What could go wrong? Discovering and describing failure modes in computer vision

Gabriela Csurka, Tyler L. Hayes, Diane Larlus, and Riccardo Volpi

Naver Labs Europe
`name.lastname@naverlabs.com`
`https://europe.naverlabs.com`

Abstract. In this work, we propose a simple yet effective solution to *predict* and *describe* via natural language potential failure modes of computer vision models. Given a pretrained model and a set of samples, our aim is to find sentences that accurately describe the visual conditions in which the model under-performs. In order to study this important topic and foster future research on it, we formalize the problem of Language-Based Error Explainability (LBEE) and propose a set of metrics to evaluate and compare different methods for this task. We propose solutions that operate in a joint vision-and-language embedding space, and can characterize through language descriptions model failures caused, *e.g.*, by objects unseen during training or adverse visual conditions.

1 Introduction

The sharp contrast between the ideal conditions found in standard benchmarks and the unpredictable nature of the real world majorly hinders the deployment of computer vision (CV) systems in the wild. Despite the most meticulous efforts, samples used to train and validate visual models will only represent a fraction of the diversity that these models will face once deployed. It is thus critical to detect model vulnerabilities and bring them to the user in an interpretable way [5].

Different works have addressed the problem of Language-Based Error Explainability (LBEE) before [8, 16, 28, 41, 43]. Yet, none of them proposed ways to quantitatively assess the predicted descriptions, confining them mainly to a qualitative inspection of the model's error modes. We assert that the LBEE problem requires appropriate metrics, in order to rank methods and track progress in this field. In this paper, we take different steps in this direction through the following contributions: *i)* we provide a rigorous formalization of the LBEE problem (Sec. 2), *ii)* we propose a family of methods to tackle it (Sec. 3), and *iii)* we introduce different metrics to evaluate performance (Sec. 4). We show the effectiveness of our proposed solutions through extensive experiments, focusing on both classification and segmentation segmentation tasks.

Related work. There is a growing body of work connecting explainable AI and CV. Several focus on producing visual explanations, visualizing image regions that contribute the most to model failures [29, 33, 36, 45]. A few works aim at generating counterfactual explanations [10, 13, 15], identifying small image edits that cause the model to fail. Recently, diffusion models have been used to this end [2, 18, 38]. In contrast to these methods which only explain failures on a particular image, our aim is to provide explanations of the model's overall behavior on an entire image set, in order to uncover consistent error patterns and major weaknesses under particular conditions. Furthermore, we rely on natural language as a more interpretable interface.

Explaining errors made by CV models via natural language descriptions is a recent research line. Early methods proposing human-in-the-loop approaches [4, 9, 17, 34, 42] have been followed by solutions that automatically detect consistent failure modes [8, 16, 25, 28, 41, 43]. In particular, [8, 43] propose to learn a mixture model to partition the data according to classification errors related to spurious correlations. [16] characterizes model failures as directions in a latent space found with SVMs. [20] identifies biases by analyzing captions from misclassified images with a Vision-Language Model (VLM) and ranks the keywords from the captions with CLIP [27]. [41] uses conditional text-to-image diffusion models to generate synthetic images that are grouped based on how the model misclassifies them. [28] extracts human-understandable concepts (tags) with VLMs and examine the model's behavior conditioning on the presence or absence of the combination of those tags. Also [25] leverages text-to-image models: encoding specific information in the prompt, they synthesize images from the combinatorially large set of data sub-populations, and run tests to find out on which ones the model under-performs. In contrast with the above works, we provide a rigorous formalization of the LBEE problem, and propose solutions that can be seamlessly applied to arbitrary image task—porting this line of work to semantic segmentation for the first time.

Finally, a recent work proposes methods to describe differences between two image sets [7]. While this method could be applied to LBEE, the proposed evaluation based on GPT-4 can only assess whether the difference between the sets is correct but not if this difference is related or not to model failure.

2 Language-Based Error Explainability

We can define the LBEE problem as automatically finding and describing potential errors from a model based on its performance on a given dataset. Formally, let $\mathcal{M}_\theta$ be a computer vision model parameterized by θ, for an arbitrary computer vision task, such as classification or segmentation. Let $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^M$ be a set of images sampled from an arbitrary target distribution, and let $\omega_\theta^{\text{avg}}$ be the average performance of the model on such data. Finally, let $\mathcal{S} = \{s_n\}_{n=1}^N$ be a broad and generic set of sentences that describe various visual elements and conditions, which can be either manually defined or generated with a large language model (LLM)—see Appendix C.1 in the Supplementary.

Problem formulation. Given a target set $\mathcal{X}$, our goal is to find a set of sentences $\mathcal{R}_\mathcal{S} \subset \mathcal{S}$ describing any likely failure causes for the model $\mathcal{M}_\theta$ with respect to images from $\mathcal{X}$—such as unseen objects, visual conditions never encountered during training, or errors of any other nature. Intuitively, each selected sentence in $\mathcal{R}_{\mathcal{S}}$ should describe elements present in images on which the model fails but not in those in which the model succeeds.

Letting $\omega_\theta^{s_n}$ be the average performance of the model on images for which the sentence s_n is relevant, we determine that such sentence is *describing a failure mode* if the difference between the global average $\omega_\theta^{\text{avg}}$ (average over all images in $\mathcal{X}$) and $\omega_\theta^{s_n}$ is larger than a threshold β, with β being a predefined margin. The desired output $\mathcal{S}_\beta^*$—used to evaluate the performance of the methods— is the collection of these sentences

$$\mathcal{S}_\beta^* = \{s_n \in \mathcal{S} |\, \omega_\theta^{s_n} < \omega_\theta^{\text{avg}} - \beta\} \ . \tag{1}$$

Note that β allows the user to set an arbitrary severity level for the error descriptions, *i.e.*, a higher β restricts the desired set $\mathcal{S}_\beta^*$ to more challenging sentences. We provide more details in Appendix C.2 of the Supplementary.

Relation with prior art. In [8, 43] the language descriptions, while part of the proposed methodology, are not part of the problem formulation and are not quantitatively evaluated. Instead, we treat the language-based description of errors made by computer vision models as the problem itself, providing means to evaluate them. Furthermore, those methods are tailored to the classification task and assume access to ground truth or prior knowledge about the error types to look for (*e.g.*, bias-conflicting sub-populations or context). In contrast, our solution is task-agnostic and we do not use any knowledge about errors prior to evaluation, neither assume—in general—access to class information.

3 A family of approaches to solve LBEE

We propose a family of simple and efficient approaches to tackle the LBEE problem. In contrast with prior art [8,16,43], designed for classification and often requiring access to ground-truth or privileged information, our proposed methods are unsupervised and task-agnostic. Our solutions rely on two key elements: *i)* a joint vision-and-language space $\mathcal{F}$, for which we use Open-CLIP [14], and *ii)* the sentence set $\mathcal{S}$ introduced in the previous section, assumed to be large enough to describe images from the target set and to contain potentially relevant reasons for model failure (see Appendix C.1 in the Supplementary.).

We recall that the problem at hand is the following: Given a set of images, denoted by $\mathcal{X}$, a task-specific pretrained model $\mathcal{M}_\theta$, and a large sentence set $\mathcal{S}$, we want to select the subset from $\mathcal{S}$ that describes the samples from $\mathcal{X}$ on which the model $\mathcal{M}_\theta$ underperforms. Our proposed solution is illustrated in Fig. 1 and follows the steps detailed below.

Step 1: Splitting the target set into easy and hard subsets. Given the model $\mathcal{M}_\theta$ and a confidence measure φ_θ, we define two thresholds t_φ^h and t_φ^e and split the target set $\mathcal{X}$ into three sets: an easy one $\mathcal{X}^e = \{\mathbf{x}_i \in \mathcal{X} | \varphi_\theta(\mathbf{x}_i) > t_\varphi^e\}$, a hard one $\mathcal{X}^h = \{\mathbf{x}_i \in \mathcal{X} | \varphi_\theta(\mathbf{x}_i) < t_\varphi^h\}$, and, if $t_\varphi^e > t_\varphi^h$, a neutral set with the other images. If ground-truth for the target set is available, we can use model performance to split the data; otherwise, any unsupervised confidence measure.

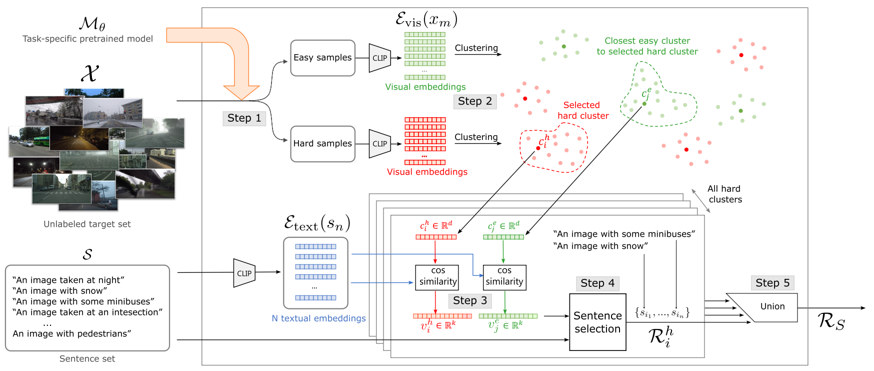

Fig. 1: Overview. Provided with a pretrained model $\mathcal{M}_\theta$, a target image set $\mathcal{X}$, and a set of sentences $\mathcal{S}$, the steps of our LBEE pipeline are the following. Step 1: we split the images from $\mathcal{X}$ into easy and hard sets based on the model's confidence. Step 2: we embed images in the CLIP space and cluster the hard and easy sets independently. Step 3: we assign to each hard prototype the closest easy prototype in this space. Step 4: we embed the sentences in the CLIP space and compute their cosine similarities with the cluster prototypes; based on these sentence-prototype similarities, we select sentences for the hard prototypes (several alternatives are explored in Sec. 3). Step 5: we aggregate cluster-specific sentence sets to produce the global output.

Step 2: Clustering easy and hard subsets. Let us denote by $\mathcal{E}_{\text{vis}}$ and $\mathcal{E}_{\text{txt}}$ the visual and text encoders mapping images and sentences into the joint space $\boldsymbol{\mathcal{F}}$, respectively. Let $\Phi = \{\mathcal{E}_{\text{vis}}(\mathbf{x}_m)\}_{m=1}^M$ be the set of visual representations of images in $\mathcal{X}$ and let Φ^h and Φ^e be the set of representations associated with hard and easy samples, respectively. We cluster the samples in Φ^h and Φ^e independently, using an arbitrary clustering method[1] and represent each cluster by its centroid, referred also as its *prototype*. We compute it as the average of the embeddings of the samples in the cluster. Let $\mathcal{C}^h = \{\mathbf{c}_i^h\}_{i=1}^{|\mathcal{C}^h|}$ and $\mathcal{C}^e = \{\mathbf{c}_j^e\}_{j=1}^{|\mathcal{C}^e|}$ be the set of prototypes associated with hard and easy samples, respectively. In the remainder of the paper, we will use $\mathbf{c}_i^h$ and $\mathbf{c}_j^e$ to indistinctly refer to the clusters or to their prototypes.

Step 3. Matching sentences with prototypes. Let us denote by $\mathbf{s}_n = \mathcal{E}_{\text{text}}(s_n)$ the textual embeddings of the sentences s_n. These embeddings lie in the same joint embedding space as the images and, in turn, as the prototypes $\mathbf{c}_j^e$ and $\mathbf{c}_i^h$. Therefore, for each prototype $\mathbf{c}$ (easy or hard) we can compute the cosine similarity (*i.e.*, the dot product between L2-normalized representations) to each textual embedding $\mathbf{s}_n$. We can store these similarities into a vector $\mathbf{v}_i^{h/e} \in \mathbb{R}^N$, where the nth element of the vector is $< \mathbf{c}_i^{h/e}, \mathbf{s}_n >$. This allows us to characterize each cluster in terms of its semantic similarity with the sentences in $\mathcal{S}$.

[1] We use Agglomerative Clustering with Ward-linkage [40] in our experiments.

Step 4: Retrieving sentences for hard prototypes. Here, we look for a set of peculiar sentences for each hard cluster. To select relevant sentences for a given cluster (easy or hard), we simply rank the sentences by their similarity to the cluster center (by ranking the elements of the vectors $\mathbf{v}_j^e$ and $\mathbf{v}_i^h$). Then we either retain the top ranked sentences or all sentences with a corresponding value in $\mathbf{v}^{h/e}$ above a predefined threshold τ. We denote the set of retained sentences for $\mathbf{c}_i^h$ and $\mathbf{c}_j^e$ by $\mathcal{S}_i^h$ and $\mathcal{S}_j^e$, respectively. While $\mathcal{S}_i^h$ effectively describes images within $\mathbf{c}_i^h$, not all sentences will point to failure reasons. We need to find sentences describing the visual features that make the cluster *hard.* We propose therefore to contrast the hard cluster with its closest easy cluster: its proximity in $\boldsymbol{\mathcal{F}}$ implies content similarity, hence, contrasting allows isolating the attributes that characterize the hard cluster specifically, and not the easy one.

