Zero-Shot Textual Explanations via Translating Decision-Critical Features

Toshinori Yamauchi[1] Hiroshi Kera[1,2] Kazuhiko Kawamoto[1]
[1]Chiba University [2]National Institute of Informatics
t.yamauchi@chiba-u.jp, kera@chiba-u.jp, kawa@faculty.chiba-u.jp

Abstract

Textual explanations make image classifier decisions transparent by describing the prediction rationale in natural language. Large vision–language models can generate captions but are designed for general visual understanding, not classifier-specific reasoning. Existing zero-shot explanation methods align global image features with language, producing descriptions of what is visible rather than what drives the prediction. We propose TEXTER, which overcomes this limitation by isolating decision-critical features before alignment. TEXTER identifies the neurons contributing to the prediction and emphasizes the features encoded in those neurons—i.e., the decision-critical features. It then maps these emphasized features into the CLIP feature space to retrieve textual explanations that reflect the model's reasoning. A sparse autoencoder further improves interpretability, particularly for Transformer architectures. Extensive experiments show that TEXTER provides more faithful and interpretable explanations than existing methods. The code is available at https://github.com/tttt-0814/TEXTER.

1. Introduction

Textual explanations describe why an image classifier makes a particular prediction by revealing the visual evidence behind the decision. The explanations help users judge whether the model relies on meaningful patterns and debug the model. Understanding what evidence classifiers use is crucial for building reliable vision systems.

Recent large vision–language models (LVLMs), such as BLIP [26], LLaVA [27], and GPT-4V [1], can produce descriptive captions, but these models are designed for general visual understanding rather than explaining classifier decisions. Supervised concept-based approaches make predictions interpretable through intermediate concepts. Concept bottleneck models (CBMs) [24, 28, 35] predict human-defined concepts before classification, requiring concept annotations and model retraining. Natural language explanation (NLE) models [45, 47] generate textual rationales from annotated explanation–label pairs. Both approaches reflect annotation bias and describe human-labeled concepts rather than revealing the visual evidence that drives predictions.

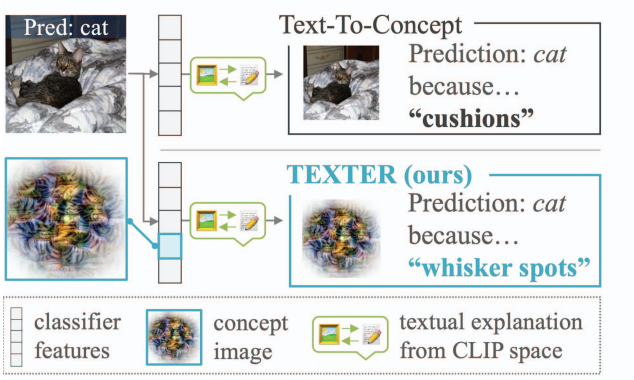

Figure 1. Comparison between Text-To-Concept [32] and the proposed TEXTER for explaining a *cat* prediction. Text-To-Concept, which relies on global image features, produces the explanation "cushions," describing dominant but irrelevant regions. In contrast, TEXTER isolates decision-critical features, such as whisker spots, through a concept image and translates it into the explanation "whisker spots," faithfully reflecting the model's rationale.

Zero-shot alignment methods [32, 46] eliminate this supervision by aligning pretrained classifier features with vision–language models such as CLIP [41]. These methods align entire images with language, describing *what is visible* rather than *what drives the prediction*. For example, as shown in the top of Fig. 1, a representative zero-shot method, Text-To-Concept [32] outputs "cushions" for a *cat* prediction because the method focuses on dominant visual elements instead of decision-critical features. Therefore, a faithful explanation should align language with decision-critical features, not entire images.

To achieve this goal, we propose **TEXTER** (**T**extual **EX**planations via **T**ranslating d**E**cision-c**R**itical features), which isolates the decision-critical features before alignment. TEXTER realizes this isolation by emphasizing the features encoded in the neurons contributing to the prediction. Specifically, TEXTER first identifies the neurons that

contribute to the prediction using Integrated Gradients [52]. The concepts encoded by these neurons are then visualized through feature visualization [11]. A sparse autoencoder (SAE) [15] is then applied to obtain clearer and more interpretable concept representations. The resulting *concept images* emphasize the internal visual evidence that drives the classifier's prediction. As illustrated in the second row of Fig. 1, the concept image for a *cat* prediction highlights the internal visual evidence contributing to the prediction, and translating this concept image into natural language yields the explanation "whisker spots." TEXTER performs this translation by mapping concept images into the CLIP feature space to retrieve textual descriptions that reflect the classifier's reasoning. Focusing on decision-critical features rather than global representations, TEXTER differs from previous zero-shot approaches and yields more faithful textual explanations.

Extensive experiments demonstrate that TEXTER produces faithful and interpretable explanations across both CNN and Transformer architectures. Quantitative analyses show that the concept images capture decision-critical features, while qualitative results confirm that the provided textual explanations reflect the model's actual reasoning.

Our contributions are summarized as follows.

- We propose TEXTER, which translates the decision-critical features of classifiers into the CLIP feature space to provide textual explanations.
- We demonstrate generalizability across both CNN and Transformer architectures by incorporating the SAE to isolate decision-critical features, enabling TEXTER to perform effectively across these model types.
- Extensive experiments show that TEXTER provides explanations that faithfully reflect the rationale behind the predictions. Both quantitative and qualitative evaluations confirm effectiveness.

2. Related work

Providing textual explanations for image classification models has been widely explored, ranging from concept-based reasoning to large vision–language models. Here, we discuss representative approaches, including concept bottleneck models and natural language explanation models, and describe how our study differs from them. Finally, we explain recent zero-shot approaches and their limitations.

