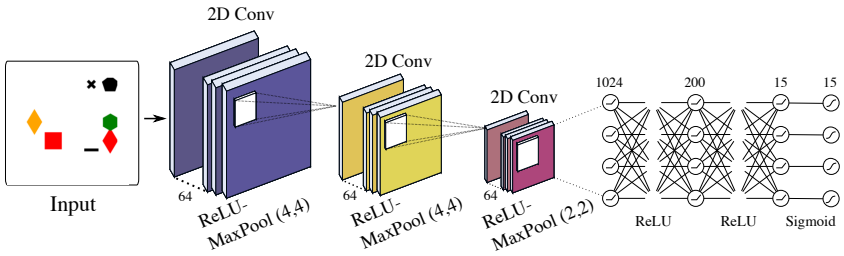

Figure 5: Convolutional Neural Network used for classifying the toy dataset.

The setup mentioned above is, in fact, far away from realistic scenarios, as it doesn't consider possible confounding. Thus, to make it more realistic, we insert spurious correlations between the pairs $(\boldsymbol{z}^j_{\{1:5\}}, \boldsymbol{z}^j_{\{6:10\}}), (\boldsymbol{z}^j_{\{6:10\}}, \boldsymbol{z}^j_{\{11:15\}})$ with a correlation factor p_{cor}. For example, when $z_1 = 1$, $z_6 = \text{Bernoulli}(p_{\text{cor}})$; when $z_1 = 0$, $z_6 = \text{Bernoulli}(1 - p_{\text{cor}})$.

B.2 CNN CLASSIFICATION MODEL USED FOR THE TOY EXAMPLE

The CNN classification model used for the toy dataset is shown in Figure 5. Specifically, 3 convolutional layers with a kernel size of 5 and 64 output channels were used, each followed by a ReLU activation and max pooling layer. Then, the result is flattened into a linear vector, followed by 2 linear layers and a sigmoid activation function. The output is a 15-dimensional binary classification probability. The model is trained for 100 epochs with an Adam optimizer with learning rate $3e-4$. For reproducibility purposes, the model is initialized and trained with a seed of 0.

B.3 VISUALIZATIONS

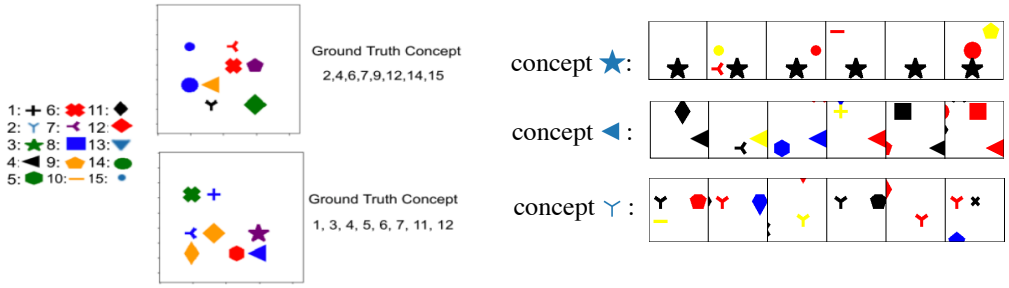

Figure 6: Examples from the toy dataset and concepts discovered.

As an example visualization, in Figure 6, two random images from the toy dataset are displayed on the left, while three example concepts discovered by CausalConcept are plotted on the right. We could observe that CausalConcept is able to derive meaningful clusters as concepts, which provide a sanity check for usability of the latent concepts.

B.4 RESULTS ON TOY DATASET

From the results shown in Table 5, we could observe that, as we increase p_{cor} to mimic an increase in confounding levels in real life, our CausalConcept consistently outperforms the baseline by a bigger margin. CausalConcept achieves higher causal effects (ACE) and higher causal accuracy change (ΔAcc), while maintaining the best RAcc, indicating faithfulness to the original predictions. Moreover, we note that the improvement is even stronger in real data experiments, as the added artificial confounding is more complicated in real-life scenarios.