Let $\mathbf{c}_j^e \in \mathcal{C}^e$ be the closest easy prototype to $\mathbf{c}_i^h \in \mathcal{C}^h$ based on their cosine similarity in $\boldsymbol{\mathcal{F}}$. In the following, we propose three different methods to isolate sentences that are peculiar to the hard cluster $\mathbf{c}_i^h$. For a fair and more straightforward comparison between different methods, we fix the number of sentences retained for each cluster and for each method to a predefined K, *i.e.* $|\mathcal{R}_i^h| = K$, where $\mathcal{R}_i^h$ is the sentence set retained by a method for the hard cluster $\mathbf{c}_i^h$.

SetDiff: Sentence set differences. Given the pair of clusters ($\mathbf{c}_i^h$,$\mathbf{c}_j^e$), we first select relevant sentences for both—$\mathcal{S}_i^h$ and $\mathcal{S}_j^e$—and then remove from $\mathcal{S}_i^h$ the sentences that are present in $\mathcal{S}_j^e$, yielding $\mathcal{R}_i^h = \mathcal{S}_i^h \setminus \mathcal{S}_j^e$. Since this approach does not guarantee that $|\mathcal{R}_i^h| = K$, we fill $\mathcal{S}_j^e$ with sentences corresponding to values in $\mathbf{v}_j^e$ above a threshold[2] τ and build $\mathcal{R}_i^h$ with the K most similar sentences to $\mathbf{c}_i^h$ that are not in $\mathcal{S}_j^e$.

PDiff: Prototype difference. Assuming that higher scores in the element-wise difference between the similarity vectors $\mathbf{d}_i^h = \mathbf{v}_i^h - \mathbf{v}_j^e$ better describe the hard clusters than the easy ones,[3] we can rank $\mathbf{d}_i^h$ and retain the sentences corresponding to the top K values. We denote this set by $\mathcal{S}_i^d$ and define $\mathcal{R}_i^h = \mathcal{S}_i^d$.

FPDiff: Filtered PDiff. A drawback of **PDiff** is that it does not guarantee that the selected sentences accurately describe the hard cluster. A remedy to this is to only retain sentences that have high similarity to the prototype $\mathbf{c}_i^h$, namely sentences in $\mathcal{S}_i^d$ that are also in $\mathcal{S}_i^h$. We define $\mathcal{R}_i^h = \mathcal{S}_i^h \cap \mathcal{S}_i^d$. Also here, to guarantee that $|\mathcal{R}_i^h| = K$, we fill $\mathcal{S}_i^h$ with sentences corresponding to values in $\mathbf{v}_i^h$ above a threshold τ and build $\mathcal{R}_i^h$ with the top K sentences having the highest values in $\mathbf{d}_i^h$ that are also in $\mathcal{S}_i^h$.

TopS. We also add a baseline called **TopS** , describing the hard clusters without any contrasting, by selecting the top K ranked sentences from $\mathcal{S}_i^h$ ($\mathcal{R}_i^h = \mathcal{S}_i^h$).

Step 5: Producing the final output. The final set of sentences that describe potential reasons for failure for the whole dataset is the union of all the sentence

[2] We set $\tau = 0.25$ based on the observation that the average similarity between our sentence sets and image sets is around 0.17 due to the well-known modality gap [22].

[3] This, up to some normalization, is equivalent to first consider the prototype differences and then compute the similarity between each sentence s_n and $\mathbf{d}_i^h$.

sets retained for all the hard clusters, namely

$$\mathcal{R}_{\mathcal{S}} = \bigcup_{i=1}^{|C^h|} \mathcal{R}_i^h \quad . \tag{2}$$

We conclude by positioning these methods with respect to prior works.

Relations with prior art. Previous work related to ours [8, 43] addresses a problem that does not perfectly align with ours. Yet, the manner they propose to assign explanations to predicted error modes can be related to some of the solutions proposed above. DOMINO [8], which associates sentences based on average CLIP embedding from which the average class representative is extracted, is closely related to **PDiff** : when we analyze the errors per class (see Sec. 5), this version of our method can be interpreted as a variant of DOMINO in which the class prototype is replaced with the closest easy prototype. FACTS [43] relies on a captioning tool [26] to assign a single tag to each partition without contrasting, hence, it is mostly related to **TopS** .

4 Evaluation metrics for LBEE

Our third contribution is a set of metrics to evaluate predicted language explanations by LBEE methods, which can retrieve error-related sentences without any supervision for arbitrary visual tasks. Given the output of a method, namely the set of sentences $\mathcal{R}_{\mathcal{S}} \subset \mathcal{S}$, we need to assess *i) whether these sentences actually point to reasons for model failure*, *ii) how well the retrieved sentences characterize the image set assigned to the cluster*, and *iii) how well the predicted sentence set $\mathcal{R}_{\mathcal{S}}$ covers the set of potential explanations given by $\mathcal{S}^*_\beta$*. We propose metrics that allow measuring a method's performance in these dimensions.

i) Average Hardness Ratio (HR). Our main goal is retrieving sentences s_n that *point to reasons for model failure*, *i.e.* $s_n \in \mathcal{S}^*_\beta$. This means that the hardness score of the sentence $\omega_\theta^{s_n}$—formally defined in Appendix C.2 of the Supplementary — satisfies the condition $\omega_\theta^{s_n} < \omega_\theta^{avg} - \beta$. The lower this value is, the more the sentence is correlated with model failure. We define the Hardness Ratio (HR) for a given cluster as the ratio of sentences among the retrieved ones in $\mathcal{R}_i^h$ (see Sec. 3) for which the above condition holds. The AHR is the average across all clusters:

$$\text{HR}_i = \frac{|\mathcal{R}_i^h \cap \mathcal{S}^*_\beta|}{|\mathcal{R}_i^h|} \quad \text{and} \quad \text{AHR} = \frac{1}{|C^h|}\sum_{i=1}^{|C^h|} \text{HR}_i \tag{3}$$

***ii)* Average Coverage Ratio (ACR).** Next, we introduce a metric to assess the *coverage* of the sentences associated with the hard clusters, *i.e.* the ratio of images in the cluster for which the retrieved sentence holds. Intuitively this shows how well a sentence $s_n \in \mathcal{R}_i^h$ characterizes a hard cluster $\mathbf{c}_i^h$. CR$_i$ is the mean of these ratios taken over the sentences retained in $\mathcal{R}_i^h$. Formally,

$$\text{CR}_i = \frac{1}{|\mathcal{R}_i^h|} \sum_{s_n \in \mathcal{R}_i^h} \frac{1}{|\mathbf{c}_i^h|} \sum_{\mathbf{x}_m \in \mathbf{c}_i^h} \mathit{\Gamma}(\mathbf{x}_m, s_n) \quad , \tag{4}$$

where $\Gamma(.,.)$ is a binary operator that takes as input an image and a sentence, and outputs 1 if the sentence s_n is relevant for the image $\mathbf{x}_m$ and zero otherwise (see Appendix C.2 of the Supplementary for details). The ACR is the average across all clusters.

iii) $\mathcal{R}_\mathcal{S}$ vs. $\mathcal{S}^*_\beta$. To assess if the union of the retained sentences $\mathcal{R}_\mathcal{S}$ (defined in Eq. (2)) accurately covers the potential errors represented by $\mathcal{S}^*_\beta \subset \mathcal{S}$, we compute the True Positive Rate (TPR) of the retrieved sentences and the Jaccard Index (JI) between the two sets. TPR $= |\mathcal{R}_\mathcal{S} \cap \mathcal{S}^*_\beta|/|\mathcal{S}^*_\beta|$ indicates how well a method covers the ground-truth explanations, while JI $= |\mathcal{R}_\mathcal{S} \cap \mathcal{S}^*_\beta|/|\mathcal{R}_\mathcal{S} \cup \mathcal{S}^*_\beta|$ measures the overall coverage, also taking into account false positives.

5 Experiments

Experimental setup. We tackle three tasks: semantic segmentation of urban scenes, classification in the presence of spuriously correlated data, and large-scale ImageNet classification. For the first, we consider a ConvNeXt [24] segmentation model trained on Cityscapes [3] and test on three challenging datasets, WD2 [44], IDD [37] and ACDC [31]. The second task evaluates ResNet50 image classification models trained on different spuriously correlated data derived from NICO++ and demonstrates how the proposed approaches could be used for understanding model failures due to such subtle biases [12, 47]. We consider two cases, an unsupervised case where the full set is split with class prediction entropy and a supervised one, where following [8, 43] we analyze the performance on each class separately. In the latter case we use the class prediction to split the data into easy and hard sets and, in addition to evaluating LBEE, we compare our simple split-and-cluster partitioning to the more complex partitioning methods from [8, 43], using their evaluation protocol (see in Appendix B of the Supplementary). Finally, to showcase the scalability of our approaches, we analyze different pre-trained architectures on the ImageNet 1K validation set, focusing on the top-1 classification performance and seeking for explanations for each class individually. Details about each task, related models and about the datasets in Appendix A of the Supplementary.

Default design choices. We use Open-CLIP [14] as visual-textual embedding space. To generate the GT sentence set $\mathcal{S}^*_\beta$, we use $\beta = .2 * \omega_\theta^{\text{std}}$ with $\omega_\theta^{\text{std}}$ being the standard deviation of ω_θ computed over $\mathcal{X}$. We set the number of hard clusters to $C = 15$ for large datasets and $C = 5$ for per-class data analyses, and set the same number for the easy set ($C^h = C^e$). We split the data using the output entropy in the unsupervised cases and class probabilities or ranking in the per-class case (see Appendix D in the Supplementary). We retain three sentences for each cluster.

Below, we provide qualitative examples and results with default configurations, dataset-specific analyses as well as sensitivity to various parameters.

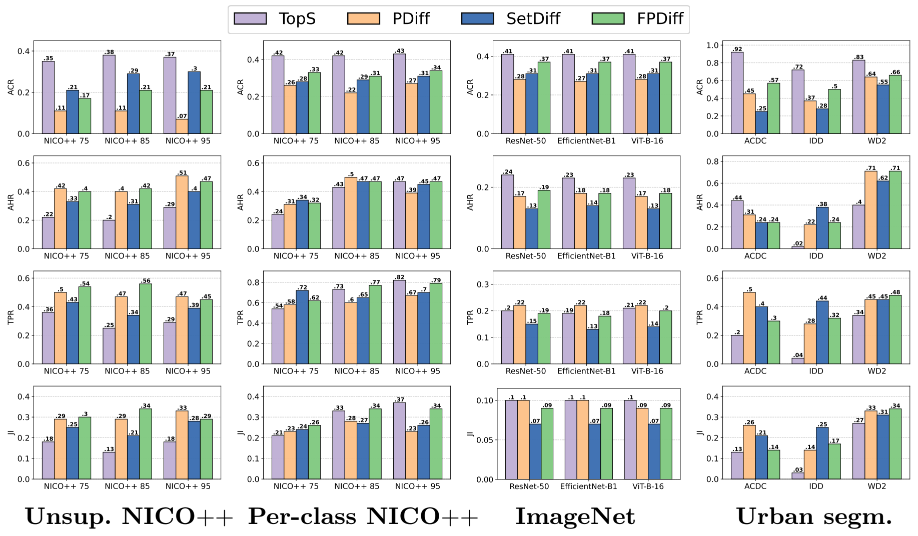

Fig. 2: From top to bottom ACR, AHR, TPR and JI scores on NICO++ unsupervised (first column) and supervised per-class (second column), ImageNet per-class (third column) and Urban Scene Segmentation (last column). For all metrics, higher is better.

5.1 Results

In Figs. 3, 5, 7 and 9 we report qualitative results, showing the sentences selected by our methods for various hard clusters. where each row represents a hard cluster with the three sentences provided by the different methods (further qualitative results are provided in Appendix F of the Supplementary). Focusing, *e.g.*, on urban segmentation, we can see that our methods can successfully isolate the main reasons for clusters to be *hard*, for example difficult visual conditions (*rainy/foggy weather*) or the presence of elements unseen during training (such as *tunnel, mud* or *light reflections*). Indeed, such elements can represent real challenges for a model trained on Cityscapes as observed also *e.g.* in [19].

