Evolving Interpretable Visual Classifiers with Large Language Models

Mia Chiquier, Utkarsh Mall, and Carl Vondrick

Columbia University
{mia.chiquier,utkarshm,vondrick}@cs.columbia.edu

Abstract. Multimodal pre-trained models, such as CLIP, are popular for zero-shot classification due to their open-vocabulary flexibility and high performance. However, vision-language models, which compute similarity scores between images and class labels, are largely black-box, with limited interpretability, risk for bias, and inability to discover new visual concepts not written down. Moreover, in practical settings, the vocabulary for class names and attributes of specialized concepts will not be known, preventing these methods from performing well on images uncommon in large-scale vision-language datasets. To address these limitations, we present a novel method that discovers interpretable yet discriminative sets of attributes for visual recognition. We introduce an evolutionary search algorithm that uses the in-context learning abilities of large language models to iteratively mutate a concept bottleneck of attributes for classification. Our method produces state-of-the-art, interpretable fine-grained classifiers. We outperform the baselines by 18.4% on five fine-grained iNaturalist datasets and by 22.2% on two KikiBouba datasets, despite the baselines having access to privileged information.

Keywords: Visual Recognition, Interpretable Representations

1 Introduction

Multimodal foundation models like CLIP [1] obtain excellent performance on many visual recognition tasks due to their flexibility to represent open-vocabulary classes. These models have the potential to impact many scientific applications, where computer vision systems could automate recognition in specialized domains. However, since foundation models are neural networks, they are largely black-box and we therefore have no means to explain or audit the predictions they produce, limiting their trust. Moreover, given that foundation models are trained on large corpora of web-scraped data [1, 2], they are not optimized to represent rare and fine-grained concepts, such as images from various scientific fields, which are infrequently described on the internet.

The computer vision community has been building interpretable models by integrating language, where classifiers are constructed with a bottleneck of sparse or discrete attributes [3–7]. Language-based approaches have the benefit of being interpretable. However, they rely on attributes that are either hand-designed,

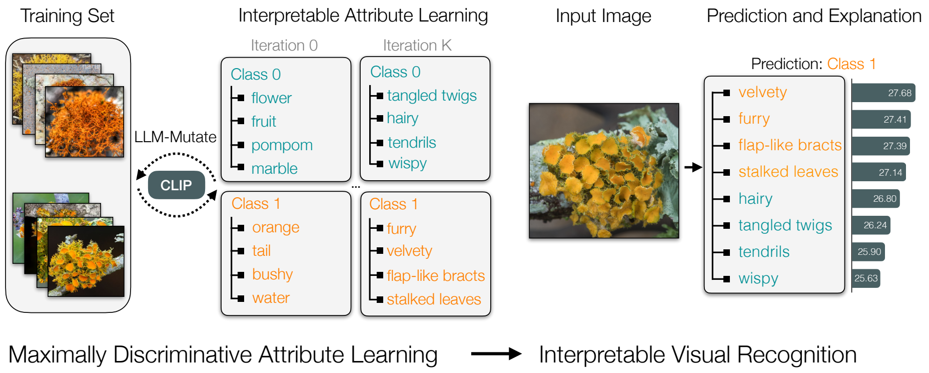

Fig. 1: Learning Interpretable Classifiers. Can we find text attributes for a concept by looking at the images without their class names? LLM-Mutate is a framework that learns sets of maximally discriminative visual attributes per class without access to class names or any form of prior knowledge.

requiring expert knowledge, or extracted from external sources, such as LLMs. Since the attributes are not learned, they often obtain poor performance on specialized classes infrequently discussed in web-scale training sets.

In this paper, we propose a framework to learn interpretable visual representations from images. Our approach can discover discrete, discriminative attributes from purely visual training sets of specialized concepts, and even unwritten concepts that do not appear on the internet. Solving this problem has typically been challenging because the loss is not differentiable with respect to discrete attributes, limiting the application of modern deep learning methods. Gradient-free methods, such as evolutionary search [8], are able to optimize over discrete search spaces, but they have not scaled to large problems in the past.

Our approach integrates LLMs with evolutionary algorithms in order to learn discriminative and interpretable attributes for visual recognition. Evolutionary algorithms work by maintaining a set of candidates, randomly mutating them, and discarding the poorly performing ones according to an objective function. The mutation step is a bottleneck for large-scale problems, such as object recognition, as the search space is large and the lack of gradient information means the mutations are not guaranteed to drive the optimization towards convergence.

We overcome these bottlenecks by replacing the mutation step with an LLM instead, whose in-context learning abilities are able to find patterns in-between candidates and predict strong mutations that reduce the objective. We evaluate our method on images from specialized scientific domain that have been infrequently discussed on the internet due to the nature of their specificity, with the iNaturalist dataset [9]. We chose five families of plants and animals within iNaturalist, each containing five to six species, and evaluated the fine-grained classification performance on each family. We outperform all baselines, on average by 18.4% per family dataset. We also evaluated our method on imaginary concepts that have been hardly discussed on the internet before. Following the

KikiBouba experiments [10] where people associate imaginary objects with non-existent words, we learn interpretable attributes that achieve strong discriminative performance, outperforming baselines by an average of 22.2%.

