# Grounding Counterfactual Explanation of Image Classifier to Textual Concept Space

Siwon Kim[1] Jinoh Oh[2] Sungjin Lee[2] Seunghak Yu[3] Jaeyoung Do[2] Tara Taghavi[2]
[1]Data Science and Artificial Intelligence Lab at Seoul National University
[2]Amazon Alexa AI, Seattle WA, USA [3]Naver Search US
tuslkkk@snu.ac.kr, {ojino, sungjinl, domjae, taghavit}@amazon.com, seunghak.yu@gmail.com

## Abstract

*Concept-based explanation aims to provide concise and human-understandable explanations of an image classifier. However, existing concept-based explanation methods typically require a significant amount of manually collected concept-annotated images. This is costly and runs the risk of human biases being involved in the explanation. In this paper, we propose Counterfactual explanation with text-driven concepts (CounTEX), where the concepts are defined only from text by leveraging a pre-trained multi-modal joint embedding space without additional concept-annotated datasets. A conceptual counterfactual explanation is generated with text-driven concepts. To utilize the text-driven concepts defined in the joint embedding space to interpret target classifier outcome, we present a novel projection scheme for mapping the two spaces with a simple yet effective implementation. We show that CounTEX generates faithful explanations that provide a semantic understanding of model decision rationale robust to human bias.*

## 1. Introduction

Explainable artificial intelligence (XAI) aims to unveil the reasoning process of a black-box deep neural network. In the vision field, heatmap-style explanation has been extensively studied to interpret image classifiers [20, 21, 24]. However, simply highlighting the pixels that significantly contribute to model outcome does not answer intuitive and actionable questions such as "What aspect of the region is important? Is it color? Or pattern?". On the other hand, drawing human-understandable rationale from the highlighted pixels requires domain expert's intervention and can thus be impacted by the human subjectivity [11].

In contrast, concept-based explanation can provide a more human-understandable and high-level semantic explanation [3, 4, 7, 9, 11]. *Concept* fundamentally indicates an abstract idea, and it is generally equated as a word such as "stripe" or "red". The earliest approach to interpret how a specific concept affects the outcome of the target image classifier is concept activation vector, or CAV [11]. A CAV represents the direction of a concept within the target classifier embedding space and has been widely adopted to subsequent concept-based explanations [15, 19, 25].

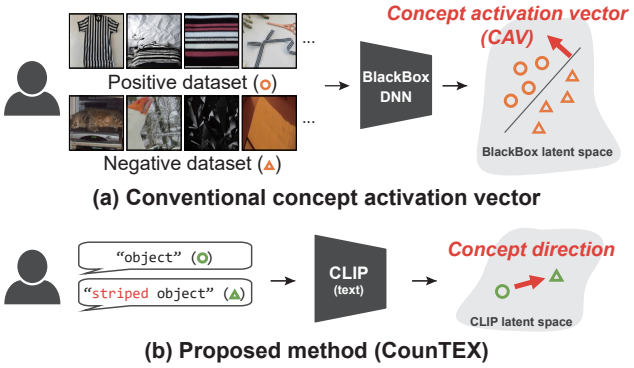


Figure 1. (a) Conventional concept-based explanation derives a CAV with the target model's embedding of manually collected concept-annotated images. (b) CounTEX derives the concept direction directly from texts in CLIP latent space.

However, the CAVs acquisition requires collections of human annotations. The CAV of a concept is typically precomputed via two steps as depicted in Figure 1 (a); 1) collecting a number of positive and negative images that best represent a concept (e.g., images with and without stripes), 2) training a linear classifier (commonly support vector machine) with the images. The vector normal to the decision boundary serves as a CAV. Collecting positive/negative datasets in step 1 is not only costly but also poses the risk of admitting human biases in two aspects; diverging CAVs for the same concept and unintended entanglement of multiple concepts. We will demonstrate in Section 2 that this may threaten credibility of explanation.

Work done during the internship at Amazon Alexa AI

To tackle such challenges, we propose Counterfactual explanation with text-driven concepts (CounTEX), which derives the concept direction only from a text by leveraging the text-image joint embedding space, CLIP [16] (Figure 1 (b)). CounTEX defines a concept direction as the direction between the two CLIP text embeddings of a neutral anchor prompt that does not contain any concept and a target prompt that includes the concept keyword, which is similar to text-guided image manipulation [8, 12].

CounTEX outputs conceptual counterfactual explanation (CE) defined by importance scores of user-specified concepts, similar to the previous conceptual CE method called CCE [1]. Given an input image, prediction of black-box classifier, and a target class, the importance scores answer the question, "How much should each concept be added/subtracted to the image to change the prediction into the target class?". Specifically, an image embedding from the target classifier is perturbed into the weighted sum of the concept directions representing various concepts, where the weights are updated until the prediction becomes the target class. The final weights serve as importance scores of corresponding concepts, indicating the amount of contribution of the concepts to the prediction.

The introduction of CLIP poses a significant challenge; how can we exploit concept directions obtained from CLIP latent space to generate CE for an arbitrary target image classifier? To this end, we propose a novel scheme using projection and inverse projection. The projection maps an image embedding from the intermediate layer of target classifier to the CLIP latent space. The perturbation is conducted in the CLIP space using the text-driven concept directions. The inverse projection brings the perturbed embedding back to the target classifier space so that it can be feed-forwarded to the remaining target classifier. We found that a projector/inverse projector consisting of a simple neural network can effectively map the two latent spaces of target classifier and CLIP and generate faithful explanations.