Concept bottleneck models (CBMs). Early concept-based interpretability methods, such as Network Dissection [4] and TCAV [23], evaluated how neurons align with human-understandable concepts. Later works introduced Concept Bottleneck Models (CBMs) [24, 35, 49, 56, 59], which predict human-defined concepts before the final label, improving interpretability but requiring concept-labeled data. Recent studies aim to reduce this annotation cost—for instance, Label-Free CBM [35] leverages CLIP for automatic concept labeling, and Hybrid CBM [28] augments predefined concepts with LLM-generated ones. Unlike these studies, our work targets classification models trained solely on images and aims to provide textual explanations without retraining the original model, thereby broadening the applicability of concept-based interpretability to existing pretrained classifiers.

Natural language explanation (NLE) models. NLE models are trained to describe why a vision or vision–language model makes a prediction [7, 18, 39, 47, 50]. They typically combine a task model (e.g., a classifier) with a language model such as GPT-2 [40] to generate explanations, as in NLX-GPT [47], which unifies the two components, and Uni-NLX [45], which extends the framework to multiple vision–language tasks. However, these models depend on paired prediction–explanation annotations and often reproduce annotation bias rather than the model's true reasoning [46]. In contrast, our approach directly generates textual explanations for classification models in a zero-shot setting. Recent large vision–language models (LVLMs), such as BLIP [26], LLaVA [27], and GPT-4V [1], have demonstrated remarkable ability in generating open-ended visual descriptions and reasoning about images. These models, however, are trained for generic vision–language understanding, whereas our work focuses on explaining the decision process of a specific classifier.

Zero-shot textual explanations. Recent work has explored zero-shot methods for generating textual explanations. While several studies leverage vision–language models [14, 31, 44, 51], approaches tailored to image classifiers remain limited. Text-To-Concept [32] addresses this by linearly aligning the classifier feature space with CLIP [41], enabling direct comparison between visual features and text embeddings. Its zero-shot classification accuracy demonstrates the effectiveness of this alignment, eliminating the need for manual concept annotation required by methods such as TCAV [13, 23, 61]. A related approach, ZSNLE [46] trains a lightweight multi-layer perceptron to align the CLIP text-encoder space with the classifier's weight space. Unlike Text-To-Concept, which projects classifier features into the vision–language space, ZSNLE performs the inverse mapping. Another work [3] further develops this CLIP-alignment framework by decomposing representations into component-wise contributions aligned to textual attributes. However, these methods rely on global image features rather than those directly responsible for the model's decision, which motivates our proposed method. Overall, while concept-based and vision–language models have advanced textual interpretability, existing methods either depend on annotated concept labels or fail to isolate

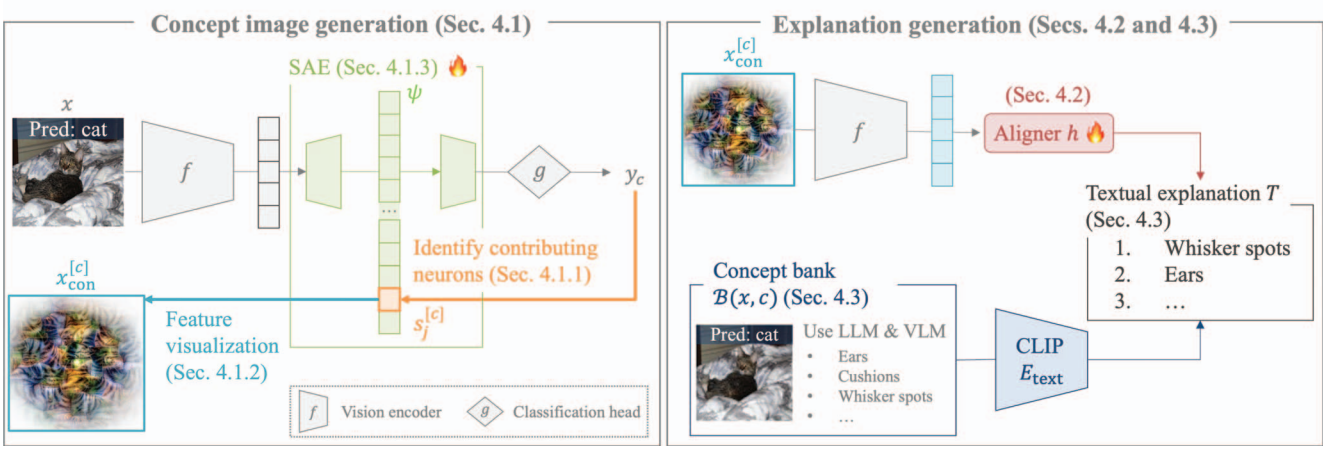

Figure 2. Overview of the proposed TEXTER. The left part illustrates the concept image generation process described in Sec. 4.1, while the right part shows the explanation generation process, in which each component is detailed in Secs. 4.2 and 4.3. Only the SAE modules and the aligner h are trainable, using a subset of the classifier's training dataset. Note that these trainable modules require no additional annotations and are not involved in the inference process of the classifier, and therefore do not affect the original accuracy.

the neurons most responsible for a prediction. Our work addresses this gap by combining neuron-level concept visualization with zero-shot textual grounding.

3. Problem setup

We set up our problem. Let $\mathcal{X}$ denote the set of input images and let the classifier be $\mathcal{F} = g \circ f : \mathcal{X} \to \mathcal{Y}$, where $f : \mathcal{X} \to Z_f$ is a vision encoder mapping an input $x \in \mathcal{X}$ to a feature vector $f(x) \in Z_f$, and $g : Z_f \to \mathcal{Y}$ is a classification head. We take $\mathcal{Y} = \mathbb{R}^C$ to be the logit space for a C-class problem, and write $y \in \mathcal{Y}$ for a logit vector. The classifier is trained solely on image domain, and no language modeling is associated. For such $\mathcal{F}$, we want to explain its decision (i.e., prediction) in natural language.

To this end, we have two challenges. The first is how to associate the latent feature space of the target classifier (i.e., Z_f) with some language domain. We introduce a VLM that consists of an image encoder $E_{\text{img}} : \mathcal{X} \to Z_{E_{\text{img}}}$ and a text encoder $E_{\text{text}} : \mathcal{T} \to Z_{E_{\text{text}}}$, where $\mathcal{T}$ denotes the set of text descriptions. Since the target classifier and the VLM do not share a common feature space, our first goal is to align Z_f with $Z_{E_{\text{img}}}$. Once the alignment is completed, the visual and linguistic features can be associated within the VLM feature space.