In Fig. 2, we provide numerical results with the default setting for the different methods, tasks and datasets using the metrics introduced in Sec. 4. First, we observe that **TopS** is the best performing method in terms of ACR scores; this is not surprising since sentences closest to the prototype have obviously the highest coverage. Yet, these sentences are not necessarily pointing to error explanations as indicated by several low AHR scores, with IDD being the most dramatic.

Concerning the metrics related to the failure explanations (AHR/TPR/JI), in both NICO++ scenarios, where we have ground truth relevance scores, and on WD2, **FPDiff** performs best or close to it. It performs worse on ACDC and IDD: there is no clear winner among the proposed methods in these two datasets, since the higher performance of **PDiff** and **SetDiff** in terms of TPR and JI comes at the price of a lower ACR.

Table 1: Examples of sentences from the user-defined sentence set $\mathcal{S}$. For each sentence we show if it belongs to $\mathcal{S}^*_\beta$ and to $\mathcal{R}_\mathcal{S}$ for **TopS** (**TS**), **PDiff** (**PD**), **SetDiff** (**SD**) and **FPDiff** (**FP**) for the datasets WD2, IDD and ACDC. "✓" means $s_n \in \mathcal{S}^*_\beta$ and "×" means $s_n \notin \mathcal{S}^*_\beta$—namely, the hardness score of s_n is below the required level. For the methods, "+"/"+" indicate that the sentence is in $\mathcal{R}_\mathcal{S}$, where "+" means true positive (*i.e.*, the sentence is also in $\mathcal{S}^*_\beta$) while "+" means false positive. An empty space indicates that the sentence was not in the corresponding $\mathcal{R}_\mathcal{S}$.

	WD2					IDD					ACDC				
Sentences (*An image ...*)	$\mathcal{S}^*_\beta$	**TS**	**PD**	**SD**	**FP**	$\mathcal{S}^*_\beta$	**TS**	**PD**	**SD**	**FP**	$\mathcal{S}^*_\beta$	**TS**	**PD**	**SD**	**FP**
"taken at night"	✓	+	+	+	+	✓		+	+	+	✓	+			
"taken in the evening"	✓		+	+	+	✓		+		+	✓	+			
"taken at dusk"	✓		+	+	+	✓		+	+	+	✓				
"taken in a foggy weather"	✓			+		✓					×	+	+	+	+
"taken in a rainy weather"	✓	+	+		+	✓					×	+	+		+
"taken in a snowy weather"	✓		+	+	+	×					×	+			
"taken in a dull weather"	✓	+	+	+	+	×					×	+	+	+	+
"with reflection on the road"	✓	+	+	+	+	✓			+		✓				+
"with shadows on the road"	×				+	×			+		✓			+	
"with water on the road"	✓	+				×					×			+	+
"with mud on the road"	×			+	+	×		+	+		×				
"with obstacle on the road"	✓	+			+	×	+				×				
"with road barrier on the road"	×			+		×			+	+	✓	+	+	+	+
"with rail track on the road"	×					✓			+	+	✓	+	+	+	+
"with rocks on the road"	×			+	+	×					×				
"showing a highway scene"	×			+		×			+	+	×	+	+	+	+
"showing an industrial scene"	✓					✓					×				
"showing construction site"	×					✓		+	+	+	×		+	+	
"showing sub-urban scene"	×	+		+		×	+			+	✓				
"with rickshaw on the road"	×	+	+	+	+	×	+	+		+	✓		+		
"with motorbike on the road"	×		+		+	×				+	✓				+
"with vehicle on the road"	×	+				×	+				×	+	+	+	+
"with tram on the road"	×					×			+		×	+	+		+
"with jeep on the road"	×	+		+	+	×					×				
"with animal on the road"	×			+	+	✓			+	+	×				
"with people on the road"	×	+			+	×	+			+	×		+		
"with crowded foreground"	×					×		+	+	+	×				
"with motion blur"	✓	+	+		+	✓			+	+	✓		+		
"with underexposure"	×			+	+	×					✓			+	
"with low contrast"	✓					×			+	+	×			+	
"of a tunnel"	✓	+	+	+	+	✓		+			✓	+	+	+	+
"of fences"	×					×		+			✓		+		
"of guard-rail"	×					×					×		+	+	+
"of a traffic jam"	✓					✓	+	+		+	×				
"of traffic lights"	✓					✓					×		+		
"of a parking"	×		+			✓		+	+	+	×		+		

In the following (complemented by Appendix E of the Supplementary), we provide more in-depth analysis focusing on the different datasets.

5.2.1. ACDC

In Fig. 2 we observe that for ACDC the best AHR performance is obtained with **TopS** and the best TPR with **PDiff** . To shed light on these results, we complement them with Tab. 1 showing sentences from the user-defined sentence set $\mathcal{S}$ that are either in $\mathcal{S}^*_\beta$ or in any $\mathcal{R}_\mathcal{S}$ (see more sentences in Appendix F).

From these results we can make the following observations. *i)* The main advantage of **TopS** comes from retrieving the sentences related to images taken at night or in the evening. *ii)* The other methods fail in retrieving these sentences because the closest easy clusters also contain night or evening images. *iii)* On the other hand, these methods often select sentences referring to "*fog*", or "*rain*" (see *e.g.* Fig. 3) that are indeed valid characteristics of the images in the cluster, but yet these conditions do not affect sufficiently the model to have a hardness score that satisfies the condition $\omega_\theta^{s_n} < \omega_\theta^{\text{avg}} - \beta$.

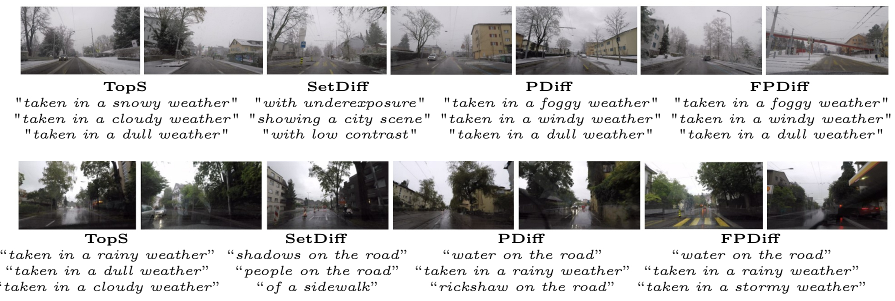
TopS
"taken in a snowy weather"
"taken in a cloudy weather"
"taken in a dull weather"
SetDiff
"with underexposure"
"showing a city scene"
"with low contrast"
PDiff
"taken in a foggy weather"
"taken in a windy weather"
"taken in a dull weather"
FPDiff
"taken in a foggy weather"
"taken in a windy weather"
"taken in a dull weather"
TopS
“taken in a rainy weather”
“taken in a dull weather”
“taken in a cloudy weather”
SetDiff
“shadows on the road”
“people on the road”
“of a sidewalk”
PDiff
“water on the road”
“taken in a rainy weather”
“rickshaw on the road”
FPDiff
“water on the road”
“taken in a rainy weather”
“taken in a stormy weather”

Fig. 3: Two hard clusters in ACDC explained by different methods.

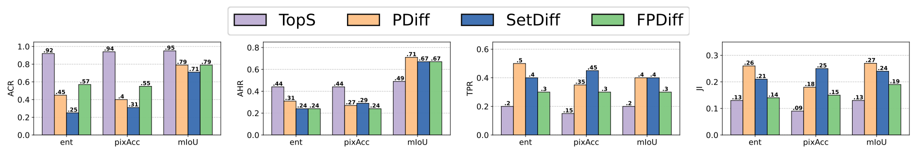

Fig. 4: Unsupervised and supervised splitting (ACDC). Results of different metrics (from left to right ACR, AHR, TPR and JI) on the ACDC dataset when the split is done with entropy, pixel accuracy and the mIoU.

Varying the easy/hard splitting strategy. In order to evaluate the effectiveness of the entropy as a metric to split datasets into easy and hard partitions, in Fig. 4 we compare the results obtained by splitting the ACDC set with entropy *versus* using metrics that rely on annotated samples, namely the pixel accuracy (average of correctly predicted pixels) or mIoU (in all cases we used the $\pm 0.2 * std$ strategy as detailed in Appendix D). We can observe that while using the GT mIoU improved significantly the per cluster scores ACR and AHR, globally the TPR and JI changed only slightly. This suggests that, in general, the sentences retained by **PDiff** , **FPDiff** and **SetDiff** have better coverage in the cluster (higher ACR) and retain more often a hardness score that satisfies the required constraint. The TPR and JI scores are comparable across splitting strategy. By analyzing the sentences retrieved in each setting, we discovered that with GT mIoU the contrastive methods are able to recover the sentences such as “*taken at night/in the evening*” and “*taken at dusk*”, but fails retrieving sentences such as “*with road barrier/rail track on the road*” and “*with motion blur/underexposure*”, sentences that were retrieved when we used the entropy.

Finally, sentences referring to *fog* and *rain* having scores slightly below the required condition $\omega_\theta^{s_n} < \omega_\theta^{\text{avg}} - \beta$, are present in $\mathcal{R}_\mathcal{S}$ for all splitting strategies and most methods. They could be interpreted as less severe false positives.

5.2.2 IDD

In the case of IDD, the best AHR/TPR/JI results in Fig. 2 are obtained with **SetDiff** . The lower ACR suggests that the sentences retained by this method have lower coverage in the cluster than the ones retained by the other methods.

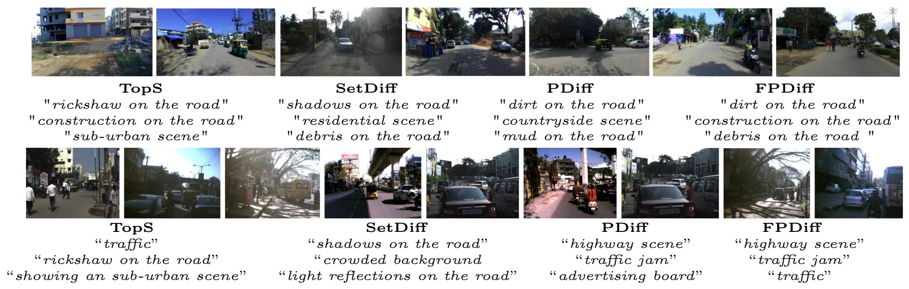

Fig. 5: Two out of 15 hard clusters from IDD explained by different methods.

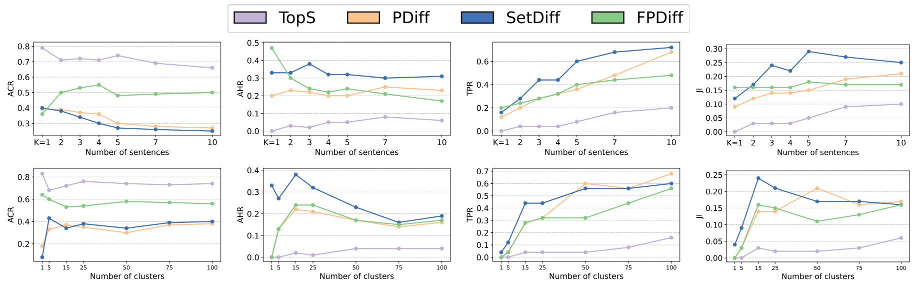

Fig. 6: From left to right, we show how the ACR, AHR, TPR and JI metrics vary on the IDD experiment, by **varying the number of sentences selected for each cluster** K **(top)** and **varying the number of clusters** C **(bottom)**. The very poor numbers corresponding to $C = 1$ (not performing any clustering just contrasting the easy versus hard sets) emphasizes the importance of clustering.

IDD is a large dataset with high content variation posing more challenges. Splitting into 15 clusters makes the content in each cluster very heterogeneous (see Fig. 5) and the selection of the three most *representative sentences* rather difficult. We complement these analyses with two further experiments on IDD where we vary the number of selected sentences or we vary the number of clusters.

Varying the number of retained sentences (K**).** We show in Fig. 6 (top) results for IDD when we vary the number of retained sentences per cluster. We observe that with the increase of the number of sentences retained, ACR increases for **FPDiff** , but decreases for the other methods; yet, ACR is stable starting from $K = 5$. On the contrary, AHR is generally stable as we vary K, except for **FPDiff** , where the best value is obtained with $K = 1$. Both TPR and JI benefit from the increase of the number of sentences (increasing K). Overall, **SetDiff** outperforms the other methods in terms of AHR, TPR and JI. This result, combined with the low ACR, suggests that **SetDiff** is more capable of detecting rarer failure reasons, for which there are less support example images in the cluster/dataset.