Table 5: Faithfulness (RAcc) and causality (ACE, ΔAcc) evaluation on the toy dataset. Cls.Acc denotes model's classification accuracy.

p_{cor}	Cls.Acc	Method	RAcc ↑	ACE ↑	ΔAcc ↑
0.50	95.4%	ConceptSHAP	97.6%	0.070	6.1%
		CausalConcept	**98.4%**	**0.102**	**9.4%** (+3.3%)
0.65	99.0%	ConceptSHAP	**99.7%**	0.038	3.5%
		CausalConcept	99.3%	**0.084**	**6.8%** (+3.4%))
0.75	96.1%	ConceptSHAP	98.3%	0.069	6.0%
		CausalConcept	**98.9%**	**0.123**	**12.16%** (+6.16%)

Table 6: A summary of the datasets.

Dataset	Train	Test	Label dim.	Avg. size
Toy (image)	48k	12k	15	(240, 240)
IMDB (text)	37.5k	2.5k	2	215
AG (text)	120k	7.6k	4	43

C HYPERPARAMETERS USED

For all concept experiments, the following parameters are universally applied as a selected default, which demonstrated better performances during experiments: For regularizer losses, $\lambda_1 = 0.1$ and $\lambda_2 = 0.5$. In TH$(\cdot, \beta)$ function, threshold is set to be $\beta = 0.1 = \frac{1}{n}$, where n is the number of concepts selected. For the top-N neighborhood, $N = \frac{1}{4}$BS, where BS is the effective batch size, which we have set as 128 during the experiments. For the masking strategy, we always recommend masking random concepts with a probability of 0.2 as the optimal strategy, as masking maximum concepts may lead to a highly uneven distribution of ACE among discovered concepts.

As all dataset class sizes are small (2 in IMDB/toy or 4 in AG-News), the number of concepts is chosen to be 10 for all experiments. When the number of classes is larger, we recommend choosing a larger number of concepts to ensure a faithful reconstruction of the original input.

For training the concept model, we always use an Adam optimizer with a learning rate of $3e-4$. All models are all trained using 100 epochs. In the CausalConcept models, causal loss is always turned on at half of the overall number of epochs. After turning on causal loss, all parameters are set to untrainable except for the concept vectors, which ensures that the reconstruction ability is not forgotten.

The same hyperparameters are set for the conceptSHAP models, which are also found to generate the optimal performances. The threshold is set to be $\beta = 0.3$, as recommended by the original paper on NLP datasets.

For the causal loss regularizer, $\lambda_c = 1$ is set for all experiments, except for $\lambda_c = 3$ in the case of IMDB with BERT. A higher λ_c will usually lead to a higher output change (ACE and ΔAcc), accompanied by a decrease in faithfulness (RAcc).

To reproduce, all experiments were run with a random seed of 0.

D RUN-TIME

As our model optimizes for causality loss, the run-time is slightly longer than the baseline method ConceptSHAP (Yeh et al., 2020), but is still short. A summary of runtime is shown in table 7. All models shown are run on the GTX 1080Ti graphic card with 12 GB memory. Generally, as post-hoc

Table 7: A summary of runtime (in seconds) on datasets for BERT.

Dataset	β-TCVAE	kmeans	PCA	conceptSHAP	CausalConcept
IMDB	475.9	37.7	0.8	199.3	227.2
AG	1525.6	15.51	2.5	1749.65	2242.1

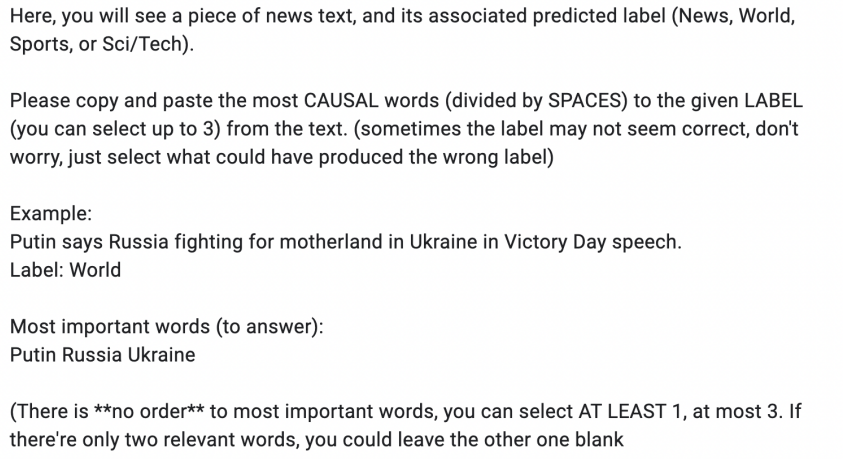

Figure 7: Human study instructions with a demonstration.