The second challenge is to isolate decision-critical information from the feature space of the target classifier. A direct alignment of the global image feature to the VLM space, as in prior work [32, 46], captures only dominant visual attributes rather than explaining the model's decision (see Fig. 1). Therefore, our second goal is to find a projection φ that isolates the decision-critical feature $z^{(c)}$ for the predicted class c from the global image feature $f(x)$.

4. Method

We present TEXTER, a zero-shot framework that explains the rationale behind the prediction in natural language. Figure 2 illustrates the overview of TEXTER. It has three stages, concept image generation, Vision–Language space alignment, and textual explanation generation. Here, we provide an overview of each stage, followed by detailed explanations in the subsequent sections.

Concept image generation (Sec. 4.1). As discussed in Sec. 3, we need to find a projection from the feature vector $f(x)$, which captures the global image feature, to the decision-critical feature $z^{(c)}$ for the predicted class c:

$$f(x) \quad \overset{\varphi}{\longmapsto} \quad z^{(c)}. \tag{1}$$

The concept images generated through feature visualization [11] serve this purpose. Specifically, the concept image $x_{\text{con}}^{(c)}$ associated with x is designed to emphasize the decision-critical feature of x for the predicted class c in the latent space, and we ragard $f(x_{\text{con}}^{(c)})$ as the decision-critical feature $z^{(c)}$. Therefore, the process $f(x) \mapsto x_{\text{con}}^{(c)} \mapsto f(x_{\text{con}}^{(c)})$ can be regarded as a realization of φ.

Vision–Language space alignment (Sec. 4.2). The next step aligns the feature space Z_f with $Z_{E_{\text{img}}}$. This alignment is achieved by training an affine layer h to bring the two feature spaces closer. After alignment, the text space $Z_{E_{\text{text}}}$ becomes associated with the latent space of the target classifier.

Textual explanation generation (Sec. 4.3). Explanations are generated by comparing the aligned classifier features for the concept image $x_{\text{con}}^{(c)}$ with the textual features of the VLM. Since an infinite set of textual descriptions cannot be compared directly, candidates descriptions are retrieved from a concept bank $\mathcal{B}(x, c)$, which contains a set of plausible concept descriptions. The top-k_{con} descriptions with the highest similarity scores are then selected as the textual explanations. Unlike prior works that use global image features $f(x)$ for comparison, TEXTER compares $f(x_{\text{con}}^{(c)})$—the decision-critical features $z^{(c)}$—thereby producing explanations that reflect the model's decision.

4.1. Concept image generation

The mapping φ, which isolates the decision-critical features $z^{(c)}$ for the predicted class c, is realized through the concept image $x_{\text{con}}^{(c)}$. The key insight is that the most influential neurons encode the features underlying the prediction [19, 36], and the concept image is designed to emphasize these decision-critical features. The generation process involves three steps: identifying the contributing neurons, visualizing their concepts, and applying a Sparse Autoencoder (SAE) to improve interpretability.

4.1.1. Identifying the contributing neurons

A contribution score is computed for each neuron, and the top-scoring neurons are selected. Let z be the feature vector at an arbitrary layer of the classifier for an input image x (i.e., $z = f^{(\ell)}(x)$). The contribution score $s_j^{(c)}$ for the $j-$th neuron z_j is computed using integrated gradients [52]:

$$s_j^{(c)} = \left(z_j - z'_j\right) \sum_{m=1}^{M} \frac{\partial F_c\left(z' + \frac{m}{M}(z - z')\right)}{\partial z_j} \cdot \frac{1}{M}, \quad (2)$$

where the baseline z' serves as a reference feature vector from which the integrated gradients accumulate contributions toward the final prediction, and M denotes the number of steps[1]. Here, F_c denotes the function that maps a feature vector to the class-c logit y_c, and it is evaluated at $\hat{z} := z' + \frac{m}{M}(z - z')$, i.e., $y_c = F_c(\hat{z})$. A higher score indicates that the corresponding neuron contributes more to the prediction [52].

After computing the scores for all neurons in z, the neurons are sorted in descending order of $s_j^{(c)}$, and the top-k_{neu} indices are selected:

$$U^{(c)} = \{u_1, \ldots, u_{k_{\text{neu}}}\}. \quad (3)$$

The neurons indexed by $U^{(c)}$ are those contributing most to the class-c prediction.

[1]The baseline z' is set to the zero vector and $M = 100$ by default.

4.1.2. Visualizing concepts in contributing neurons

The concepts encoded by the identified neurons are visualized to obtain the concept image $x_{\text{con}}^{(c)}$. We employ feature visualization [11, 16, 33, 34, 37], a general approach for understanding internal representations of neural networks. Feature visualization optimizes an input $x' \in \mathcal{X}$ to maximize a criterion $\mathcal{L}(x') \in \mathbb{R}$, formulated as:

$$x_{\text{con}}^{(c)} = \underset{x' \in \mathcal{X}}{\operatorname{argmax}}\, \mathcal{L}(x') - \lambda \mathcal{R}(x'), \quad (4)$$

where λ balances the $\mathcal{L}$ and the regularizer $\mathcal{R}$. We adopt magnitude-constrained optimization (MACO) [11], which optimizes the phase spectrum while keeping the magnitude constant in the Fourier space, ensuring that the generated image remains within the natural image distribution. The criterion $\mathcal{L}$ is defined as the sum of the activations of the identified neurons:

$$\mathcal{L}(x') = \sum_{j \in U^{(c)}} [f^{(\ell)}(x')]_j, \quad (5)$$

where $[f^{(\ell)}(x')]_j$ denotes the j-th element of $f^{(\ell)}(x')$. From Eqs. (4) and (5), the concept image $x_{\text{con}}^{(c)}$ represents the features encoded by the neurons contributing to the prediction.