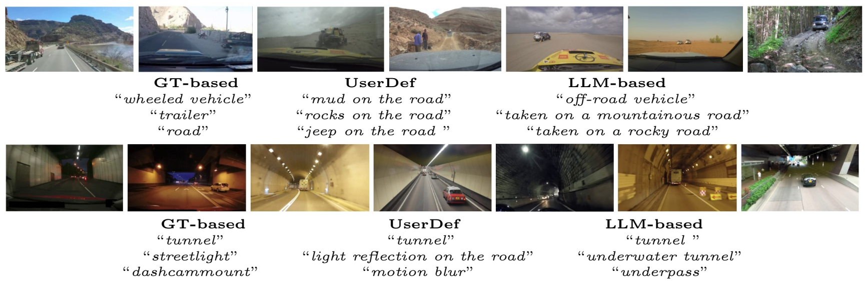

Fig. 7: Two out of 15 hard clusters from WD2 explained by **FNDiff** when we use different the sentence sets (GT based, User defined or LLM generated).

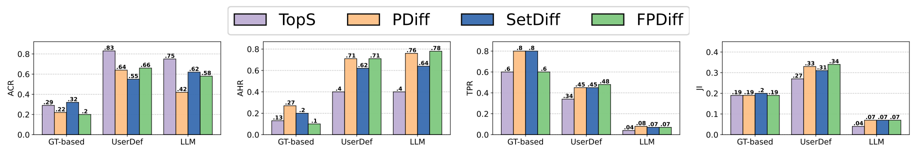

Fig. 8: Varying the sentence set. Results obtained on WD2 when $\mathcal{S}$ is the GT-based, User-defined and the LLM-generated sentence set (see Appendix C.1).

Varying the number of clusters (C). In Fig. 6 (bottom) we present results obtained on IDD when we vary the number of clusters ($|\mathcal{C}^h| = |\mathcal{C}^e| = C$). By increasing the number of clusters, we observe an increase of the TPR (as expected) for all methods—with **PDiff** being the one that benefits the most. Concerning the other metrics, the default value $C = 15$ is a reasonable trade off. These experiments suggest that IDD is overall a very challenging set, with some of the failure causes under-represented—especially the ones related to the weather conditions (see sentences in Tab. 1), or not clustered together according to these characteristics—*cf.* Fig. 5. Furthermore, as the filtering in **FPDiff** relies on the **TopS** ranking, the very poor performance of **TopS** on this particular set (IDD) affects negatively the performance of **FPDiff** compared to **PDiff** especially when we increase the number of clusters or retained sentences. Finally, the low performance obtained with $C = 1$ (no clustering) shows that describing the full hard set by contrasting with the full easy set is a sub-optimal solution.

5.2.3 WD2

We analyse on WD2 the performance of the different methods, varying the sentence set $\mathcal{S}$, as, for this dataset, we have User-defined, GT-based as well as a significantly larger LLM-based sentence set (see Appendix C.1).

Sensitivity to the choice of the sentence set. First, we notice in Fig. 8 that both the User-defined and the LLM-generated sentence sets yield similar ACR and AHR scores. Yet, for the latter we have many sentences that are semantically related and carry similar hardness scores, for example "*taken at night*" and "*taken at mesmerizing moonlight* /'*spellbound starlight* / *miraculous*

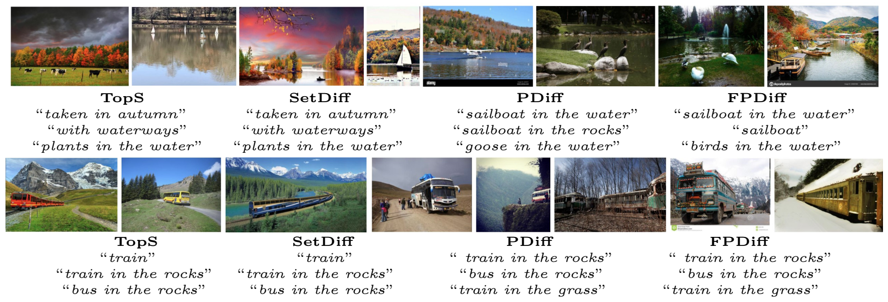

Fig. 9: Two hard clusters from NICO$^{85}_{++}$ (unsup. case) explained by different methods.

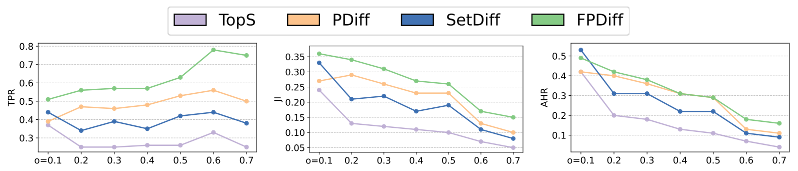

Fig. 10: Varying β (by varying o). Results obtained on NICO$^{85}_{++}$ (unsupervised). The value β defines the GT set $\mathcal{S}^*$, so its value impacts only the AHR/TPR/JI metrics (not the output of the methods or ACR). On x-axis: we plot o, where $\beta = o \cdot \omega_\theta^{\text{std}}$ with $\omega_\theta^{\text{std}}$ being the standard deviation of ω_θ (entropy) computed over $\mathcal{X}$.

midnight". Since we limit the number of sentences retained by each method, we cannot retrieve them all hence the much lower TPR and JI scores.

The GT-based sentence set mainly describes the presence of the classes ("*Image showing* <class>"), therefore, it is not sufficient to describe the hard clusters to the desired level of details. Further, with $o = 0.2$, $\mathcal{S}^*_\beta$ contains only five sentences that are: image showing *tunnel*, *bridge*, *traffic light*, *bus* or *snow*. **PDiff** and **SetDiff** were able to retain four out of five sentences (except "*showing a bus*") and **TopS** and **FNDiff** three out of five (missing also "*showing a bridge*"); the missing sentences had not enough image coverage in the cluster. Still, all methods introduce false positives for clusters not containing these elements (which explains the low JI score). The low ACR scores (even for **TopS**) suggest that it is not easy to find shared content for all clusters These results highlights the importance of properly designing $\mathcal{S}$.

5.2.4. NICO$^{85}_{++}$

In Fig. 9 we provide qualitative examples for NICO$^{85}_{++}$ (in the unsupervised case using prediction entropy to split the data) showing that our methods can capture the "class-context" associations rarely seen in the training set (*landways-rocks*, *birds-water*) making the model struggling with their classification. We also use this dataset to evaluate the methods when we vary the hardness level β.

Varying β (by varying o). Recall that β is the hyper-parameter (margin) used to select the ground-truth sentence set $\mathcal{S}^*$, i.e sentences with a score $\omega_\theta^{s_n}$

below $\omega_\theta^{\text{avg}} - \beta$. Lower β means more explanations accepted as valid (including causes affecting less strongly the model), while higher β is more restrictive focusing on causes that drops more the model's performance. In Fig. 10 we analyse the impact of varying β on the AHR, TPR, and JI metrics using the spurious classification task on NICO$^{85}_{++}$ (unsup. case), where we have GT image-sentence relevance scores. As the number of sentences retained by the method is fixed to $K = 3$, varying β only affects $\mathcal{S}^*$ and not $\mathcal{R}_\mathcal{S}$. Therefore, the value of ACR (that does not depend on β) is the same for each o value. Intuitively, with this study we assess how much each method focuses on the hardest sentences. Indeed, increasing the value of o (and hence increasing β) corresponds to accepting fewer and fewer sentences as valid explanations, namely the ones with the highest hardness score (lowest average accuracy). We observe that the **FPDiff** method, while also experiencing a drop in performance as we make the problem harder, holds a better performance than the other methods for all the values of o.

6 Concluding remarks

Assessing model performance and in particular failure modes for arbitrary task in arbitrary environments should be *easy*, *efficient* and *interpretable*. While the gold standard for performance evaluation is using a human-annotated test set, this process is costly and cannot scale for computer vision models to be deployed in many and diverse scenarios. Moreover, quantitative evaluation measures such as dataset-level accuracy values do not fully reflect the details of the model performance. It is important to know what lies beyond performance values and if the model is failing on particularly critical images.

In this work, we posit that it is of the utmost importance being able to assess model performance on non-annotated samples, and to move beyond hard-to-interpret numbers. We address this by first formulating the LBEE problem and designing a family of *task-agnostic* approaches that do not require error specifications or user-provided annotations. We complement these two contributions with different metrics to benchmark methods for LBEE and rely on those to carry out an in-depth analysis of the methods we introduce. We showcase how we can retrieve relevant sentences pointing to important errors, *e.g.*, related to environments beyond the model's comfort zone.

In the future, we plan to overcome the need to provide the sentence set $\mathcal{S}$, a design choice that makes evaluation more tractable, but that also limits our methods. Defining the sentence set $\mathcal{S}$ a priori has the advantage that our methods can be deployed off the shelf without any overhead, but yet the set might omit certain unexpected failure reasons or unforeseen factors. A possible solution could be to build the sentence set $\mathcal{S}$ directly from the target set $\mathcal{X}$, *e.g.*, by running a VLM to describe all images in it and merge these descriptions into the sentence set. However, in addition to being prohibitively expensive, a solution like this needs to be iterated for every new target set before running any method. Furthermore, in the case of a large image set, it would also require further steps to remove redundancy and to regroup similar sentences.

References

1. Antol, S., Agrawal, A., Lu, J., Mitchell, M., Batra, D., Zitnick, L., Parikh, D.: VQA: Visual Question Answering. In: ICCV (2015)
2. Augustin, M., Boreiko, V., Croce, F., Hein, M.: Diffusion Visual Counterfactual Explanations. In: NeurIPS (2022)
3. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The Cityscapes Dataset for Semantic Urban Scene Understanding. In: CVPR (2016)
4. d'Eon, G., d'Eon, J., Wright, J.R., Leyton-Brown, K.: The Spotlight: A General Method for Discovering Systematic Errors in Deep Learning Models. In: ACM FAccT (2022)
5. Doshi-Velez, F., Kim, B.: Towards A Rigorous Science of Interpretable Machine Learning. arXiv:1702.08608 (2017)
6. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In: ICLR (2021)
7. Dunlap, L., Zhang, Y., Wang, X., Zhong, R., Darrell, T., Steinhardt, J., Gonzalez, J.E., Yeung-Levy, S.: Describing Differences in Image Sets with Natural Language. In: CVPR (2024)
8. Eyuboglu, S., Varma, M., Saab, K., Delbrouck, J.B., Lee-Messer, C., Dunnmon, J., Zou, J., Ré, C.: Domino: Discovering Systematic Errors with Cross-Modal Embeddings. In: ICLR (2022)
9. Gao, I., Ilharco, G., Lundberg, S., Ribeiro, M.T.: Adaptive Testing of Computer Vision Models. In: ICCV (2023)
10. Goyal, Y., Wu, Z., Ernst, J., Batra, D., Parikh, D., Lee, S.: Counterfactual Visual Explanations. In: ICML (2019)
11. He, K., Zhang, X., Ren, S., Sun, J.: Deep Residual Learning for Image Recognition. In: CVPR (2016)
12. Hendricks, L.A., Burns, K., Saenko, K., Darrell, T., Rohrbach, A.: Women also Snowboard: Overcoming Bias in Captioning Models. In: ECCV (2018)
13. Hendricks, L.A., Hu, R., Darrell, T., Akata, Z.: Generating Counterfactual Explanations with Natural Language. In: ICML Workshops (2018)
14. Ilharco, G., Wortsman, M., Wightman, R., Gordon, C., Carlini, N., Taori, R., Dave, A., Shankar, V., Namkoong, H., Miller, J., Hajishirzi, H., Farhadi, A., Schmidt, L.: OpenCLIP. Zenodo, `https://github.com/mlfoundations/open_clip` (2021)
15. Jacob, P., Zablocki, É., Ben-Younes, H., Chen, M., Pérez, P., Cord, M.: STEEX: Steering Counterfactual Explanations with Semantics. In: ECCV (2022)
16. Jain, S., Lawrence, H., Moitra, A., Madry, A.: Distilling Model Failures as Directions in Latent Space. In: ICLR (2023)
17. Jain, S., Salman, H., Wong, E., Zhang, P., Vineet, V., Vemprala, S., Madry, A.: Missingness Bias in Model Debugging. In: ICLR (2022)
18. Jeanneret, G., Simon, L., Jurie, F.: Diffusion Models for Counterfactual Explanations. In: ACCV (2022)
19. de Jorge, P., Volpi, R., Torr, P., Gregory, R.: Reliability in Semantic Segmentation: Are We on the Right Track? In: CVPR (2023)
20. Kim, Y., Mo, S., Kim, M., Lee, K., Lee, J., Shin, J.: Discovering and Mitigating Visual Biases through Keyword Explanation. In: CVPR (2024)