Another advantage of deriving concept directions from text is that it allows to utilize a wide variety of concepts at a marginal cost. Unlike previous studies that produce explanations with only a small number of concepts, we present faithful explanations consisting of a much larger number of diverse concepts derived for generic classifiers and datasets

Our contributions can be summarized as follows:

1. We propose a novel explanation framework CounTEX to derive the concept direction only from text by leveraging the CLIP joint embedding space.

2. We propose projection/inverse projection scheme to utilize the concept directions defined in the CLIP latent space to explain an arbitrary target image classifier.

3. We show qualitatively and quantitatively improved results that verify CounTEX effectively addresses the limitations of image-driven CAVs.

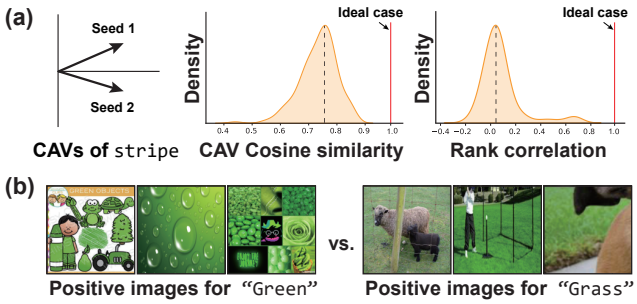


Figure 2. (a) Left: PCA results of misaligned CAVs for the same concept "stripe", Middle: Cosine similarity between two CAVs over various concepts, Right: Rank correlation between concept importance scores generated from different seeds. (b) Positive image examples for concept "green" and "grass"

## 2. Limitations of Image-driven Concept

This section explores two major risks of image-driven CAVs that can threaten the credibility of explanations. Concepts are often abstract words, and it is nontrivial to decide "good" examples that best represent a concept. The decision often depends on the intuition of the annotators and can be very subjective, and there are two risks; 1) Diverging CAVs and 2) unintended entanglement.

**Diverging CAVs (Figure 2 (a))** A CAV may vary upon the positive/negative dataset composition. For each positive and negative dataset of a concept, we randomly selected 30 images using two different random seeds from the concept datasets used in the CCE paper [1] and obtained CAVs respectively from the two compositions. If a CAV is robust to dataset composition, then the two CAVs derived from different seeds should align for the same concept. However, the result depicted in Figure 2 (a) left shows that the CAVs for the concept "stripe" from two seeds are not aligned well, showing a low cosine similarity of 0.73. Note that the result is from principal component analysis. The cosine similarity distribution shown in Figure 2 (a) middle indicates that the misalignment is common for 170 concepts used in CCE.

Unstable explanations are the problematic consequences of diverging CAVs. Figure 2 (a) right shows the distribution of Spearman's rank correlation coefficients between the concept importance scores generated by CCE using the two CAVs sets from different seeds. The rank correlation is very low, which suggests that diverging CAVs lead to very different explanations for the same outcome. This threatens the reliability of the explanation.

**Unintended entanglement (Figure 2 (b))** A CAV of a concept can suffer from an *unintended entanglement* with other concepts. Figure 2 (b) shows positive images for the concept "green" and "grass". The images for the concept

"green" are highly likely to include grass images, and most of the images for the concept "grass" are highly likely to be green colored. This *unintended entanglement* can lead to misleading explanations such as the "grass" concept receiving a high importance score even though the image contains green but no grass at all. We observed that this kind of misbehavior does occur in CCE but not in CounTEX (details are described in Section 4.2).

To overcome the above-mentioned limitations of image-driven CAVs, a significant number of carefully chosen concept-annotated images are required. The limitations are alleviated in CounTEX because it does not depend on image collections. In addition, because it leverages the CLIP latent space pre-trained on extensively crawled large-scale datasets, the risk of unintended entanglement is reduced.

## 3. Method

The key idea of CounTEX is to derive concept direction from text by leveraging the CLIP embedding space. CLIP is trained as a joint embedding space of text and images [16], and it enables us to define concept directions without any positive/negative image examples.

This approach raises the following research questions.

1. Given concept directions textually driven in *CLIP latent space*, how can we exploit them to interpret the outcome of an *arbitrary target classifier*?
2. How can we obtain the text-driven concept directions using CLIP?
3. What other modeling consideration is needed to generate accurate counterfactual explanation (CE)?

Section 3.1 to 3.3 will show how we address question 1. Section 3.4 and 3.5 will demonstrate how we address question 2 and 3, respectively. The overall flow of CounTEX is visualized in Figure 3.

### 3.1. Problem definition: Counterfactual XAI

A typical CE takes three inputs, a black-box classifier $f(\cdot)$, an input image $x$, and a target class $y_t$. Conceptual CE takes an additional input, i.e., a predefined concept library $C = \{c_1, ..., c_N\}$ with $N$ concept keywords. The output is $\boldsymbol{w} \in \mathbb{R}^N$, a vector of concept importance scores. An importance score $w_i$ of a concept $c_i$ indicates the amount by which $c_i$ needs to be added/subtracted to change the prediction to $y_t$. Here, we define a generic *perturbation function* $p(\cdot)$, which takes four inputs; $f$, $x$, $C$, and $\boldsymbol{w}$. The output is a perturbed prediction; $y_p = p(f, x, C, \boldsymbol{w})$.