4.1.3. SAE for interpretable concept representations

We employ SAE [5, 12] to obtain more interpretable concept representations. Empirically, we find that the SAE is not essential for CNNs but crucial for Transformers, which recognize objects more compositionally [22, 42, 55], making their entangled feature space less effective for isolating decision-critical factors [54]. As shown in previous studies, SAEs factorize DNN features into sparse, axis-aligned units, enhancing interpretability. See Sec. 5.1 for detailed analysis.

We adopt the TopK SAE [15, 54], which learns an encoder $\Psi(\cdot)$ that maps $f(x)$ to a sparse representation:

$$\Psi(f(x)) = \text{TopK}(W_{\text{enc}}(f(x) - b_{\text{pre}})), \quad (6)$$

where W_{enc} and b_{pre} are trainable weights. The operator $\text{TopK}(\cdot)$ keeps the K largest entries of a vector and sets the others to zero. Training minimizes the reconstruction loss:

$$L_{\text{SAE}} = ||f(x) - W_{\text{dec}}\Psi(f(x))||_2^2, \quad (7)$$

where W_{dec} denotes the decoder weights. Details of the SAE configuration are provided in Appendix B.1.

TEXTER identifies the contributing neurons in its sparse representation and generates concept images for them. Accordingly, by regarding the SAE as an additional module of the classifier, in Eq. (2), $z = \Psi(f(x))$, and $y_c = F_c(\hat{z}) = [g(W_{\text{dec}}\hat{z})]_c$. In Eq. (5), $f^{(\ell)}(x')$ is replaced with $\Psi(f(x'))$.

4.2. Vision-Language space alignment

We use CLIP [41] as the VLM and align the classifier feature space with the CLIP vision feature space. To achieve this, following Text-To-Concept [32], an affine aligner $h(f(x)) = Wf(x) + b$ is trained by minimizing the alignment loss:

$$\min_{W,b} \frac{1}{|\mathcal{D}_{\text{train}}|} \sum_{x \in \mathcal{D}_{\text{train}}} \|Wf(x) + b - E_{\text{img}}(x)\|_2^2, \quad (8)$$

where $\mathcal{D}_{\text{train}}$ denotes the training dataset, and W and b are trainable weights. After training is completed, the aligned features lie in the joint vision–language feature space.

4.3. Textual explanation generation

After training, the aligner h maps the features of the concept image into the CLIP feature space. To obtain textual explanations, similarities between the aligned feature $h(f(x_{\text{con}}^{(c)}))$ for the concept image and the text embeddings $E_{\text{text}}(t)$ of all candidate descriptions $t \in \mathcal{B}(x, c)$ (see the next paragraph) are computed using cosine similarity.

$$\text{sim}(h(f(x_{\text{con}}^{(c)})), E_{\text{text}}(t_i)), \quad t_i \in \mathcal{B}(x, c), \quad (9)$$

where $\text{sim}(\cdot, \cdot)$ denotes cosine similarity. The top-k_{con} descriptions with the highest similarity scores are then selected as the textual explanations, $T = \{t_1, \ldots, t_{k_{\text{con}}}\}$.

Construction of concept bank. The concept bank $\mathcal{B}(x, c)$ is a set of plausible concept descriptions, collected to include those likely to be relevant to the prediction c of the input image x. Given an input image x and its predicted class c, TEXTER constructs concept bank $\mathcal{B}(x, c)$ by leveraging both a large language model (LLM) and a vision-language model (VLM). Inspired by prior studies [31, 35, 59], an LLM is prompted to generate generic descriptions characterizing the predicted class c. Complementarily, a VLM is prompted with the input image x and class c to generate visually grounded descriptions. We use `GPT-3.5-turbo` [38] as the LLM and `Qwen2.5-VL-7B-Instruct` [2, 53, 58] as the VLM. Further details are provided in Appendix B.2.

5. Experiments

We evaluate TEXTER on both CNN- and Transformer-based classifiers to validate that concept images capture decision-critical features rather than dominant image content. We first assess whether concept images preserve the information used in predictions (Sec. 5.1). We then qualitatively and quantitatively compare generated explanations with existing methods in multi-label classification setting (Sec. 5.2) and qualitatively analyze the reasoning patterns TEXTER reveals across architectures (Sec. 5.3). Finally, we demonstrate TEXTER's effectiveness in class-wise explanation scenarios (Sec. 5.4).

Table 1. Evaluation of concept image validity (higher is better). A checkmark (✓) indicates that the concept images are generated from the SAE feature vectors, whereas a dash (–) indicates that they are generated from the feature vectors of the classifier's vision encoder f.

Model	SAE	Acc_1	Acc_5	R_{conf}	Cos
ResNet-18	-	0.92	1.00	1.38	0.72
	✓	0.80	0.96	1.16	0.66
ResNet-50	-	0.92	1.00	1.19	0.74
	✓	0.83	0.97	1.05	0.70
DINO ResNet-50	-	0.79	0.94	1.07	0.63
	✓	0.48	0.80	0.61	0.63
ViT	-	0.11	0.17	0.06	0.11
	✓	0.99	1.00	1.04	0.44
DINO ViT-S/8	-	0.08	0.15	0.06	0.34
	✓	0.89	0.96	0.91	0.61

Models and setup. We use ResNet-18, ResNet-50 [17], and DINO ResNet-50 [6] for CNNs, and ViT [9] and DINO ViT-S/8 [6] for Transformers. The aligner h maps the classifier features into the CLIP feature space using the ViT-B/16 vision encoder. We adopt the same training data and protocol as [32], using 20 % of the ImageNet-1K training dataset [8] for $\mathcal{D}_{\text{train}}$. For aligners with publicly released weights, we use the official checkpoints provided in the authors' GitHub repository.[2] All other training settings for the aligner follow the original configuration in [32]. In Eq. (3), we set $k_{\text{neu}} = 6$. For the concept bank, unless otherwise specified, we obtain 100 descriptions from the LLM and 30 from the VLM, resulting in 130 descriptions in $\mathcal{B}(x, c)$ (i.e., $|\mathcal{B}(x, c)| = 130$). Additional implementation details are provided in Appendix B.3. Unless otherwise specified, the following experiments use the ImageNet-1K dataset.