21. Li, J., Li, D., Xiong, C., Hoi, S.: BLIP: Bootstrapping Language-Image Pre-training for Unified Vision-Language Understanding and Generatione. In: ICML (2022)
22. Liang, W., Zhang, Y., Kwon, Y., Yeung, S., Zou, J.: Mind the Gap: Understanding the Modality Gap in Multi-modal Contrastive Representation Learning. In: NeurIPS (2022)
23. Liu, H., Li, C., Wu, Q., Lee, Y.J.: Visual Instruction Tuning. In: NeurIPS (2023)
24. Liu, Z., Mao, H., Wu, C.Y., Feichtenhofer, C., Darrell, T., Xie, S.: A ConvNet for the 2020s. In: CVPR (2022)
25. Metzen, J.H., Hutmacher, R., Hua, N.G., Boreiko, V., Zhang, D.: Identification of Systematic Errors of Image Classifiers on Rare Subgroups. In: ICCV (2023)
26. Mokady, R., Hertz, A., Bermano, A.H.: ClipCap: CLIP Prefix for Image Captioning. arXiv:2111.09734 (2021)
27. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, Girishand Askell, A., Mishkin, P., Clark, J., Krueger, G., Sutskever, I.: Learning Transferable Visual Models From Natural Language Supervision. In: ICML (2021)
28. Rezaei, K., Saberi, M., Moayeri, M., Feizi, S.: PRIME: Prioritizing Interpretability in Failure Mode Extraction. In: ICLR (2024)
29. Ribeiro, M.T., Singh, S., Guestrin, C.: 'Why Should I Trust You?' Explaining the Predictions of any Classifier. In: SIGKDD (2016)
30. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M.S., Berg, A.C., Fei-Fei, L.: ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision (IJCV) **115**, 211–252 (2015)
31. Sakaridis, C., Dai, D., Van Gool, L.: ACDC: The Adverse Conditions Dataset with Correspondences for Semantic Driving Scene Understanding. In: ICCV (2021)
32. Schuhmann, C., Beaumont, R., Vencu, R., Gordon, C., Wightman, R., Cherti, M., Coombes, T., Katta, A., Mullis, C., Wortsman, M., Schramowski, P., Kundurthy, S., Crowson, K., Schmidt, L., Kaczmarczyk, R., Jitsev, J.: LAION-5B: An Open Large-scale Dataset for Training Next Generation Image-text Models. In: NeurIPS (2022)
33. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization. In: ICCV (2017)
34. Singla, S., Nushi, B., Shah, S., Kamar, E., Horvitz, E.: Understanding Failures of Deep Networks via Robust Feature Extraction. In: CVPR (2021)
35. Tan, M., Le, Q.V.: EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In: ICML (2019)
36. Tursun, O., Denma, S., Sridharan, S., Fookes, C.: Towards Self-Explainability of Deep Neural Networks with Heatmap Captioning and Large-Language Models. arXiv:2304.02202 (2023)
37. Varma, G., Subramanian, A., Namboodiri, A., Chandraker, M., Jawahar, C.: IDD: A Dataset for Exploring Problems of Autonomous Navigation in Unconstrained Environments. In: WACV (2019)
38. Vendrow, J., Jain, S., Engstrom, L., Madry, A.: Dataset Interfaces: Diagnosing Model Failures Using Controllable Counterfactual Generation. In: ICML Workshops (2024)
39. Wang, P., Yang, A., Men, R., Lin, J., Bai, S., Li, Z., Ma, J., Zhou, C., Zhou, J., Yang, H.: OFA: Unifying Architectures, Tasks, and Modalities Through a Simple Sequence-to-Sequence Learning Framework. In: ICML (2022)
40. Ward, J.H.: Hierarchical Grouping to Optimize an Objective Function. Journal of the American Statistical Association **58**(301), 236—-244 (1963)

41. Wiles, O., Albuquerque, I., Gowal, S.: Discovering Bugs in Vision Models using Off-the-shelf Image Generation and Captioning. In: NeurIPS (2022)
42. Wong, E., Santurkar, S., Madry, A.: Leveraging Sparse Linear Layers for Debuggable Deep Networks. In: ICML (2021)
43. Yenamandra, S., Ramesh, P., Prabhu, V., Hoffman, J.: FACTS: First Amplify Correlations and then Slice to Discover Bias. In: ICCV (2023)
44. Zendel, O., Honauer, K., Murschitz, M., Steininger, D., Fernandez Dominguez, G.: WildDash - Creating Hazard-Aware Benchmarks. In: ECCV (2018)
45. Zhang, J., Bargal, S.A., Lin, Z., Brandt, J., Shen, X., Sclaroff, S.: Top-down Neural Attention by Excitation Backprop. In: ECCV (2018)
46. Zhang, X., He, Y., Xu, R., Yu, H., Shen, Z., Cui, P.: NICO++: Towards Better Benchmarking for Domain Generalization. In: CVPR (2023)
47. Zhao, J., Wang, T., Yatskar, M., Ordonez, V., Chang, K.W.: Men Also Like Shopping: Reducing Gender Bias Amplification using Corpus-level Constraints. In: EMNLP (2017)
48. Zhou, B., Lapedriza, A., Khosla, A., Oliva, A., Torralba, A.: Places: A 10 million Image Database for Scene Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) **40**(6), 1452–1464 (2018)

Appendix

This Supplementary is organized as follows. In Appendix A we describe the tasks and related datasets used in our experiments. Then we provide a comparison of our method with prior art on error-mode discovery methods (Appendix B), following the evaluation protocol of [43]. Next, in Appendix C.1 we provide details regarding the sentence sets used in our experiments. In Appendix C.2 we specify how we define the ground-truth sentences for a given task in LBEE. In Appendix C.3 we describe the VQA process used to generate pseudo-GT relevance scores between a sentence and an image when this information is not available. In Appendix D we recall and detail our default design choices and in Appendix E we provide an in-depth analyses of our ImageNet 1K results. Finally, in Appendix F we provide additional qualitative results.

A Datasets and Tasks

In the following, we detail the three main tasks we tackle in this paper: semantic segmentation of urban scenes, classification in the presence of spuriously correlated data, and ImageNet-1K classification.

Urban scene segmentation. We consider a ConvNeXt [24] segmentation model trained on Cityscapes [3], comprised of images from 50 European cities collected at daytime in clear weather conditions. We build our evaluation sets $\mathcal{X}$ using three datasets: *i)* WildDash 2 (WD2) [44], which contains challenging visual conditions such as motion blur, various road types, difficult weather, *etc.* and *ii)* India Driving Dataset (IDD) [37], which contains images from Hyderabad, Bangalore and their peripheries, *iii)* ACDC [31] contains images recorded in adverse conditions, namely *fog*, *rain*, *snow* and *night* representing challenging domain shift for the Cityscapes model. For the main experiments we use the user-defined sentence set $\mathcal{S}$ with 148 sentences describing content related to urban scenes, road conditions, and image quality (see details in Appendix C). In Sec. 5.2.3 main paper we compare these results with the results obtained on WD2 when using both a smaller GT-based sentence set derived from the metadata information available with the dataset and a larger sentence set of 1016 sentences generated automatically by an LLM (both sets detailed in Appendix C).

Classification with spurious correlations. We use the NICO++ [46] dataset, consisting of real-world images of concept classes (*mammals*, *birds*, *plants*, *airways*, *landways* and *waterways*) taken in six contexts (*dim lighting*, *outdoor*, *grass*, *rock*, *autumn*, *water*). We follow the train, biased validation, and unbiased test splits from [43], aimed at creating different levels of class-context correlations. We use three partitionings NICO^{75}_{++}, NICO^{85}_{++} and NICO^{95}_{++} with increased levels of correlation.[4] We train a ResNet-50 model on each training set, and use the unbiased test set for evaluation.

[4] This means setting the correlation level to 75%, 85% and 95% respectively between a class and its natural context in the training/validation set, as detailed in [43].

To evaluate error descriptions of the hard clusters, we build a sentence set $\mathcal{S}$ with 130 sentences that include information related to the the six classes, the six contexts, various sub-classes and their combination. Note that in this case we have GT relevance $\Gamma(\mathbf{x}_m, s_n)$ between sentences and images.

We consider two cases, an unsupervised and a supervised one. In the former case, we consider the full set and split it with entropy values computed on class predictions. In the latter case, similarly to [8, 43], we consider images from each class separately and use the class probability scores to split the data into easy and hard sets.

ImageNet-1K classification. We consider three different architectures (ResNet-50 [11], VIT-B-16 [6], and EfficientNet-B1 [35]) trained on the ImageNet-1K [30] training set and evaluate these models on the ImageNet-1K validation set. We build $\mathcal{S}$ as the union of a set of sentences {"*An image of a* `<class>`"} (for the 1K classes) with a set of sentences corresponding to some place classes from Places365 [48] in form of {"*An image taken at* `<location>`"}, and a set of sentences describing image type (photo, drawing), image quality (blurry, noise, JPEG compression), weather conditions, *etc.* In total we have 1417 sentences (see details in Appendix C). To assess the relevance between a sentence and an image, we use a combination of ground truth based information (for the first 1000 sentences corresponding to the 1000 class names) and a VQA model to assess the relevance for the other sentences (as described in Appendix C.3).

With this dataset, we focus on the top-1 classification performance of different models and aim at analyzing failure cases for this task—namely, the reasons for not predicting the correct class on top. In ImageNet classification, several errors arise from confusions between semantically similar classes. For example, the model will miss-classify dog breeds, bird/fish species, similar objects, *etc.* In such cases, knowing the correct class label can help to deduce the main reason from the retrieved description. Indeed, for example, if the selected sentence is {"*An image of a water snake*"} without the knowledge of the correct class, it is difficult to interpret whether the model predicts false positives (images of different type of snakes) or the model failed to recognize the water snake (false negative) for some reasons. Furthermore, while, class independent failure reasons such as those related to image capturing conditions (*e.g.* blurred or low contrast images) might emerge when processing the full dataset as a whole, errors related to the image content is more difficult to interpret if it is a failure reason without the knowledge of the target class (*e.g.* finding that the image was taken at a *beach* has different effect on the model for the *swimming suit* class than for the *cow* or *ski suit* class). Finally, even if we use a very large number of clusters with the full set, the content of the nearest easy cluster might differ in many aspects from the hard one, making rather it difficult to select in priority which is the one causing the model to fail.

For all these reasons we opted to follow the prior art and apply LBEE methods by analyzing the data for each class independently. Still, in contrast to prior art, where only false negative errors are considered (the images that belong to the given class but labeled with a wrong class label), we also include false positives.

Hence, the easy set $\mathcal{X}^e$ contains the images where the class has been correctly predicted (images from the current class) while the hard set is a union of false positives and false negatives.

B Comparison with prior art

DOMINO [8] and FACTS [43] mainly focus on evaluating the error mode discovery [8, 43], assuming to know the set of error types, and limit the language description of these error modes to a qualitative role. In contrast, in our problem formulation, we treat the language-based description of errors made by computer vision models as the problem itself. For this reason, a direct comparison with prior art is not straightforward. In this section, we make an attempt by simply comparing our split-and-cluster partitioning to the more complex, learnt partitioning methods by [8, 43], following their evaluation protocol.

We perform this analysis using the NICO^{75}_{++} , NICO^{85}_{++} and NICO^{95}_{++} sets from [43], corresponding to a 75%, 85% and 95% correlation level, respectively, between the classes and their natural context in the training set. This artificially introduces strong spurious correlations between classes and contexts, makes the model to fail in scenarios where the test set does not follow the same biased distribution seen at training.

To tackle this setting, [43] propose to learn a partitioning Gaussian Mixture model on a *Biased* validation set that follows the same bias as the training set and evaluate the partitioning on an *Unbiased* set where all class-context are equally represented (see statistics in Tab. 2). Note that this is done for each class independently (per-class case). Using the code provided by [43][5], we learn and evaluate all possible set combinations (see Tab. 3). Note that (*Biased*-*Biased*) and (*Unbiased*-*Unbiased*) sets means that we learnt and test the partitioning function on the exact same data.

In our case, there is no learning involved: we simply split and cluster the data as discussed in Sec. 3 in the main paper. Then, for a given (A, B) pair of sets, where $A, B \in \{\textit{Biased}, \textit{Unbiased}\}$, we assign images from the set B to the cluster prototypes obtained with the images in set A, where the union of easy and hard prototypes is considered. We use the class probability to split the data—the same measure used by FACTS [43] to learn the partitioning function, except that we only use the probability of the target class while they use the probabilities of all classes.