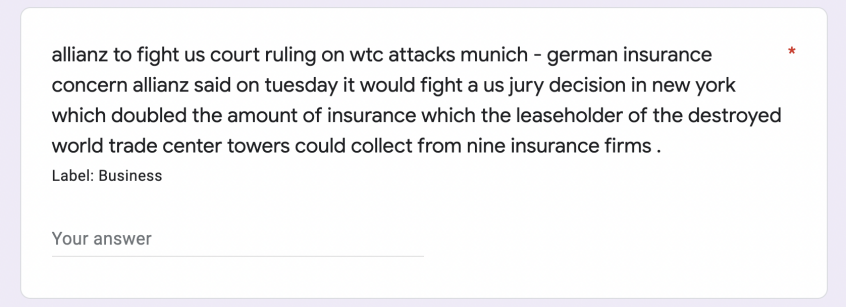

Figure 8: Human study question and answer.

explainability methods, the runtimes are very light and, therefore, a concern that is less important than the model quality. For example, on a dataset of size 50k such as IMDB, it only takes 227.2 seconds (3.8) minutes to train our CausalConcept model.

E HUMAN STUDY SETUP

For the human study, 100 examples are randomly selected from the test set $\mathcal{D}_{\text{test}}$. The questionnaire takes the format of a google form, where the instructions in Figure 7 are shown to the participants. An example question looks like the one in Figure 8. For the 100 questions repeated twice, 4 volunteers (Ph.D. students) have answered them. The volunteers are all proficient in English. The volunteers report an average time of 30 minutes for answering 50 questions. As the volunteers are working also in AI-related areas and are briefed about the purpose and usage of survey data beforehand, they understand fully the data collection and usage. Thus, implicit consent is granted by participation.

F HYPERPARAMETER COMPARISONS

The proposed method of CausalConcept includes many tunable hyperparameters, including the top-N neighborhood, threshold, etc. While these parameters are set at the default mentioned in Appendix C, there are two hyperparameters that users can customize the most: the layer to interpret at and number of concepts . To better understand how these two parameters may affect the generated concepts, we conduct comparisons on both. We evaluate in terms of causal effects and topic quality. For causal effects, we have reported the number of effective concepts left after post-processing, the recovering accuracy (RAcc), the Average Causal Effect (ACE), and the induced change in accuracy (ΔAcc). For topic quality, we have reported coherence scores, including averaged Pointwise Mutual Information

(PMI) (c_uci score), normalized PMI (c_npmi score), c_v score which measures how often the topic words appear together in the corpus, and word2vec similarity (Röder et al., 2015).

The following comparisons are all conducted on the AG-news dataset with BERT, where the other hyperparameters mentioned in Appendix C stay the same.

F.1 LAYER-WISE COMPARISON

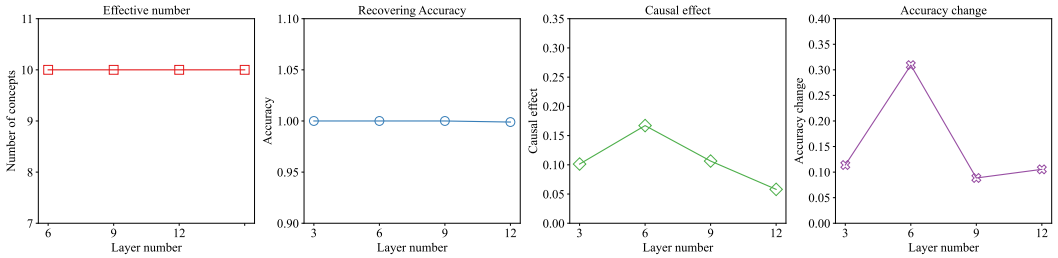

Figure 9: Layer-wise effective number of concepts, RAcc ↑, ACE ↑, and Δ Acc ↑.