5.1. Validity assessment of concept images

TEXTER generates explanations from concept images $x_{\text{con}}^{(c)}$, which are produced from the SAE feature vectors, rather than from the original images. Therefore, we assess whether the concept images preserve the features that drove the original predictions of the corresponding classifiers.

Metrics. We use three metrics.

- Accuracy (Acc): Top-1 and Top-5 accuracy for the original predicted label c when classifying concept images.
- Confidence ratio (R_{conf}): Ratio of model confidence (softmax probability) for class c on the concept image to that on the original image.

[2] https://github.com/k1rezaei/Text-to-concept

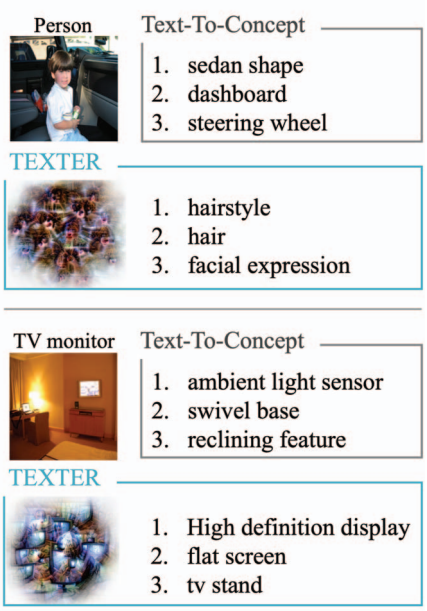

Figure 3. Comparison of provided explanations between Text-To-Concept and the proposed method. The figure presents two cases: the prediction of *person* (top) and the prediction of *TV monitor* (bottom). For each result generated by TEXTER, the corresponding concept image is displayed.

- Cosine similarity (Cos): Cosine similarity between the logit vectors of the original and concept images $\cos(g(f(x)), g(f(x_{\text{con}}^{(c)})))$.

Higher values across all metrics indicate better preservation of decision-critical information.

Results. Table 1 summarizes results on 1,000 randomly selected ImageNet-1K test images (200 classes, 5 images per class). We compare the concept images generated directly from the classifier's vision encoder f (i.e., without applying the SAE).

Without the SAE, concept image validity remains high for CNN models, whereas Transformer models obtain very low scores. This difference is consistent with the compositional and entangled nature of Transformer feature spaces [22, 42, 54, 55], which makes it difficult to isolate decision-critical features from the raw representations. Applying the SAE addresses this issue and leads to large improvements for Transformer models. For CNNs, the sparsity imposed by the SAE can introduce small decreases (e.g., DINO ResNet-50), but the resulting concept images remain largely comparable.

Overall, TEXTER with the SAE generates concept images that reliably capture decision-critical features across both CNN and Transformer architectures.

5.2. Evaluation of textual explanations

Existing methods that rely on global image features often highlight dominant image contents rather than the actual rationale behind predictions. This limitation becomes more apparent in images containing multiple objects. To demonstrate this limitation and the effectiveness of TEXTER, we conduct experiments in a multi-label classification setting.

We use each pre-trained classifier and fine-tune only its classification layer on the PASCAL VOC dataset [10], where each image may contain multiple object instances. The model produces class-wise predictions via sigmoid activation and outputs multiple classes whose predicted probabilities exceed a threshold of 0.3. To accommodate this setting, the concept bank contains the union of concept sets from the predicted classes (130 descriptions per class). In the following, we discuss both the qualitative and quantitative results.

5.2.1. Qualitative results

Figure 3 compares explanations provided by Text-To-Concept and TEXTER. Text-To-Concept describes visually dominant features regardless of predicted classes. For *person* (top), Text-To-Concept focuses on background elements (e.g., sedan shape) rather than person-specific features. For *TV monitor* (bottom), Text-To-Concept highlights furniture features (e.g., light sensor) instead of monitor-specific features. This occurs because Text-To-Concept relies on global image features. TEXTER provides more appropriate explanations by extracting features from concept images. For *person*, TEXTER identifies hairstyle, hair, and facial expression; *TV monitor*, TEXTER identifies high-definition display, flat screen, and tv stand. These results demonstrate that concept images help isolate the decision-critical features that drive model predictions.

5.2.2. Quantitative results

We quantitatively evaluate textual explanations using semantics-based metrics, following [46]. These metrics measure how well textual explanations semantically align with the images. Prior work [46] compares explanations with the original images; however, such evaluations assess only how well the explanations describe the image rather than the reasoning behind the prediction. Therefore, we use concept images as comparison targets, as they highlight decision-critical features rather than the visual content itself (as discussed in Sec. 5.1). We also conduct evaluations on ImageNet-1K and, for completeness, include a comparison with the original images (see Appendix C.2).

Metrics. Given textual explanations T and concept images $x_{\text{con}}^{(c)}$, we evaluate semantic consistency using the following metrics. Further details are provided in Appendix C.1.

- CLIP-Score [20]: Text-image semantic similarity between $x_{\text{con}}^{(c)}$ and $t \in T$ in CLIP embedding space.

Table 2. Quantitative evaluation using semantics-based metrics. Each metric is computed between the explanations provided by each method and the concept images generated by the proposed method. Best results in bold.