The goal of conceptual CE is to find the optimal $\boldsymbol{w}^*$ that minimizes the gap between $y_t$ and perturbed prediction $y_p$ and to provide $\boldsymbol{w}^*$ as an explanation. Specifically,

$$\min_{\boldsymbol{w}} \mathcal{L}(p(f, x, C, \boldsymbol{w}), y_t)$$

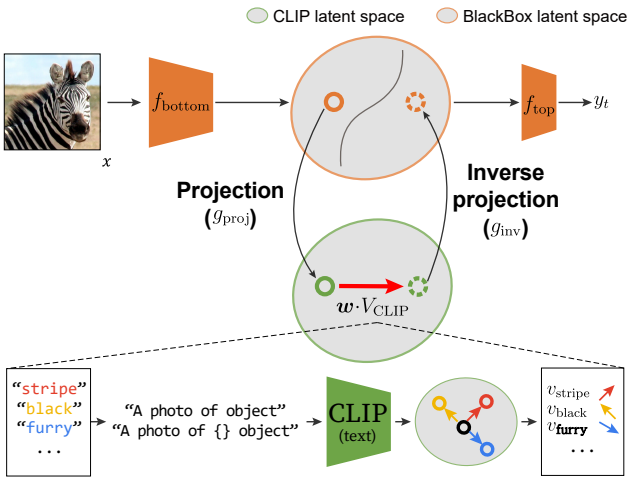


Figure 3. Overall flow of CounTEX.

where $\mathcal{L}$ is commonly defined as cross-entropy loss.

For a misclassified image, $y_t$ can be specified as the ground truth class. In this case, CE shows the root cause of the incorrect prediction. On the other hand, when applied to a correct prediction, CE can help identify features important for making a prediction correct against an arbitrary $y_t$.

### 3.2. Perturbation in CLIP latent space

In conventional conceptual CE methods, including CCE, perturbation function $p(\cdot)$ is defined on embedding linearly perturbed in the embedding space of $f$. Specifically,

$$p(f, x, C, \boldsymbol{w}) = f_{\text{top}}\big(f_{\text{bottom}}(x) + \boldsymbol{w} \cdot V_f\big)$$

where $f_{\text{bottom}}(x)$ is the bottom layers of $f$, $f_{\text{top}}(x)$ is the top linear layer, and $V_f$ is a bank of CAVs of all concepts in $C$. The subscript $f$ denotes that CAVs in $V_f$ are defined with concept image embeddings from $f_{\text{bottom}}$. However, as discussed in Section 2, constructing $V_f$ in target classifier's embedding space requires a significant number of positive/negative examples for each $c$ in $C$.

In CounTEX, the linear perturbation is instead conducted in the CLIP space using the text-driven concept direction bank $V_{\text{CLIP}}$ whose details will be described in Section 3.4. This implies that we need to map the image embedding from the target classifier's latent space to the CLIP's latent space where the perturbation operates. We also need to map the image embedding perturbed in the CLIP space back to the target classifier's embedding space to feed-forward it through the remaining $f_{\text{top}}(\cdot)$. We introduce projection and inverse projection functions, $g_{\text{proj}}$ and $g_{\text{inv}}$, for this purpose.

In summary, our perturbation $p$ is modeled as,

$$p(f, x, C, \boldsymbol{w}) = f_{\text{top}}\Big(g_{\text{inv}}\big(g_{\text{proj}}(f_{\text{bottom}}(x)) + \boldsymbol{w} \cdot V_{\text{CLIP}}\big)\Big)$$

These steps are visualized in the Figure 3. $g_{\text{proj}}$ and $g_{\text{inv}}$ are

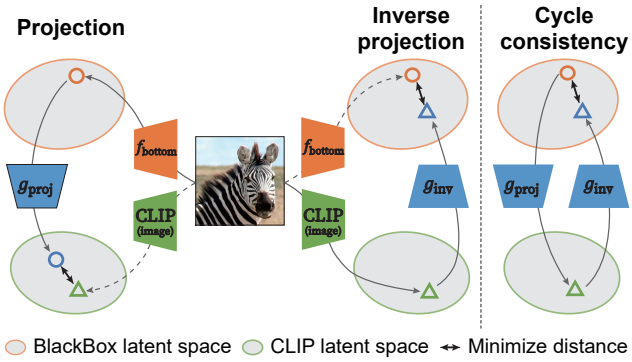


Figure 4. Diagram of projection, inverse projection and cycle consistency.

the functions with learnable parameters that are trained to map the latent spaces of $f$ and CLIP.

### 3.3. Learning Projection and Inverse Projection

We train $g_{\text{proj}}$ and $g_{\text{inv}}$ based on two desiderata which are also illustrated in Figure 4. First, ideally, an image should generate the same embedding regardless of how it is computed. There are two possible paths for computing an image embedding in CLIP latent space; 1) computing an embedding directly from the CLIP image encoder denoted by $\text{CLIP}_{\text{image}}(x)$, 2) projecting the embedding $f_{\text{bottom}}(x)$ into the CLIP space. A projector should minimize the distance between the embeddings computed by the two paths.

The desideratum should hold for the inverse projector as well; 1) Computing an embedding directly from $f_{\text{bottom}}$ and 2) projecting $\text{CLIP}_{\text{image}}(x)$ into that of $f$ via an inverse projection should result in closely located embeddings.

We use the distance between embeddings from the two paths as the loss terms to train $g_{\text{proj}}$ and $g_{\text{inv}}$ as below:

$$\mathcal{L}_{\text{proj}} = ||g_{\text{proj}}(f_{\text{bottom}}(x)) - \text{CLIP}_{\text{image}}(x)||^2$$
$$\mathcal{L}_{\text{inv}} = ||f_{\text{bottom}}(x) - g_{\text{inv}}(\text{CLIP}_{\text{image}}(x))||^2.$$

$g_{\text{proj}}$ and $g_{\text{inv}}$ are separately trained with corresponding loss.