To compare what each layer discovered, as BERT has 12 layers, we experimented on the 3rd, 6th, 9th, and penultimate layer respectively, all with 10 concepts.

Quantitatively, we plotted out the effective number of concepts, recovering accuracy, causal effect and accuracy change in Figure 9. All layers demonstrate similar performances in recovering accuracy, which is close to 100%. The intermediate layers, especially the 6th layer, produce a higher causal effect and recovering accuracy. This is because the intermediate layers discover concepts on the token-level, while the penultimate layer concepts are sentence-level (on the [CLS] token). Thus, the token-level concepts will have more fine-grained control.

Figure 10: Wordclouds of concepts generated on the 12th layer, including a sports concept, a technology concept, and a political concept.

Figure 11: Wordclouds of concepts generated on the 9th layer, including a government concept, a China concept, and an Adjective (mostly) concept.

Qualitatively, we plotted some wordclouds of the keywords in discovered concepts in Figure 10 and Figure 11. From the Figure 10, we could see that, in the penultimate layer, concepts are more concentrated on each class. For example, the first concept would correspond to the class “Sports”, the second to “Sci/Tech”, and the third to “World” news. The emphasis on events is also clearer, such as the third one talking about the Iraq War. However, When we move to earlier layers, the concepts’ class labels are more mixed together. In Figure 11, the first concept concerns government, which includes terms such as “government”, “internet”, “security”, “bomb”, “baseball”, etc. It could, however, correspond to many class labels, such as “Sci/Tech”, “World”, or even “Sports”.

Similarity, the second concept talks about China, including “china”, “billion”, “people”, “activitists”, “announcement”, etc. The third concept is interesting as it covers mostly adjective words which do not seem to correlate too much in semantic meanings, such as “low”, “big”, “closer”, and “third”. Similar observations are also confirmed in papers such as (Dalvi et al., 2021), which derives concepts using agglomerative hierarchical clustering combined with human annotations in BERT latent representations. They observe that BERT finds more lexical information in the earlier layers.

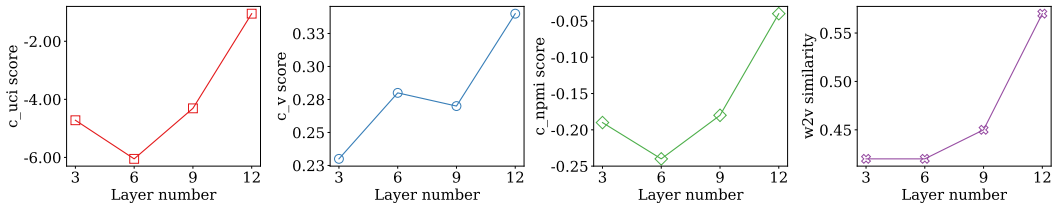

Figure 12: Layer-wise Topic Coherence Comparison.

In terms of topic quality, we evaluated the concept keywords using coherence metrics. As shown in Figure 12, all coherence scores showed a general trend of concepts becoming more coherent as the layer number increases. The conclusion is consistent with the wordcloud visualizations.

Thus, in real-life debugging scenarios, we recommend using the penultimate layer, which will find more coherent topics. However, there could be continued work to discover information learned in the prior layers and to investigate how information flows through layers in a hierarchical way.

F.2 Number of concepts

In the penultimate layer of BERT, we experiment with 3, 5, 10, 50, and 100 concepts.