Model	Method	CLIP-Score ↑	LPIPS (A) ↓	LPIPS (S) ↓	FS ↑
ResNet-18	Random	0.2065	0.7591	0.7045	0.5730
	Text-To-Concept	0.2086	0.7591	0.7055	0.5703
	TEXTER	**0.2155**	**0.7481**	**0.6938**	**0.5994**
ResNet-50	Random	0.2050	0.7775	0.7041	0.5874
	Text-To-Concept	0.2073	0.7698	0.7017	0.6032
	TEXTER	**0.2271**	**0.7609**	**0.6972**	**0.6378**
DINO ResNet-50	Random	0.2065	0.7579	0.6957	0.5042
	Text-To-Concept	0.2172	0.7485	0.6938	0.5220
	TEXTER	**0.2340**	**0.7385**	**0.6882**	**0.5512**
ViT	Random	0.2127	0.7804	0.6971	0.1329
	Text-To-Concept	0.2187	0.7909	0.7006	0.1458
	TEXTER	**0.2283**	**0.7727**	**0.6952**	**0.1526**
DINO ViT-S/8	Random	0.2182	0.7648	0.6943	0.3686
	Text-To-Concept	0.2224	0.7592	0.6946	0.3933
	TEXTER	**0.2334**	**0.7539**	**0.6937**	**0.4082**

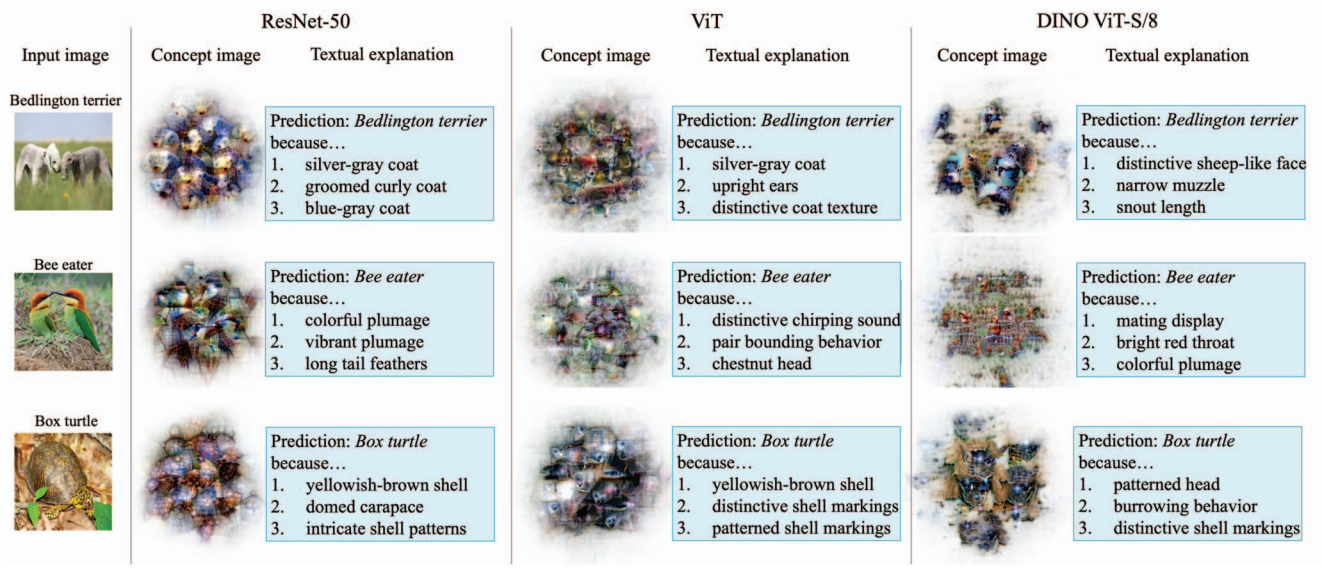

Figure 4. Qualitative comparison of the textual explanations and concept images generated by the proposed method for the same prediction across ResNet-50, ViT, and DINO ViT-S/8. Each row corresponds to one input image and its predicted class.

- LPIPS [60]: Perceptual similarity. We generate the image x_g from T using Stable Diffusion [43] and compute LPIPS between $x_{\text{con}}^{(c)}$ and x_g using pretrained LPIPS models[3] based on AlexNet (A) [25] and SqueezeNet (S) [21].
- Feature Similarity (FS). Cosine similarity between classifier features, $\cos\left(f(x_{\text{con}}^{(c)}), f(x_g)\right)$.

Baselines. We compare TEXTER with the following baselines.

- **Random**, which randomly selects textual explanations from the concept bank.
- **Text-To-Concept** [32], which generates explanations following Eq. (9) but computes similarity using the original image x instead of the concept image $x_{\text{con}}^{(c)}$.

In both Text-To-Concept and TEXTER, the top-3 textual explanations are used ($k_{\text{con}} = 3$), whereas Random selects three at random.

[3]https://github.com/richzhang/PerceptualSimilarity

Results. The results are shown in Tab. 2. All results are computed on 100 randomly selected images from the PASCAL VOC test set, using the top-scoring class for each image. Note that all methods are evaluated using concept images generated by TEXTER. TEXTER obtains higher scores than Random and Text-to-Concept across all metrics and models. These results indicate that TEXTER provides explanations with better semantic alignment with the concept images than the baselines. We additionally evaluate faithfulness via a text-to-region insertion/deletion test; details are provided in Appendix C.3.

5.3. Qualitative comparison across models

Figure 4 compares the textual explanations and concept images generated by TEXTER for the same prediction on the ImageNet dataset across ResNet-50, ViT, and DINO ViT-S/8. The results reveal model-dependent tendencies in which features drive predictions.

For Bedlington terrier (first row), ResNet-50 and ViT focus on texture attributes such as the silver-gray coat, whereas DINO ViT-S/8 attends to structural facial features such as the sheep-like face and narrow muzzle. For Bee eater (second row), ResNet-50 highlights partial visual features such as colorful plumage, whereas ViT and DINO ViT-S/8 retrieve concepts related to interactions between two birds, such as pair bonding behavior and mating display. For Box turtle (third row), ResNet-50 and ViT primarily describe shell appearance, whereas DINO ViT-S/8 also captures behavioral context through concepts like burrowing behavior.

These observations show that Transformer-based models, particularly self-supervised ones like DINO, incorporate relational and contextual information alongside appearance features [22, 42, 55]. Additional results are provided in Appendix D.1.

5.4. Class-wise explanations

Existing zero-shot approaches typically rely on global image–text similarity and therefore do not explicitly support class-conditioned analysis without providing an external class query. In contrast, TEXTER retrieves explanations conditioned on an arbitrary target class, enabling class-wise analysis in multi-class settings. This is particularly important when multiple classes share overlapping visual cues, where background structures may resemble features of competing classes.