The second desideratum is that projection and inverse projection should not introduce any unnecessary perturbation. Two loss terms defined above do not guarantee whether an image embedding will return to the same embedding after the sequential projection and inverse projection. To reduce the "round trip" error, we introduce an additional cycle consistency loss $\mathcal{L}_{\text{cycle}}$ to fine-tune $g_{\text{proj}}$ and $g_{\text{inv}}$. $\mathcal{L}_{\text{cycle}}$ is defined as the distance as below:

$$\mathcal{L}_{\text{cycle}} = ||g_{\text{inv}}\big(g_{\text{proj}}(f_{\text{bottom}}(x))\big) - f_{\text{bottom}}(x)||^2.$$

After the training of the projector and inverse projector, they are jointly fine-tuned with $\mathcal{L}_{\text{proj}} + \mathcal{L}_{\text{inv}} + \mathcal{L}_{\text{cycle}}$ for a few epochs.

| Category | Prompt template |
|---|---|
| Color | `"A photo of {} object"` |
| Texture | `"A photo of {} object"` |
| Scene | `"A photo of object on {}"` |
| Material | `"A photo of object made of {}"` |
| Part | `"A photo of object containing {}"` |
| Object | `"A photo of object along with {}"` |

Table 1. Prompt templates of $t_{\text{trg}}$ for six concept categories

We empirically found that a simple architecture as MLPs is sufficient for both projection and inverse projection. It is noteworthy that the training can be done with any other dataset different from the training dataset of $f(\cdot)$. The training dataset does not even need any annotation as the training is conducted in an unsupervised manner. The more comprehensive the dataset, the more accurate projection and inverse projection can be expected.

### 3.4. Constructing $V_{\text{CLIP}}$ via Concept Prompting

We construct $V_{\text{CLIP}}$ using only text in the CLIP latent space by prompting concept keywords in $C$. First, for each $c$, we generate a pair of prompts composed of source text $t_{\text{src}}$ and target text $t_{\text{trg}}$. $t_{\text{src}}$ is fixed through all concepts as a concept-neutral generic phrase, "`A photo of object`", following the zero-shot classification prompt strategy from the original CLIP paper [16]. To generate syntactically and semantically correct prompts, the template for $t_{\text{trg}}$ is determined according to the category of $c$. For concept categorization, we leverage $C_{\text{BRODEN}}$ [4]. $C_{\text{BRODEN}}$ is one of the most widely used predefined concept libraries and provides concept categories such as "texture" and "color". Every concept belongs to one of categories, e.g., concept "stripe" belongs to "texture". Templates of $t_{\text{trg}}$ for various concept categories are shown in Table 1.

The generated prompt pair is then used to derive the concept direction. $[t_{\text{src}}, t_{\text{trg}}]$ is tokenized and encoded with CLIP text encoder ($\text{CLIP}_{\text{text}}$), yielding a text embedding pair $[\text{CLIP}_{\text{text}}(t_{\text{src}}), \text{CLIP}_{\text{text}}(t_{\text{trg}})]$. Then the direction $v_c$ of a concept $c$ is computed as the difference between the two text embeddings $v_c = \text{CLIP}_{\text{text}}(t_{\text{trg}}) - \text{CLIP}_{\text{text}}(t_{\text{src}})$. After the computation, it is normalized to a unit vector. By iterating for $\forall c \in C$, we construct the final concept direction bank $V_{\text{CLIP}} = \{v_c \in \mathbb{R}^l | c \in C\}$, where $l$ denotes the dimension of the CLIP text embedding.

### 3.5. Optimizing $w$

We introduce three constraints and corresponding loss terms for optimizing $\boldsymbol{w}$. First, the prediction on the perturbed embedding should change to the target class. Therefore, $\mathcal{L}_{\text{CE}}$ is included to minimize the cross entropy between the predicted label and the target class $y_t$. Second, an identity loss $\mathcal{L}_{\text{id}}$ enforces the minimal perturbation so that the

perturbed embedding does not deviate too much from the original image embedding. Lastly, sparse weights are enforced to ensure concise and human-understandable explanations. The sparseness loss $\mathcal{L}_{\text{reg}}$ imposes L1 and L2 sparseness of $\boldsymbol{w}$ following the conventional approach [1].

Each loss term is formulated as follows:

$$\mathcal{L}_{\text{CE}} = \text{CE}\Big(f_{\text{top}}\big(g_{\text{inv}}(g_{\text{proj}}(f_{\text{bottom}}(x)) + \boldsymbol{w} \cdot V_{\text{CLIP}})\big), \mathbf{y}_t\Big)$$
$$\mathcal{L}_{\text{id}} = \text{MSE}\big(g_{\text{proj}}(f_{\text{bottom}}(x)), g_{\text{proj}}(f_{\text{bottom}}(x)) + \boldsymbol{w} \cdot V_{\text{CLIP}}\big)$$
$$\mathcal{L}_{\text{reg}} = ||\boldsymbol{w}||_1 + ||\boldsymbol{w}||_2$$
$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{CE}} + \alpha \cdot \mathcal{L}_{\text{reg}} + \beta \cdot \mathcal{L}_{\text{id}},$$

where $\mathbf{y}_t$ denotes a one-hot representation of $y_t$. $\mathcal{L}_{\text{total}}$ is the sum of the three loss terms, and the hyperparameters $\alpha$ and $\beta$ are numbers smaller than 1. $\boldsymbol{w}$ is updated to minimize $\mathcal{L}_{\text{total}}$. If the prediction changes to the target class before the loss converges, then the optimization is terminated.

# 4. Experimental Results

## 4.1. Experimental settings

**Black-box Model:** We adopted three image classifiers, CLIP+linear, ResNet18, and ResNet50. CLIP+linear is a linear probe CLIP [16] composed of a linear layer on top of a frozen CLIP image encoder with vision transformer (ViT-B/32) architecture. Unlike ResNet, CLIP+linear does not require projection as it shares the pre-trained CLIP latent space where we define concept directions.