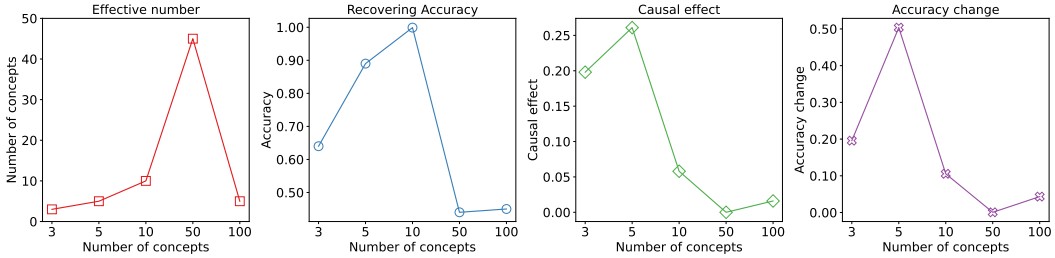

Figure 13: Concept-wise effective number of concepts, RAcc ↑, ACE ↑, and Δ Acc ↑.

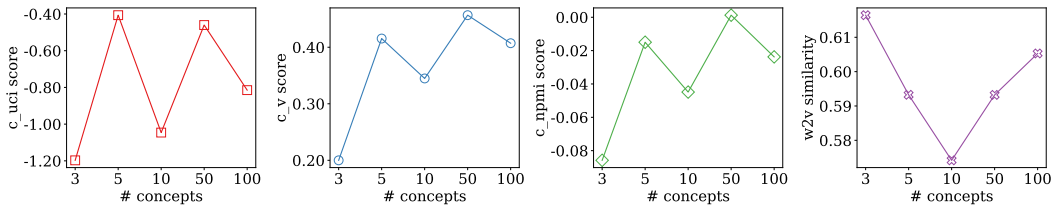

Figure 14: Concept-wise Topic Coherence Comparison.

From Figure 13, we could see that the performance is very dependent on the number of concepts. The effective number of concepts, recovering accuracy, causal effect, and accuracy change all appear to be elbow-shaped. In this case, 5 concepts provided the highest impact on output predictions, as it is close to the number of classes (4) in the AG-News dataset. Increasing the number of concepts to 10 would yield a better recovering accuracy. As the number of concepts increases to 50 and 100, we observe that the model fails to learn completely. In practice, we have often observed the best number to be positively correlated with the number of dataset classes. In other words, a dataset with more classes

will require a higher number of concepts for faithful reconstruction. In terms of topic coherence, we could observe from Figure 14 that the topic coherence scores usually oscillate, but mostly display a generally upward trend of becoming more coherent as the number of concepts increases.

G CLASSIFICATION MODELS USED FOR TEXT EXPERIMENTS

G.1 TRANSFORMER CLASSIFICATION MODEL TRAINED FROM SCRATCH

The self-trained transformer model used during text experiments follows a simple structure: the input text is truncated to max length 512 and passed to an embedding layer of dimension 200. Then, the embeddings are passed through a positional encoding layer with dropout rate 0.2. Then, 6 transformer layers follow with a hidden dimension of 200 and 2 heads. Finally, we mean pool the transformed embeddings and pass through a linear classifier head. The linear outputs are activated with a Sigmoid function to produce class probabilities.

To train the transformer model, we use either the IMDB or AG-News dataset. We train for 10 epochs with a batch size of 128 and an Adam optimizer with learning rate $3e-4$. We also use a learning rate step scheduler with step size 1 and gamma of 0.95.

Table 8: Hyperparameters for finetuning BERT model.

Dataset	AG-News	IMDB
LR	$5e-5$	$3e-4$
train BS	8	8
eval. BS	16	16
seed	42	42
optimizer	Adam betas $= (0.9, 0.999)$ epsilon $= 1e-8$	Adam betas $= (0.9, 0.999)$ epsilon $= 1e-8$
LR scheduler	linear	linear
warmup steps	7425	1546
training steps	74250	15468

G.2 PRETRAINED AND FINETUNED BERT MODEL

For AG-News, we take the finetuned version of bert-base-uncased model on huggingface: "fabriceyhc/bert-base-uncased-ag_news". For IMDB, we finetuned by ourselves on the bert-base-uncased model. The hyperparameters used for both finetuning are reported in 8, where LR stands for learning rate and BS stands for batch size.

The huggingface code and models are all licensed under Apache 2.0, which allows for redistribution and modification. Similarly, the codebase used for replicating the visualization method (Chefer et al., 2021) and the baseline method (Chen et al., 2018) are licensed under the MIT license, which allows for redistribution of the code.