Figure 5 illustrates an example where the ground-truth label is *water snake* but the model predicts *stick insect*. Text-To-Concept produces only class-agnostic descriptions, whereas TEXTER generates explanations for both classes. With the target class set to *stick insect*, TEXTER produces explanations such as slender antennae and slender body segments, indicating that twig-like background fragments are misinterpreted as insect limbs. With the target class set to the ground-truth *water snake*, TEXTER produces explanations such as slithering through vegetation and venomous look, suggesting that cues relevant to the ground-truth class remain present despite the incorrect prediction.

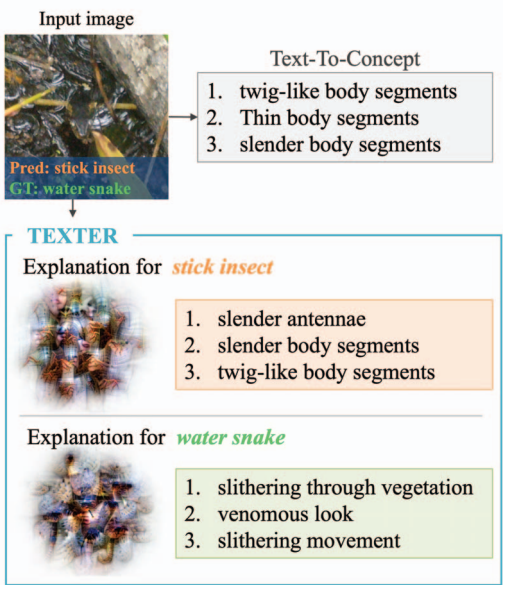

Figure 5. Comparison of the provided explanations between Text-To-Concept and the proposed method for an input whose ground-truth label is *water snake* but is misclassified as *stick insect*. For the proposed method, concept images targeting each class are shown. All explanations are provided from a shared concept bank constructed as the union of those for *stick insect* and *water snake*.

This class-conditioned analysis helps clarify the reasoning behind both the predicted class and the ground-truth class within a single image, leading to a deeper understanding of the model's behavior.

6. Conclusion

We propose TEXTER, a zero-shot framework that explains model predictions in natural language. TEXTER isolates decision-critical features by generating concept images and produces class-specific textual explanations by aligning concept image features with the CLIP feature space. Incorporating a Sparse Autoencoder (SAE) yields more interpretable concept representations and improves explanation faithfulness across both CNN- and Transformer-based models. Experimental results demonstrate that TEXTER captures decision-critical features more faithfully than the existing methods that rely on global image features. This work takes a step toward interpretable vision models that explain what drives model predictions in natural language more effectively.

Acknowledgments

This work was supported by JSPS KAKENHI Grant Number JP23K24914, JST PRESTO Grant Number JPMJPR24K4, JST BOOST Program Grant Number JPMJBY24C6, and ROIS NII Open Collaborative Research 261S07-24168.

References

[1] Josh Achiam et al. Gpt-4 technical report. arXiv: 2303.08774, 2023.

[2] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond. *arXiv preprint arXiv:2308.12966*, 2023.

[3] Sriram Balasubramanian, Samyadeep Basu, and Soheil Feizi. Decomposing and interpreting image representations via text in vits beyond CLIP. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

[4] David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. Network Dissection: Quantifying Interpretability of Deep Visual Representations . In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3319–3327, 2017.

[5] Usha Bhalla, Alex Oesterling, Suraj Srinivas, Flávio P. Calmon, and Himabindu Lakkaraju. Interpreting clip with sparse linear concept embeddings (splice). *CoRR*, abs/2402.10376, 2024.

[6] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2021.

[7] Meghal Dani, Isabel Rio-Torto, Stephan Alaniz, and Zeynep Akata. Devil: Decoding vision features into language. *CoRR*, 2023.

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255, 2009.

[9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *The International Conference on Learning Representations (ICLR)*, 2021.

[10] Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John M. Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *Int. J. Comput. Vis.*, 88(2):303–338, 2010.

[11] Thomas FEL, Thibaut Boissin, Victor Boutin, Agustin Martin Picard, Paul Novello, Julien Colin, Drew Linsley, Tom ROUSSEAU, Remi Cadene, Lore Goetschalckx, Laurent Gardes, and Thomas Serre. Unlocking feature visualization for deep network with MAgnitude constrained optimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

[12] Thomas FEL, Victor Boutin, Louis Béthune, Remi Cadene, Mazda Moayeri, Léo Andéol, Mathieu Chalvidal, and Thomas Serre. A holistic approach to unifying automatic concept extraction and concept importance estimation. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[13] Thomas Fel, Agustin Picard, Louis Bethune, Thibaut Boissin, David Vigouroux, Julien Colin, Rémi Cadène, and Thomas Serre. Craft: Concept recursive activation factorization for explainability, 2023.

[14] Yossi Gandelsman, Alexei A Efros, and Jacob Steinhardt. Interpreting CLIP's image representation via text-based decomposition. In *The Twelfth International Conference on Learning Representations*, 2024.

[15] Leo Gao, Tom Dupre la Tour, Henk Tillman, Gabriel Goh, Rajan Troll, Alec Radford, Ilya Sutskever, Jan Leike, and Jeffrey Wu. Scaling and evaluating sparse autoencoders. In *The International Conference on Learning Representations (ICLR)*, 2025.

[16] Ada Gorgun, Bernt Schiele, and Jonas Fischer. Vital: More understandable feature visualization through distribution alignment and relevant information flow, 2025.

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[18] Lisa Anne Hendricks, Zeynep Akata, Marcus Rohrbach, Jeff Donahue, Bernt Schiele, and Trevor Darrell. Generating visual explanations. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 3–19, 2016.

[19] Evan Hernandez, Sarah Schwettmann, David Bau, Teona Bagashvili, Antonio Torralba, and Jacob Andreas. Natural language descriptions of deep visual features. In *The International Conference on Learning Representations (ICLR)*, 2022.

[20] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. CLIPScore: A reference-free evaluation metric for image captioning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7514–7528, 2021.