Evaluation protocol. Before discussing the comparative results, we recall the metrics and the evaluation protocol we use, proposed by [8, 43]. In particular they propose to rely on the Precision-at-K (**P@K**) to measure how accurately a partitioning method aggregates samples from the same context given a class. Formally, let $\mathcal{X}$ denote the test set, $\{\mathbf{z}_n\}_{n=1}^N$ the ground-truth partitions and $\{\widehat{\mathbf{z}}_m\}_{m=1}^M$ the predicted set of discovered partitions. Given K, for each ground-truth partition $\mathbf{z}_n$ the most "similar" predicted partition $\widehat{\mathbf{z}}_m$ is selected

[5] `https://github.com/yvsriram/FACTS`

Table 2: The number of images in the *Biased* validation set for each (class, context) pair in corresponding three NICO++ datasets. The *Unbiased* validation sets have 50 images for each (class, context) pairs in each of the three sets. The spurious contexts (1 to 6) are in order *rock, grass, dim lighting, autumn, water, outdoor*.

NICO++ set	NICO$^{95}_{++}$						NICO$^{85}_{++}$						NICO$^{75}_{++}$					
class ↓ context →	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
mammals	638	4	2	2	3	3	638	4	2	2	3	3	638	10	5	7	8	9
birds	7	321	2	2	3	3	7	321	2	2	3	3	19	321	5	7	8	9
plants	7	4	154	2	3	3	7	4	154	2	3	3	19	10	154	7	8	9
airways	7	4	2	220	3	3	7	4	2	220	3	3	19	10	5	220	8	9
waterways	7	4	2	2	266	3	7	4	2	2	266	3	19	10	5	7	266	9
landways	7	4	2	2	3	277	7	4	2	2	3	277	19	10	5	7	8	277

based on the size of the intersection $\mathbf{z}_n \cap \widehat{\mathbf{z}}^K_m$, where $\widehat{\mathbf{z}}^K_m$ is the subset containing the top-K elements of $\mathcal{X}$ according to the likelihood of belonging to $\widehat{\mathbf{z}}_m$. For FACTS/DOMINO the ranking corresponds to the partitioning function's output, while in our case to the assignment to the closest prototype. If we denote by $\widehat{\mathbf{z}}^K_{nm}$ the set that has the largest overlap with $\mathbf{z}_n$ given K, the **P@K** is defined as

$$P@K = \frac{1}{N}\sum_{n=1}^{N}\frac{|\mathbf{z}_n \cap \widehat{\mathbf{z}}^K_{nm}|}{K}$$

where the average is computed, as in [43], over the five bias-conflicting groups (excluding the spuriously correlated GT partitions, such as the context *water* for *waterways*).

Experimental results. We show in Tab. 3 the **P@K** values averaged over the six classes for the different NICO++ sets. Each time we show results averaged over three different variants of the current model trained with different seed[6]. In our case, we used the exact same ResNet50 models and the same *Biased* and *Unbiased* sets on which we perform the split and the clustering. We use five easy and five hard clusters as it is our default setting for the per-class case and as in the code from [43] the default number of slices was also set to ten.

From Tab. 3 we can see that our split-and-cluster method provides competitive, or sometimes even better, slice discovery performance than the more complex partition learning methods. Our method performs slightly worse than FACTS on the *Unbiased* set both when their partitioning was trained on the *Unbiased* or on the *Biased* set. Surprisingly though, both FACTS and DOMINO underperform when tested on the *Biased* set even when the partitioning function was trained on this set. In contrast, our method is more robust and performs similarly on both sets. When the FACTS model is trained on the *Biased* set,

[6] Since we were not able to reproduce the original paper's numbers with the code, namely, the results corresponding to the (*Biased*,*Unbiased*) pairs, for fairness we ran their code with three different seeds and provide here the averaged results. We do not show the variances in the table to keep it simple, but the values are most often between 1 and 2.

Table 3: Comparisons with prior art. We compare the hard clusters obtained via Step 3 of our proposed methods (see Fig. 1 in the main paper) with the Domino and FACTS methods using averaged **P@K** scores with $K = 10$. For our method, we split the dataset into 5 "easy" and 5 "hard" clusters and use their union as final partitioning. On the top rows we show results obtained with the partitioning learnt on the *Unbiased* set and on the bottom the results with the partitioning learnt on the *Biased* set.

Method		NICO^{75}_{++}		NICO^{85}_{++}		NICO^{95}_{++}	
		Unbiased	**Biased**	**Unbiased**	**Biased**	**Unbiased**	**Biased**
Random		29.7	12.6	32.0	14.0	33.0	14.6
Domino [8]	**Unbias.**	44.4	12.1	47.0	12.3	52.7	22.3
FACTS [43]	**Unbias.**	**58.8**	28.1	**63.6**	30.0	**68.9**	39.9
Ours(5-5)	**Unbias.**	53.6	**52.3**	54.2	**58.4**	55.3	**66.1**
Domino [8]	**Biased**	26.0	17.1	19.0	19.8	17.4	25.6
FACTS [43]	**Biased**	**53.3**	31.4	**53.2**	28.7	**58.9**	34.4
Ours(5-5)	**Biased**	51.3	**52.2**	46.8	**52.5**	41.9	**46.5**

which strongly relies on priors about class-context associations (the configuration shown in their paper), it outperforms our model, for which the P@K decreases as the amount of class-context correlation increases (bottom part of Tab. 3). We hypothesize that the reason behind this result is that we do not have enough data for a proper clustering (see Tab. 2), while FACTS manages to learn the partitioning function even with a a few samples by exploiting the strong correlation between the classes and dominant context. As the model is trained mainly with images with specific combinations of class and correlated context—for example, mammals in rocky environments—the model rarely sees images from other classes in this context. Consequently, when the model learns to predict a class "c" with high probability, it also associates this high probability with the presence of the corresponding context (*e.g.*rocks) even if the class (*mammals*) is absent. Therefore, when analyzing another class, such as *plants*, and learn the partitioning based on all six probabilities, the partitioning function primarily learns to assigns images based on the highest probability score, which is an indicator of the context. For instance, a plant image taken in a rocky environment will have a high probability score for the mammal class, indicating the presence of rocks. Note that in our case, when analyzing the plant class, we disregard the probabilities of other classes, thereby using much less prior information about the image context.

In summary, despite the simplicity of our method, it performs favorably to prior art in their proposed settings, while also being able to provide natural language descriptions of error modes as shown in Sec. 5 in the main paper.

C Defining the sentence sets

Our paper focuses on two objectives, 1) identifying error modes in computer vision models and 2) describing them with natural language. To achieve this, we rely on a large set of sentences ($\mathcal{S}$), that can either be provided or automatically generated. This sentence set is expected to cover various potential reasons for model failures given a task. In this section, we first discuss three approaches to construct such sentence sets (Appendix C.1); then we detail how we create the ground-truth sentence set $\mathcal{S}^*_\beta$ for LBEE (Appendix C.2); finally, we show how to obtain pseudo-GT relevance scores between images and sentences using Visual-Question Answering (VQA) tools when not available otherwise (Appendix C.3).

C.1 Building the sentence set $\mathcal{S}$

There are several possibilities to construct the sentence sets $\mathcal{S}$. We describe a few possibilities below.

User-defined. The set of sentences can be designed ad hoc for the task at hand by the end user who has the expert knowledge and can envisage the main difficulties a model might face. While such sentence set might be non-exhaustive, it can cover the most important failure causes. Our method aim to select from them the ones that confirm the user's concerns (with additional visual support). In this case, to get pseudo-ground-truth relevance between images and a sentence, we resort to VQA (as described in Appendix C.3).

We follow this setting to build $\mathcal{S}$ in our experiments on the urban scene segmentation task. More specifically, we manually defined 148 sentences describing content related to urban scenes (buildings, pedestrians, traffic, two-wheels, trees, animals, garbage, *etc.*) in the form of {"*An image of a* `<urban scene content>`"}, road conditions (weather, lighting, season) in form of {"*An image taken in/at* `<condition>`"} or image quality (underexposure, motion blur, *etc.*) in form of {"*An image with* `<effect>`"}. A large subset of this sentence set can be seen in Tab. 5.

GT-based. If available, we can use ground-truth information (class labels and metadata) to generate a sentence set. This is quite limiting in general: we mainly use this option because it yields the advantage of providing the GT relevance between images and sentences. We used this option in all NICO++ experiments, where we build 130 sentences that include combinations from the six classes, the six contexts, and various sub-class information. For example, we have {"*A photo of* `<sub-class>` *in the* `<context>`"} such as "*A photo of a cactus in the water*".

We also built a GT-based sentence set for the WD2 dataset [44] tested in the context of urban scene segmentation task. It contains annotations for 81 classes, from which we selected 74 classes (removing the ambiguous ones most difficult for VLMs to handle, namely *'unlabeled'*, *'ego-vehicle'*, *'overlay'*, *'out-of-roi'*, *'static'*, *'dynamic'*, *'curb terrain'*). The sentences have the form {"*An image showing a* `<class>`"}, where `<class>` was replaced each time by one of the 74 classes selected. While this allows us to generate GT relevance scores for each image in WD2, this set limits our methods in selecting only from failure causes related to the presence of these classes in the image and hence they do not cover all the possible failure reasons that may explain the model's behavior.

In the case of ImageNet-1K [30], the first 1K sentences (out of 1417) were also generated from the GT information, namely the class names. We consider sentences with the form "{An image of a `<class>`"}. This was complemented by a set of user-defined sentence set related to various places in the form {"*An image taken in* `<location>`"} where we filled the `<location>` with the place class names from Places365 [48]. Finally we also added a set of sentences describing image type (photo, drawing), image quality (blurry, noise, JPEG compression), weather conditions, *etc.* yielding to a total of 1417 sentence set used for ImageNet-1K.

LLM-generated. In the case of urban scene segmentation task, we also consider a set of sentences generated by a large language model (LLM). We provided GPT-3.5 with the context and the sentence structure and prompted it to provide lists of sentences related to weather conditions, vehicles, urban objects, time of day and road conditions. For each of them we used either the sentence form {"*An image taken in/at* <`weather/road condition` or `time of day>`"} or {"*An image showing a/an* <`vehicle/urban object>`"}, where the template is filled by the LLM. This yields to a set of 1016 sentences, including most of the ones we design manually in the User-defined set. On the other end, the limitation of this approach is the noise in the set, which can also include sentences hardly helpful for the task at end, *e.g.*"*An image taken on an unpredictable/eclectic/dystopian road*" or "*An image taken in a corn maze/trick-or-treat/pajama weather*".

C.2 Defining the ground-truth $\mathcal{S}^*_\beta$ for LBEE

To evaluate the predicted sentence set $\mathcal{R}_\mathcal{S} \subset \mathcal{S}$ output by an LBEE approach, we must define a set of ground-truth sentences $\mathcal{S}^* \subset \mathcal{S}$ to compare our predictions to—namely, ground-truth error mode descriptions. To derive such GT set, we consider the function $\omega_\theta : \mathbb{R}^{H\times W\times 3} \rightarrow \mathbb{R}$ that measures the model's performance on each image $\mathbf{x}_i$. This function can be the accuracy in the case of a classification task, the mIoU (mean Intersection over Union) score for semantic segmentation, or any other task-specific metric. Given the set of images $\mathcal{X}$, the average value of this measure over the entire dataset

$$\omega_\theta^{\text{avg}} = \frac{1}{|\mathcal{X}|} \sum_{\mathbf{x}_m \in \mathcal{X}} \omega_\theta(\mathbf{x}_m) \ , \tag{5}$$

is the overall performance of $\mathcal{M}_\theta$ on $\mathcal{X}$.

Given this value and a threshold β, we define the ground-truth sentence set $\mathcal{S}^*$, associated with failure cases of $\mathcal{M}_\theta$ on $\mathcal{X}$, as the set of sentences for which the mean performance across images associated with the sentence falls beneath the overall mean performance by a given margin β. To retrieve the images associated with a sentence, we define a binary function $\Gamma(\mathbf{x}_m, s_n)$ that outputs 1 if the sentence s_n describes the image $\mathbf{x}_m$ (*i.e.*, is relevant to it), and 0 otherwise. For example, if $\mathbf{x}_m$ is an image taken at night, $\Gamma(\mathbf{x}_m, \text{"}\textit{Image taken at night}'')= 1$ whereas $\Gamma(\mathbf{x}_m, \text{"}\textit{Image taken at sunset}'') = 0$. If this relevance cannot be derived directly from ground truth information (most of the cases), we resort to VQA [1] to obtain such information for arbitrary sentences on arbitrary datasets—as detailed in Appendix C.3.