**Datasets:** We used three datasets to train the black-box models. In addition to ImageNet [6], we adopted two datasets, Animals with Attributes2 (AwA2) [23] and CUB-200-2011 (CUB) [22] for the quantitative evaluation. AwA2 has 50 classes with 85 class-wise attributes, and CUB has 200 classes and 312 class-wise attributes.

**Concept library:** We used three pre-defined concept libraries. $C_{\text{BRODEN}}$ is a benchmark concept library proposed from [4]. It has 1,197 general concepts ranging from color to scene. We also utilize attribute names of AwA2 and CUB datasets as concept libraries, $C_{\text{AwA2}}$ and $C_{\text{CUB}}$, especially for quantitative analysis. It is noteworthy that our method is not bounded to specific $C$. CounTEX allows to easily add/remove any concept by simply presenting text, unlike competitors require concept-annotated image datasets.

**Projector and inverse projector**: We empirically found that the projector/inverse projector can be sufficiently implemented with multi-layer perceptron (MLP). Both are comprised of MLP with hidden dimension of (512, 512, 512). Throughout all experiments, we used the projector and inverse projector trained with ImageNet for 50 epochs. The investigation results are shown in Section 4.5.

**Optimization details:** The weight vector $\boldsymbol{w}$ is optimized to minimize $\mathcal{L}_{\text{total}}$. $\alpha$ and $\beta$ were set to 0.1 for all experiments after hyper-parameter search. We used a stochastic gradient descent (SGD) optimizer with a learning rate $10^{-10}$ with the maximum iteration number set to 100. We terminate the optimization early once the predicted class changes to the target class.

## 4.2. Qualitative evaluation

**Identifying features contributing to correct prediction:** We first generate CEs for correctly classified examples. We intentionally select the wrong class as the target class. In this setting, a large negative score of a concept indicates that the concept caused an image to be classified as the correct class rather than the target class. In Figure 5, we show the concepts of the top three and bottom three in terms of importance scores along with the scores.

The results of the proposed method are well aligned with human perceptions ranging from low-level concepts such as color and pattern to high-level concepts including scene and shape. The top-1 and bottom-1 concepts of Figure 5 (a) and (c) show that color is the most discriminative feature that helps the image to be classified to the correct class against the target class. Meanwhile, our method can also identify scenes, such as "ice" or "campsite" in (c), implying that the model relies on class-coherent backgrounds. Moreover, the explanation of CUB-trained model with $C_{\text{CUB}}$ shown in (d) demonstrates that our method can capture various fine-grained concepts from local parts such as "back color" to global shapes such as "duck-like shape".

Results of Figure 5 (a) and (b) are for ResNet-based image classifiers. It shows that CounTEX produces quality results even when the projection and inverse projection are involved. Please refer to Appendix for more examples.

**Debugging misclassification cases:** We apply the proposed method to debug a misclassification case. We 1) identify the required features to correct the prediction using CounTEX, 2) edit the image based on it, and 3) test whether the prediction on the edited image actually changes to the correct class.

We observe that CounTEX helps to correct the misclassification. Figure 6 (a) shows an image of `Hippopotamus` misclassified to `Rhinoceros` by CLIP+linear trained on AwA2 dataset. The CE says that the concepts that need to be added and subtracted the most to correct the prediction to `Hippopotamus` are "water" and "field" respectively. We edited the background to water while preserving the object using the most recent text-guided image manipulation method [17] as shown in (b). (b) shows that the prediction on the background-edited image changes the correct class, `Hippopotamus`.

Like this, CounTEX can help to find the root cause of a model misbehavior by investigating incorrect outcomes with generated CE. The above-mentioned example suggests that the black-box model learned the correlation with background rather than the features directly related to the ob-

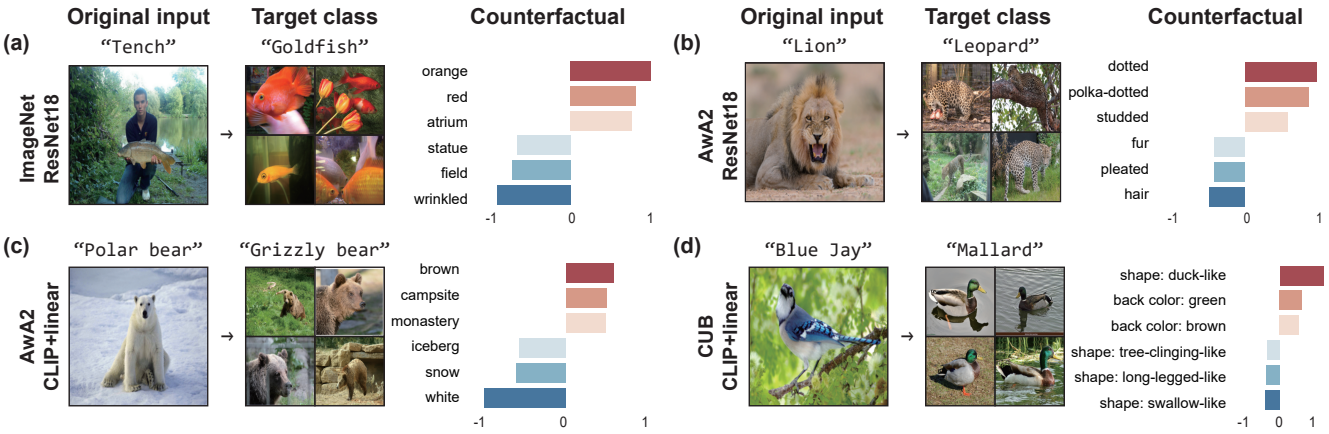


Figure 5. CEs generated by CounTEX. The input image is shown below the original prediction and the target class is shown with representative training images belong to it. The concept importance scores of top-3 and bottom-3 concepts are shown in red and blue colors, which represents positive and negative contributions, respectively. The target models and corresponding training datasets are written to the left to the input images.