[21] Forrest N. Iandola, Song Han, Matthew W. Moskewicz, Khalid Ashraf, William J. Dally, and Kurt Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and ¡0.5mb model size, 2016.

[22] Mingqi Jiang, Saeed Khorram, and Li Fuxin. Comparing the decision-making mechanisms by transformers and cnns via explanation methods. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9546–9555, 2024.

[23] Been Kim, Martin Wattenberg, Justin Gilmer, Carrie J. Cai, James Wexler, Fernanda B. Viégas, and Rory Sayres. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *Proceedings of*

the International Conference on Machine Learning (ICML), pages 2673–2682, 2018.

[24] Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2020.

[25] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2012.

[26] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. BLIP: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *Proceedings of the 39th International Conference on Machine Learning*, pages 12888–12900, 2022.

[27] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *Advances in Neural Information Processing Systems*, pages 34892–34916, 2023.

[28] Yang Liu, Tianwei Zhang, and Shi Gu. Hybrid concept bottleneck models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20179–20189, 2025.

[29] Timo Lüddecke and Alexander Ecker. Image segmentation using text and image prompts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7086–7096, 2022.

[30] Scott M. Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems (NeurIPS)*, page 4768–4777, 2017.

[31] Sachit Menon and Carl Vondrick. Visual classification via description from large language models, 2023.

[32] Mazda Moayeri, Keivan Rezaei, Maziar Sanjabi, and Soheil Feizi. Text-to-concept (and back) via cross-model alignment. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2023.

[33] Anh Nguyen, Alexey Dosovitskiy, Jason Yosinski, Thomas Brox, and Jeff Clune. Synthesizing the preferred inputs for neurons in neural networks via deep generator networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, page 3395–3403, 2016.

[34] Anh Nguyen, Jeff Clune, Y. Bengio, Alexey Dosovitskiy, and Jason Yosinski. Plug & play generative networks: Conditional iterative generation of images in latent space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3510–3520, 2017.

[35] Tuomas Oikarinen, Subhro Das, Lam M. Nguyen, and Tsui-Wei Weng. Label-free concept bottleneck models. In *The International Conference on Learning Representations (ICLR)*, 2023.

[36] Tuomas P. Oikarinen and Tsui-Wei Weng. Clip-dissect: Automatic description of neuron representations in deep vision networks. In *The International Conference on Learning Representations (ICLR)*, 2023.

[37] Christopher Olah, Ludwig Schubert, and Alexander Mordvintsev. Feature visualization. *Distill*, 2017.

[38] OpenAI. Gpt-3.5 turbo models. https://platform.openai.com/docs/models/gpt-3-5, 2025. Accessed 2025-10-13.

[39] Dong Huk Park, Lisa Anne Hendricks, Zeynep Akata, Anna Rohrbach, Bernt Schiele, Trevor Darrell, and Marcus Rohrbach. Multimodal explanations: Justifying decisions and pointing to the evidence. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[40] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI*, 2019.

[41] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 8748–8763, 2021.

[42] Maithra Raghu, Thomas Unterthiner, Simon Kornblith, Chiyuan Zhang, and Alexey Dosovitskiy. Do vision transformers see like convolutional neural networks? In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 12116–12128, 2021.

[43] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10684–10695, 2022.

[44] Leonard Salewski, A. Sophia Koepke, Hendrik P. A. Lensch, and Zeynep Akata. Zero-shot translation of attention patterns in vqa models to natural language. In *Pattern Recognition*, pages 378–393, Cham, 2024.

[45] Fawaz Sammani and Nikos Deligiannis. Uni-nlx: Unifying textual explanations for vision and vision-language tasks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, pages 4634–4639, 2023.

[46] Fawaz Sammani and Nikos Deligiannis. Zero-shot natural language explanations. In *The International Conference on Learning Representations (ICLR)*, 2025.

[47] Fawaz Sammani, Tanmoy Mukherjee, and Nikos Deligiannis. Nlx-gpt: A model for natural language explanations in vision and vision-language tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8322–8332, 2022.

[48] 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 (ICCV)*, pages 618–626, 2017.

[49] Chenming Shang, Shiji Zhou, Hengyuan Zhang, Xinzhe Ni, Yujiu Yang, and Yuwang Wang. Incremental residual concept bottleneck models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11030–11040, 2024.

[50] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2556–2565, 2018.

[51] Aleksandar Shtedritski, C. Rupprecht, and Andrea Vedaldi. What does clip know about a red circle? visual prompt engineering for vlms, 2023.

[52] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *Proceedings of the International Conference on Machine Learning (ICML)*, page 3319–3328, 2017.

[53] Qwen Team. Qwen2.5-vl, 2025.

[54] Harrish Thasarathan, Julian Forsyth, Thomas Fel, Matthew Kowal, and Konstantinos G. Derpanis. Universal sparse autoencoders: Interpretable cross-model concept alignment. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2025.

[55] Shikhar Tuli, Ishita Dasgupta, Erin Grant, and Thomas L. Griffiths. Are convolutional neural networks or transformers more like human vision? *ArXiv*, abs/2105.07197, 2021.

[56] Bowen Wang, Liangzhi Li, Yuta Nakashima, and Hajime Nagahara. Learning bottleneck concepts in image classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10962–10971, 2023.

[57] Haofan Wang, Zifan Wang, Mengnan Du, Fan Yang, Zijian Zhang, Sirui Ding, Piotr Mardziel, and Xia Hu. Score-cam: Score-weighted visual explanations for convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 111–119, 2020.

[58] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024.

[59] Yue Yang, Artemis Panagopoulou, Shenghao Zhou, Daniel Jin, Chris Callison-Burch, and Mark Yatskar. Language in a bottle: Language model guided concept bottlenecks for interpretable image classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19187–19197, 2023.

[60] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 586–595, 2018.

[61] Ruihan Zhang, Prashan Madumal, Tim Miller, Krista A. Ehinger, and Benjamin I. P. Rubinstein. Invertible concept-based explanations for CNN models with non-negative concept activation vectors. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021.