The set of images associated with a sentence s_n is defined as

$$\mathcal{X}_{s_n} = \{\mathbf{x}_m \in \mathcal{X} | \Gamma(\mathbf{x}_m, s_n) = 1\} \ . \tag{6}$$

Then, for each sentence we compute a score by averaging the model performance over the associated images

$$\omega_\theta^{s_n} = \frac{1}{|\mathcal{X}_{s_n}|} \sum_{\mathbf{x}_m \in \mathcal{X}_{s_n}} \omega_\theta(\mathbf{x}_m) \ . \tag{7}$$

We use this quantity to represent the sentence's *hardness* score (according to the measure ω_θ) and we consider a sentence as *describing a failure mode* if its value is below the global average $\omega_\theta^{\text{avg}}$ by at least a margin β. Therefore, the desired output $\mathcal{S}^*$ is the collection of all of these sentences:

$$\mathcal{S}^* = \{s_n \in \mathcal{S} |\, \omega_\theta^{s_n} < \omega_\theta^{\text{avg}} - \beta\} \ . \tag{8}$$

Varying β allows to evaluate the methods with different severity levels for the desired error descriptions, *i.e.*, a higher β restricts the desired set $\mathcal{S}^*$ to more challenging sentences (see Fig. 10 in the main paper).

C.3 Pseudo-GT image-sentence relevance

When we do not have access to GT information, we rely on VQA [1] to determine whether an image $x_m \in \mathcal{X}$ can be associated with a sentence s_n (description) or not. These binary relevance scores $\Gamma(x_m, s_n)$ are necessary to define the GT sentence set $\mathcal{S}^*$ for LBEE (see Appendix C.2) but also to derive the sentence coverage ratio in the cluster (CR defined in Sec. 4 of the main paper).

In particular, we use OFA [39] and LLaVA [23] with their publicly available pre-trained weights—but any other (possibly more recent) model can be used instead. We do not simply use OFA and LLaVA alone, but we also combine them: combining the CLIP-based LLaVA with non CLIP-based OFA representation allow us to significantly decrease the CLIP bias from the pseudo-GT relevance scores.

To proceed with VQA, we first turn each sentence $s_n \in \mathcal{S}$ into a question q_n such that the expected answer is *yes* or *no*. For example the sentence "*An image taken at night*" was turned into the prompt: "*Was the image taken at night? Reply simply with 'yes' or 'no'.* ". Then we provide VQA model with the image-question pairs (x_i, q_n) and turn the yes/no answer into a binary score that is assigned to $\Gamma(x_i, s_n)$.

Comparing VQA pseudo-GT to manually annotated GT. Given the key role that VQA plays in our experimental validation, we carry out an experimental analysis to compare the pseudo-GT image-sentence associations obtained with LLaVA [23] and/or OFA [39] to manually annotated GT. To this end, we rely on the GT annotations from WD2 [44] and the derived 74 sentences (described in Appendix C.1) from which we compute the GT scores for $\Gamma(x_i, s_n)$. Turning

Table 4: Comparison of pseudo-ground truth annotations created using LLaVA and OFA with manual ground truth annotations in WD2.

Model	**Accuracy↑**	**TP↑**	**TN↑**	**FP↓**	**FN↓**
OFA	0.799	0.489	0.852	0.148	0.511
LLaVA	0.731	0.623	0.696	0.304	0.377
OFA | LLaVA	0.701	**0.735**	**0.642**	0.358	**0.265**
OFA & LLaVA	**0.829**	0.377	0.905	**0.095**	0.623

these sentences into questions, we can gather the pseudo-GT relevance scores $\Gamma_{\text{OFA}}(x_i, s_n)$ and $\Gamma_{\text{LLAVA}}(x_i, s_n)$.

We evaluate the accuracy of these pseudo-GT associations independently, as well as combined with AND/OR logical operations (**OFA & LLaVA** and **OFA | LLaVA**, respectively) and report our results in Tab. 4. Combining the output from both models yields the highest accuracy with respect to the ground-truth. Moreover, the accuracy of both models achieves an accuracy above 70% of the GT performance, with **LLAVA** hallucinating more (FP) and **OFA** missing more associations (FN). In light of this study, we use **OFA & LLaVA** to create pseudo-GT sentence-image relevance scores whenever GT information is not available. Note that the overhead of this computation does not affect the methods themselves, as it is used for evaluation only.

D Design choices and hyper-parameter selection

In this section we recall and detail our default parameter setting. We use OpenCLIP [14] trained on LAION-2B [32] as our default visual-textual embedding space, but one could use other VLM models, such as BLIP [21], *etc.* To generate the GT sentence set $\mathcal{S}^*_\beta$, we use $\beta = o * \omega_\theta^{\text{std}}$, where $\omega_\theta^{\text{std}}$ is the standard deviation of ω_θ computed over $\mathcal{X}$ with $o = 0.2$ as default value. We set the number of hard and easy clusters to $C = 15$ for large datasets (when we analyze the full set, namely urban scene segmentation and the unsupervised analyses on the NICO++ datasets) and to $C = 5$ for small sets, namely when we analyze the dataset for each class independently (per-class analyses of NICO++ and ImageNet). We set the number of retained sentences per cluster for all methods to $K = 3$.

In the unsupervised cases, we split $\mathcal{X}$ according to $t^h_\varphi = \varphi_\theta^{\text{avg}} + a * \varphi_\theta^{\text{std}}$ and $t^e_\varphi = \varphi_\theta^{\text{avg}} - a * \varphi_\theta^{\text{std}}$, where φ_θ is the model's output entropy and we set $a = 0.2$ as default value. In the per-class analysis using NICO++ datasets, we use class probabilities ρ_θ and split according to $t^h_\rho = \rho_\theta^{\text{avg}} - a * \rho_\theta^{\text{std}}$ and $t^e_\rho = \rho_\theta^{\text{avg}} - a * \rho_\theta^{\text{std}}$. For ImageNet, the splitting is done according to the ranking, with the hard set including both false positives (the GT class not ranked first) and false negatives (an image not labeled with the current class where the model ranked the current class as first)—as described in Appendix A

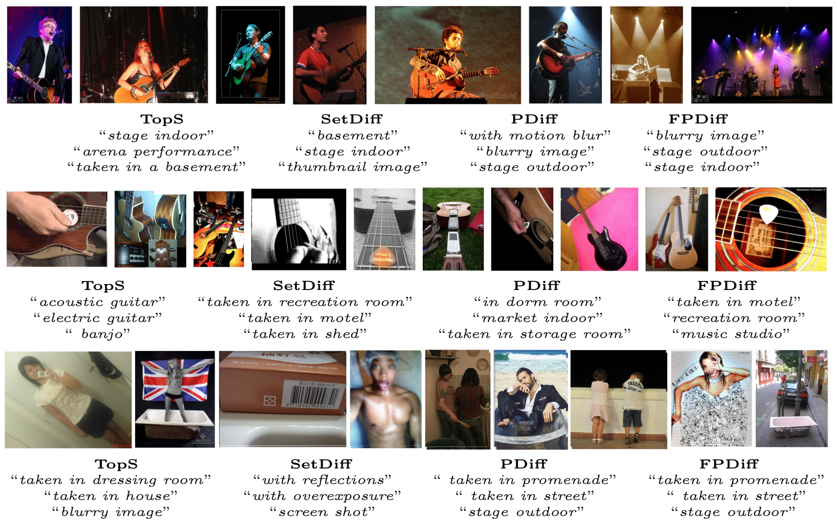

Fig. 11: Analyzing EfficientNet performance on the ImageNet 1K. The first two rows are from the ‘*acoustic guitar*” class, and the third is from the ‘*bath tube*” class.

E In-depth analyses of ImageNet 1K

Here, we focus on ImageNet 1K, where we treat each class separately. We can make several observations. *i)* When the error slices (hard clusters) mainly contain false positives from semantically similar classes or false negatives with shared content, **TopS** will generally capture the shared characteristics—see examples in Fig. 11 (top two rows), Fig. 13 and Fig. 14. *ii)* In the case of false positives with semantically similar classes, *e.g.*“*electric guitar*” and “*banjo*” in place of the “*acoustic guitar*” class (see first row in Fig. 11), the main difficulty for the other methods is that the same sentences are also highly ranked for the easy clusters, making it difficult for them to correctly identify the desired explanations—namely, contrasting the explanation between the easy and the hard cluster. This limitation is inherited from CLIP, which has difficulties to distinguish fine-grain classes. *iii)* When the false negatives share some semantic similarities and there is a candidate sentence that well describes this, all methods are able to find it. For example, see in Fig. 11 (second row) the difficulty of the model to label the “*acoustic guitar*” when the photo is showing the person holding the guitar on a “*stage*”. Note that for these images **PDiff** and **FPDiff** also identified the *blur* as potential challenge (or “*low contrast*” and “*dim weather*” condition, in Fig. 13). *iv)* Finally, we have clusters where even for a human is difficult to describe what is common in the images beyond the object class. When there is a mix between false positives and false negatives, all methods including **TopS** have difficulties to identify the failure reason. In these cases, the selected sentences are relevant to some of the images in the cluster but they do not point to the reason for the failure (see Fig. 11 last row).

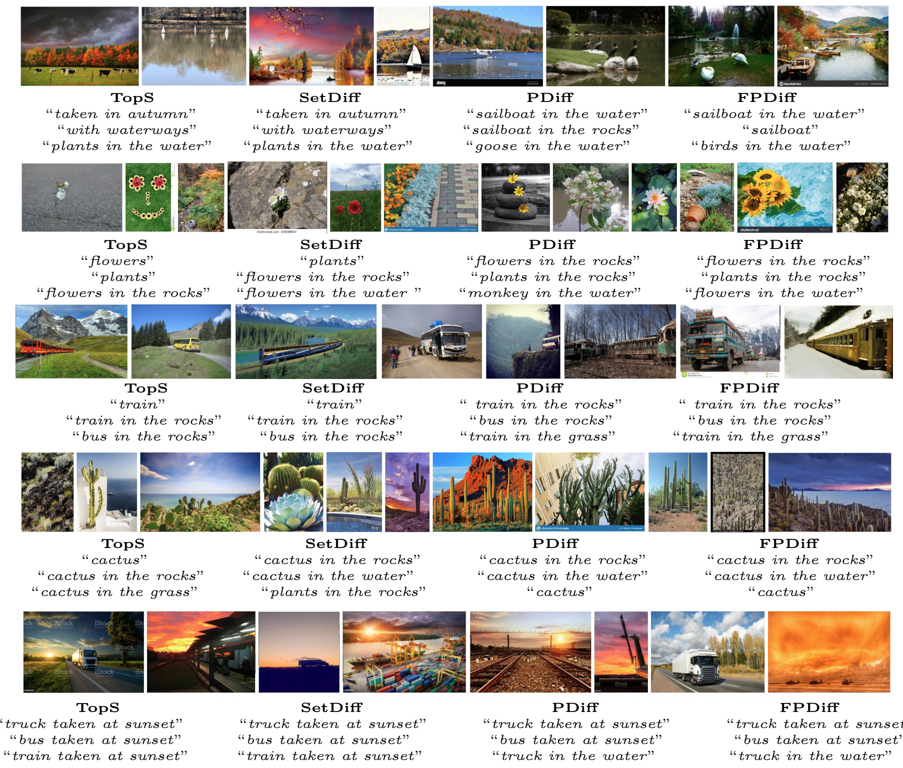

Fig. 12: Five out of 15 hard clusters from NICO++ 85 (unsupervised case) explained by different methods.

F Further qualitative analyses

We show further qualitative examples for NICO++ 85 in Fig. 12, for ImageNet 1K using the pre-trained Vit-B16 model in Figs. 13 and 14 and finally further examples obtained for the urban scene segmentation model on WD2 in Fig. 15. We conclude the section with Tab. 5 which is an extended version of Tab. 1 in the main paper.