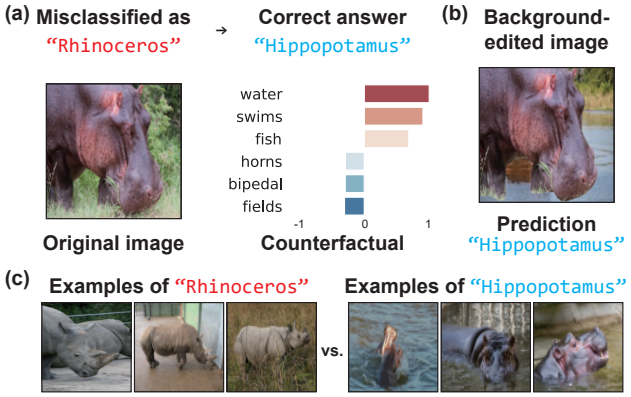


Figure 6. (a) CE generated to turn the prediction on misclassified image into correct answer. (b) Prediction changes to the correct class after top-1 concept-guided image editing. (c) Training image examples of the two classes.

ject. As shown in Figure (c), we found that "field" consistently appears as a background across most training images of Rhinoceros, while "water" frequently appears in training images of Hippopotamus. Based on this finding, we can improve the model by modifying the dataset, e.g., adding more training data of Hippopotamus with diverse backgrounds.

**Qualitative comparison with CCE** [1]: Here, we present an example where the major CAV-based competitor CCE fails to generate faithful explanation while CounTEX succeeds. Especially, Figure 7 shows that CCE assigns a large negative importance score to concept "grass" that does not even exist in the original image. This supports that conventional CAV suffers from unintended entanglement as described in Section 2. On the contrary,

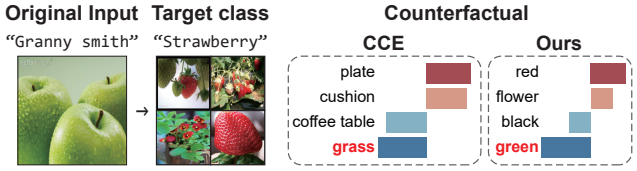


Figure 7. CE generated by CCE and CounTEX for the same prediction. CCE assigns high score to irrelevant concept "grass" unlike CounTEX.

our method successfully reveals that the "green" color contributes the most. The CLIP latent space trained with large-scale datasets makes explanation more robust to unintended entanglement. Note that both results are obtained by using the same concept library that contains "green". Please refer to the Appendix for more examples.

### 4.3. Quantitative evaluation

**Quantitative evaluation protocol:** Quantitative evaluation of conceptual CE is challenging for two reasons: 1) There is a lack of an established dataset that provides a concept-level ground-truth explanation to be compared with the output concept importance score. 2) Even if there is a ground truth, it needs to be adapted to the characteristic of CE that *contrasts* the original prediction against a target class. For this reason, quantitative evaluations of conceptual CEs have been conducted only in very limited settings [1,2].

For more systematic evaluation, we repurpose the class-wise attributes of AwA2 and CUB datasets as concept-level ground truth. They provide a binary attribute vector $\boldsymbol{a}_y$ for each class $y$. Each dimension indicates the presence/absence of the corresponding attribute. We build concept libraries $C_{\text{AwA2}}$ and $C_{\text{CUB}}$ with the attribute keywords.

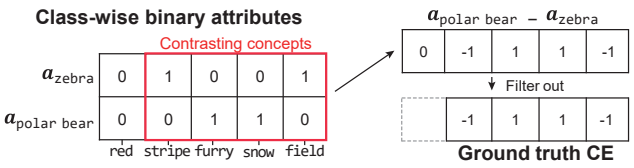


Figure 8. Example of generating ground truth CE for AwA2 dataset. Here, $y_o$ is zebra and $y_t$ is polar bear.

We then generate CE with $C_{\text{AwA2}}$ and $C_{\text{CUB}}$, so that we can compare the output concept importance scores with the given binary attributes.

We define *ground truth CE* by contrasting class-wise attributes of predicted class $y_o$ and target class $y_t$, which is also depicted in Figure 8. An ideal CE reveals only the attributes that distinguish $y_o$ from $y_t$, i.e., the attributes that exclusively appear in one class. Therefore, we first subtract the two attribute vectors, $\boldsymbol{a}_{y_o}$ and $\boldsymbol{a}_{y_t}$. Then, each dimension of the resulting vector will have one of {-1, 0, 1}, where 1 and -1 indicate an attribute that is present only in $y_t$ and $y_o$, respectively. Here, 0 indicates that an attribute is present or absent in both $y_t$ and $y_o$, which is out of the interest of CE. Therefore, we filtered out the dimensions with a value zero from $y_t - y_o$. The same dimensions are filtered out from the concept importance scores as well.

We evaluate the performance of CE using the area under roc curve (AUROC). An ideal CE should be able to rank a concept with ground truth 1 at the top and a concept with ground truth -1 at the bottom. AUROC can measure such ranking, especially when the concept important scores are continuous values while the ground truth CE is binary.

**Competitor settings:** Our primary competitor, CCE [1], did not conduct a quantitative evaluation on general image datasets including AwA2 and CUB. Therefore, we precomputed CAVs for CCE with respect to $C_{\text{AwA2}}$ and $C_{\text{CUB}}$ by collecting corresponding positive and negative concept images from Google image search. For both CCE and CounTEX, we randomly selected plenty images from the validation dataset and generated CEs for the various target classes. The generated explanations were compared with ground truth CE. Details are described in the Appendix.