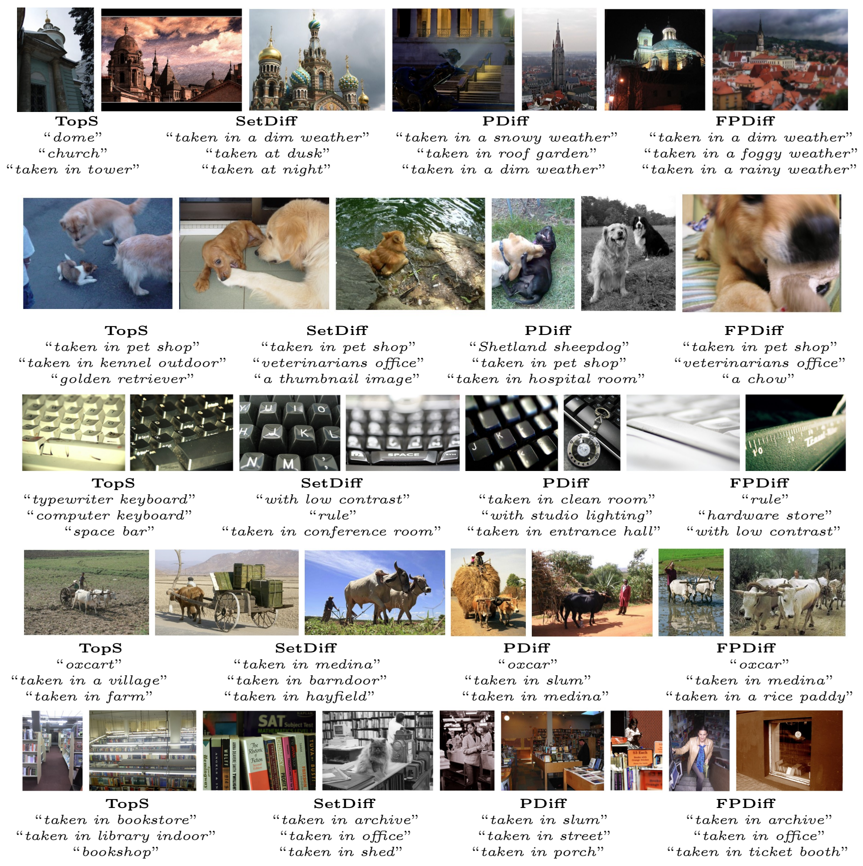

Fig. 13: Hard clusters from ImageNet 1K validation set tested with the pre-trained Vit-B16 model. Here we show examples containing mainly false negative (not recognized) images from the correct class. From top to bottom we have the classes “*church*”, “*golden retriever*”, “*computer keyboard*”, “*ox*” and “*bookshop*”.

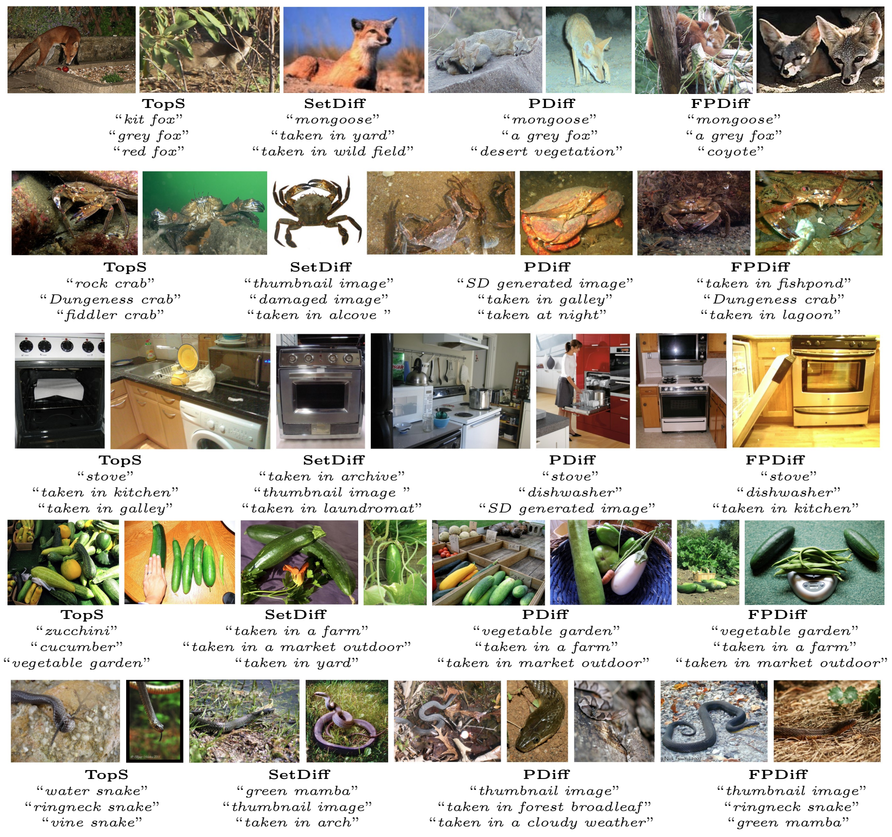

Fig. 14: Hard clusters from ImageNet 1K validation set tested with the pre-trained Vit-B16 model. Here we show examples containing mainly false positives (images from other classes) images from the correct class. From top to bottom we have the classes "*red fox*", "*rock crab*", "*microwave*", "*cucumber*" and "*water snake*".

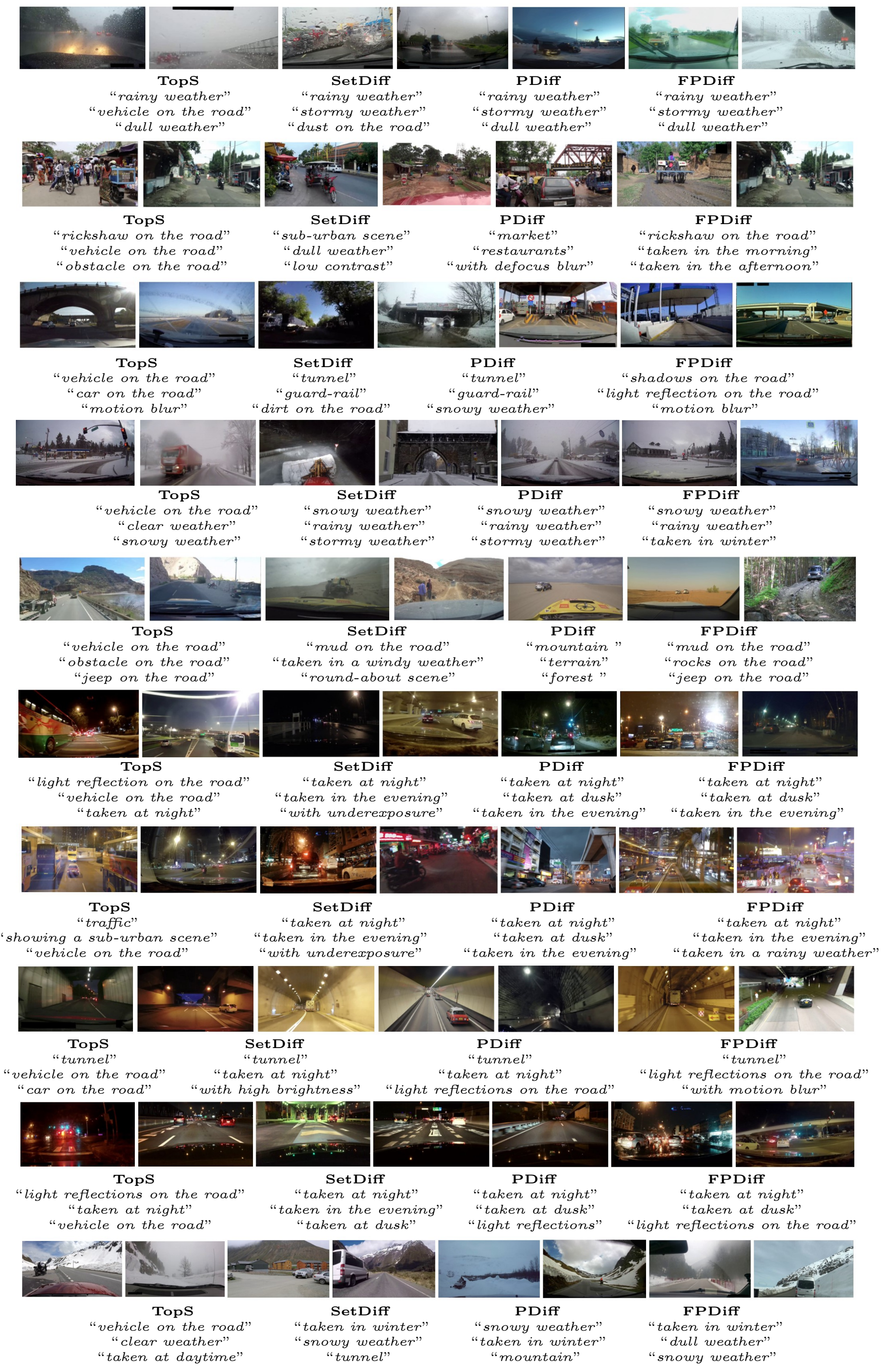

Fig. 15: Ten out of 15 hard clusters from WD2 explained by different methods.

Table 5: Example sentences from the User-defined sentence set $\mathcal{S}$. For each sentence we show if it belongs to $\mathcal{S}^*_\beta$ (obtained with $o = .2$) and to $\mathcal{R}_\mathcal{S}$ for **TopS** (**TS**), **PDiff** (**PD**), **SetDiff** (**SD**) and **FPDiff** (**FP**).For $\mathcal{S}^*_\beta$, "✓" means $s_n \in \mathcal{S}^*_\beta$ and "×" means $s_n \notin \mathcal{S}^*_\beta$—namely, for that dataset the hardness score of s_n is below the required level. For the methods, "+"/"+" indicate that the sentence is in $\mathcal{R}_\mathcal{S}$, where "+" means true positive (*i.e.*, the sentence is also in $\mathcal{S}^*_\beta$) while "+" means false positive. An empty space indicates that the sentence was not in the corresponding $\mathcal{R}_\mathcal{S}$. Note that the table is not complete as we omitted to show a few sentences that were not in any $\mathcal{S}^*_\beta$, semantically equivalent to another sentences with similar behaviour or retained by a single method for one dataset.

	WD2					IDD					ACDC				
Sentences (*An image ...*)	$\mathcal{S}^*$	**TS**	**PD**	**SD**	**FP**	$\mathcal{S}^*$	**TS**	**PD**	**SD**	**FP**	$\mathcal{S}^*$	**TS**	**PD**	**SD**	**FP**
"taken at night"	✓	+	+	+	+	✓		+	+	+	✓	+			
"taken in the evening"	✓		+	+	+	✓		+		+	✓	+			
"taken at dusk"	✓		+	+	+	✓		+	+	+	✓				
"taken at dawn"	×		+			✓		+	+		×				
"taken in a stormy weather"	✓		+	+	+	×					×				+
"taken in a foggy weather"	✓			+		✓					×	+	+	+	+
"taken in a rainy weather"	✓	+	+		+	✓					×	+	+		+
"taken in a snowy weather"	✓		+	+	+	×					×	+			
"taken in a windy weather"	✓	+	+	+	+	×					×		+		+
"taken in a dull weather"	✓	+	+	+	+	×					×	+	+	+	+
"taken in winter"	✓			+	+	×					×				+
"taken in autumn"	×		+			✓					×				
"with reflection on the road"	✓	+	+	+	+	✓			+		✓				+
"with shadows on the road"	×				+	×			+		✓			+	
"with water on the road"	✓	+				×					×			+	+
"with mud on the road"	×			+	+	×		+	+		×				
"with branches on the road"	✓					✓			+	+	×		+	+	+
"with traffic cone on the road"	✓					✓					✓		+	+	+
"with obstacle on the road"	✓	+			+	×	+				×				
"with road barrier on the road"	×			+		×			+	+	✓	+	+	+	+
"with rail track on the road"	×					✓			+	+	✓	+	+	+	+
"with rocks on the road"	×			+	+	×					×				
"showing a highway scene"	×			+		×			+	+	×	+	+	+	+
"showing an industrial scene"	✓					✓					×				
"showing construction site"	×					✓		+	+	+	×		+	+	
"showing sub-urban scene"	×	+		+		×	+			+	✓				
"with rickshaw on the road"	×	+	+	+	+	×	+	+		+	✓		+		
"with motorbike on the road"	×		+		+	×				+	✓				+
"with bike on the road"	×					×			+	+	×			+	
"with vehicle on the road"	×	+				×	+				×	+	+	+	+
"with bus on the road"	×					×	+	+		+	×			+	
"with tram on the road"	×					×			+		×	+	+		+
"with jeep on the road"	×	+		+	+	×					×				
"with animal on the road"	×			+	+	✓			+	+	×				
"with bird on the road"	×					✓					×				
"with people on the road"	×	+			+	×	+			+	×		+		
"with crowded background"	×					✓			+		×				
"with crowded foreground"	×					×		+	+	+	×				
"with motion blur"	✓	+	+		+	✓			+	+	✓		+		
"with overexposure"	✓					×					×				
"with underexposure"	×			+	+	×					✓			+	
"with low contrast"	✓					×			+	+	×			+	
"of a tunnel"	✓	+	+	+	+	✓		+			✓	+	+	+	+
"of fences"	×					×		+			✓		+		
"of guard-rail"	×					×					×		+	+	+
"of a traffic jam"	✓					✓	+	+		+	×				
"of a traffic"	×	+				×	+	+		+	×		+		
"of traffic lights"	✓					✓					×		+		
"of buildings"	×		+			×		+			×		+		
"of a parking"	×		+			✓		+	+	+	×		+		
"of a mountain"	×	+	+			×					×				