**Quantitative results:** Table 2 shows the AUROC from explanations of various black-box models and datasets. Our method outperforms CCE in assigning higher importance scores to class-discriminative features. Consistently higher AUROC over various models and datasets shows that the CEs generated with text-driven concepts are more accurate than image-driven concepts of CCE.

## 4.4. Effect of Concept Prompting

CounTEX is robust to concept prompt templates, i.e., two prompts with the same semantics produce consistent CEs, even if the constituent words are different. We measured the Spearman's rank correlation between the explanations generated with three different template configurations including the one in Table 1. The other two configurations are described in Figure 9. We replaced words in templates with different but meaning-preserving words, such as replacing "containing" with "having".

| Target model | Dataset | Library | CCE | Ours |
|---|---|---|---|---|
| CLIP+Linear | AwA2 | $C_{\text{AwA2}}$ | 0.6436 | **0.8132** |
| | CUB | $C_{\text{CUB}}$ | 0.7066 | **0.7891** |
| ResNet18 | AwA2 | $C_{\text{AwA2}}$ | 0.6113 | **0.7314** |
| | CUB | $C_{\text{CUB}}$ | 0.6979 | **0.7750** |
| ResNet50 | AwA2 | $C_{\text{AwA2}}$ | 0.5811 | **0.7316** |
| | CUB | $C_{\text{CUB}}$ | 0.6811 | **0.7336** |

Table 2. AUROC comparison. The higher AUROC indicates the more accurate interpretation.

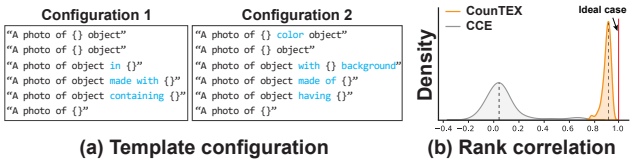


Figure 9. (a) Prompting configurations. (b) Rank correlation distribution between CE generated with different prompts. For the comparison, we also visualized the results of CCE with varying dataset composition in gray color.

The rank correlation close to 1 shown in Figure 9 indicates that the explanations of CounTEX are robust to the differences in prompt templates. Note that we used the same experimental settings as in Section 2. This is a stark contrast to CAVs that significantly diverge depending on the concept dataset collection. Extracting semantics from various text expressions depends on the language understanding performance of the CLIP text encoder. We can expect even more robust explanations if the text encoder further improves.

## 4.5. Evaluation of Projection/Inverse Projection

The mapping capability of projector/inverse projector measured by normalized mean squared error (NMSE) shows two approximators can effectively map the embedding spaces of target classifier and CLIP. NMSE is defined as $\text{NMSE}(z, \hat{z}) = \frac{||z-\hat{z}||^2}{||z||^2}$, where $z$ and $\hat{z}$ denote original and projected/inverse projected embedding, respectively. The lower NMSE indicates the more accurate projection/inverse projection capability.

Given that the distance between $z$ and $\hat{z}$ should be at least smaller than that between two different embeddings belonging to the same class, we use the average intra-class distance as a baseline. It is the average distance between 10 randomly selected image embeddings from the same class.

| NMSE | Projector ($f_{\text{bottom}} \rightarrow$ CLIP) | | Inverse projector (CLIP $\rightarrow f_{\text{bottom}}$) | |
|---|---|---|---|---|
| $f(x)$ | RN18 | RN50 | RN18 | RN50 |
| Intra-class | 0.6810 | | 0.7126 | 0.7724 |
| MLP (10) | 0.3394 | 0.3871 | 0.4370 | 0.5928 |
| MLP (50) | 0.2834 | 0.2606 | 0.3399 | 0.3544 |
| MLP (full) | **0.2479** | **0.2150** | **0.3270** | **0.2314** |

Table 3. Normalized mean squared error (NMSE) of projector and inverse projector. RN abbreviates ResNet. The numbers in parentheses mean the number of training images per class. The lower, the better.

The output of a projector lies in the CLIP latent space, so the intra-class NMSE measured in the CLIP latent space serves as a baseline. Likewise, the baseline NMSE for inverse projector is computed in the target classifier's latent space. In addition, to check if we can enhance the computational efficiency by reducing the number of training data, we trained projector and inverse projector with fewer images. The reduced training datasets are composed of randomly sampled 10 and 50 images from each class.

Table 3 shows the evaluation results. The numbers within parentheses show the number of images per class used for the training. Full indicates using the entire ImageNet training set. All evaluations are conducted with ImageNet validation set disjoint from the training dataset.

There are a few notes worth mentioning: **Note 1:** NMSE of projector and inverse projector are all lower than the baselines. This supports that the projector and inverse projector can effectively map the embeddings in the two latent spaces. **Note 2:** Projector/inverse projector trained with a small number of images, i.e., 50 images per class, shows comparable performance as using the full dataset. Considering that the images do not need annotations, the overhead for training projector/inverse projector can be regarded as marginal.

# 5. Related Works

## 5.1. Concept-based explanation

After CAV was proposed, various literature aimed to reduce the dependency of CAV on concept-annotated dataset [19, 25]. Automatic concept-based explanation (ACE) [9] has been proposed to automate the concept annotation by clustering semantically similar patches from held-out images. It shares some of our motivations to reduce manual concept annotation, but it still derives concept directions from images. Therefore it is vulnerable to the same limitations as CAVs. It also needs another human intervention to extract a concept keyword from clustered patches.

There are ante-hoc concept-based explanation methods that do not explicitly require concept-annotated images [3, 5, 13, 18]. These methods first train an image classifier and then extracts concept keywords that the model learned by manually investigating collections of images that activate certain unit/layer the most. These approaches run another risk of human bias, missing out the concepts that are highly contributing but hard to perceive.

## 5.2. Conceptual CE

Since the CE has gained attention, there have been attempts to generate human-understandable CEs [1, 2, 10, 14]. One of the earliest approaches CoCoX [2] outputs a minimal set of relevant concepts needed to correct the misclassification. The most recently proposed method CCE [1] provides a continuous vector of concept importance scores where each dimension represents the amount of a concept that needs to be added.

However, they all adopt image-driven concept directions, so inherit the limitations of CAV. Although CounTEX adopts the weight optimization scheme for the counterfactual generation process similar to the above-mentioned methods, it does not suffer from the limitations of image-driven concept directions because the directions are driven only from text using CLIP.

# 6. Conclusion

In this paper, we posed the limitations of previous image-driven concept directions and proposed CounTEX, a novel conceptual CE framework based on text-driven concept direction. We leveraged CLIP joint embedding space to derive concept directions via prompting. We introduced projection/inverse projection to utilize concept directions defined in CLIP latent space to explain the target classifier. Also, qualitative and quantitative evaluation results prove CounTEX is able to produce faithful explanation on benchmark image classifiers compared to competitor.

There is still a room for improvement in CounTEX. Especially, a more expressive pre-defined concept library including various fine-grained concepts beyond a single word or short phrase will help to elaborate the explanations. Since the library can easily be augmented for CounTEX by simply presenting textual concept, it would be valuable for future work. CounTEX can also benefit from the improvement in the modeling capacity of a joint embedding space where the concept directions are driven. Especially, if the language understanding of the text encoder improves, the explanation will become more faithful and comprehensive.


# Acknowledgement

This work was partially supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2022R1A3B1077720).

## References


[1] Abubakar Abid, Mert Yuksekgonul, and James Zou. Meaningfully debugging model mistakes using conceptual counterfactual explanations. In *International Conference on Machine Learning*, pages 66–88. PMLR, 2022. 2, 5, 6, 7, 8

[2] Arjun Akula, Shuai Wang, and Song-Chun Zhu. Cocox: Generating conceptual and counterfactual explanations via fault-lines. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 2594–2601, 2020. 6, 8

[3] David Alvarez Melis and Tommi Jaakkola. Towards robust interpretability with self-explaining neural networks. *Advances in neural information processing systems*, 31, 2018. 1, 8

[4] David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. Network dissection: Quantifying interpretability of deep visual representations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6541–6549, 2017. 1, 4, 5

[5] Chaofan Chen, Oscar Li, Daniel Tao, Alina Barnett, Cynthia Rudin, and Jonathan K Su. This looks like that: deep learning for interpretable image recognition. *Advances in neural information processing systems*, 32, 2019. 8

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 5

[7] Ruth Fong and Andrea Vedaldi. Net2vec: Quantifying and explaining how concepts are encoded by filters in deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8730–8738, 2018. 1

[8] Rinon Gal, Or Patashnik, Haggai Maron, Gal Chechik, and Daniel Cohen-Or. Stylegan-nada: Clip-guided domain adaptation of image generators. *arXiv preprint arXiv:2108.00946*, 2021. 2

[9] Amirata Ghorbani, James Wexler, James Y Zou, and Been Kim. Towards automatic concept-based explanations. *Advances in Neural Information Processing Systems*, 32, 2019. 1, 8

[10] Lisa Anne Hendricks, Ronghang Hu, Trevor Darrell, and Zeynep Akata. Generating counterfactual explanations with natural language. In *ICML Workshop on Human Interpretability in Machine Learning*, pages 95–98, 2018. 8

[11] 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*, pages 2668–2677. PMLR, 2018. 1

[12] Gwanghyun Kim, Taesung Kwon, and Jong Chul Ye. Diffusionclip: Text-guided diffusion models for robust image manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2426–2435, 2022. 2

[13] 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*, pages 5338–5348. PMLR, 2020. 8

[14] Oran Lang, Yossi Gandelsman, Michal Yarom, Yoav Wald, Gal Elidan, Avinatan Hassidim, William T Freeman, Phillip Isola, Amir Globerson, Michal Irani, et al. Explaining in style: Training a gan to explain a classifier in stylespace. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 693–702, 2021. 8

[15] Adriano Lucieri, Muhammad Naseer Bajwa, Stephan Alexander Braun, Muhammad Imran Malik, Andreas Dengel, and Sheraz Ahmed. On interpretability of deep learning based skin lesion classifiers using concept activation vectors. In *2020 international joint conference on neural networks (IJCNN)*, pages 1–10. IEEE, 2020. 1

[16] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021. 2, 3, 4, 5

[17] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022. 5

[18] Anirban Sarkar, Deepak Vijaykeerthy, Anindya Sarkar, and Vineeth N Balasubramanian. A framework for learning antehoc explainable models via concepts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10286–10295, 2022. 8

[19] Gesina Schwalbe. Concept embedding analysis: A review. *arXiv preprint arXiv:2203.13909*, 2022. 1, 8

[20] 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*, pages 618–626, 2017. 1

[21] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013. 1

[22] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011. 5

[23] Yongqin Xian, Christoph H Lampert, Bernt Schiele, and Zeynep Akata. Zero-shot learning—a comprehensive evaluation of the good, the bad and the ugly. *IEEE transactions on pattern analysis and machine intelligence*, 41(9):2251–2265, 2018. 5

[24] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014. 1

[25] Ruihan Zhang, Prashan Madumal, Tim Miller, Krista A Ehinger, and Benjamin IP Rubinstein. Invertible concept-based explanations for cnn models with non-negative concept activation vectors. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 11682–11690, 2021. 1, 8