Our primary contribution is a framework for learning interpretable visual recognition systems from images. We propose to integrate evolutionary search with LLMs, allowing us to efficiently learn interpretable, discrete, and discriminative attributes. The remainder of the paper will discuss the related work in this area, describe the method, and present experimental results on multiple datasets. Our code, data, and models are available at https://llm-mutate.cs.columbia.edu.

2 Related Work

Concept Bottlenecks. A common approach for interpretability is to create a bottleneck in the classifier, where interpretable attributes are first predicted before classifying the object category [3,4,6,11–13]. Concept bottleneck models have been comprehensively studied in the domain of zero-shot learning [14–18]. However, all these methods require a significant amount of annotations for the intermediate attributes to perform well while being interpretable. Our method instead finds interpretable intermediate concepts by learning from visual data.

Post-hoc Interpretabililty. Several post-hoc methods have been proposed to interpret deep models. Methods such as GradCAM [19] rely on the activation maps to provide explanations for classification [20–25]. Generating counterfactual examples is another way for interpretation [26–29]. Other work has looked at sample importance for explanations [30–36]. All these methods look at explanations from the perspective of a single image. In contrast to these approaches, several methods aim at understanding the behaviors of individual neurons in trained models [37–41]. One major limitation of post-hoc methods is that they can only explain the receptive field of the model, which requires some interpretation on the user's part. Concept or attribute bottleneck models on the other hand can justify the classification by providing scores for individual attributes.

Vision-Language Models. Large-scale vision-language models (VLMs), such as CLIP [1], have bridged the visual and language modalities through contrastive learning. VLMs estimate the similarity between text and an image, leading to many downstream applications, such as zero-shot classification with open-ended language. Several improvements have been proposed to the VLMs, such as training with noisy data [42–44], better training strategies [45–52], concept localization [53], multiple modalities [54], grounding abilities [55,56], or generative capabilities [57–60]. Another advantage of VLMs is that classification or retrieval can be done by using a list of attribute descriptors in conjunction with the class names [5,61,62]. Classification by using descriptions results in an intermediate step of interpretability, as these lists of descriptions are essentially concept bottlenecks. Waffle-CLIP [7] proposed using random words along with these concepts leads to more robust concept embedding. However, all these methods have access to a knowledge base (large language models in this case), which they use to retrieve descriptions for categories using the category name.

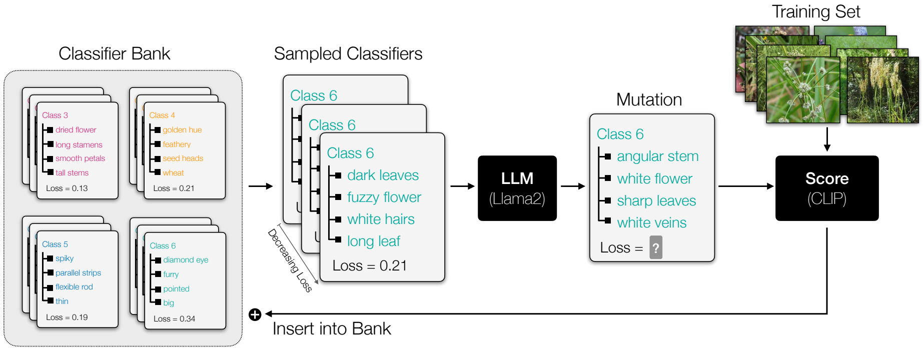

Fig. 2: Method. LLM-Mutate is an evolutionary algorithm that learns sets of discrete language attributes per class. The mutation and cross-over operations– that introduce new parameter hypotheses– are replaced by a large language model that uses in-context learning over past attributes and their scores to iteratively generate better attributes.

Such approaches fail if the categories are never discussed or are infrequently mentioned on the internet. Since our approach does not rely on the name of the category, we perform better on specialized and esoteric categories.

LLMs as Optimizers. Using a large language model to generate mutations with evolutionary search has been implemented for the task of code generation [63]. We follow a similar approach, except for the task of visual recognition. In addition, their fitness functions are a series of non-learning-based metrics. We instead use another foundation model, a vision-language model, as the fitness function, thereby scaling to open-ended visual concepts. Another instance in which LLM's are used as optimizers is for prompt discovery [64]. Similarly, concurrent work leverages LLMs as optimizers to find attributes to improve visual classification, but crucially, they provide the class name as an input to the LLM [65, 66]. As such, they use privileged information that we do not assume.

3 Discovering Visual Classifiers

3.1 Model

Given an image x, our goal is to predict its category label y. We create a concept bottleneck model $f_c(x)$ that scores whether the image x contains category c. Once optimized, the concept bottleneck model should produce a high score for $c = y$ and a low score otherwise. To score the class c from the image x, we average over a bottleneck of discrete attributes:

$$f_c(x; \mathcal{D}) = \frac{1}{|\mathcal{D}(c)|} \sum_{d_i \in \mathcal{D}(c)} \phi(d_i, x) \tag{1}$$

where $\mathcal{D}(c)$ is the set of attributes to recognize class c, and $\phi(d_i, x)$ is the score from a vision-language model (such as CLIP) to detect the attribute d_i in image

x. The model in Eq. (1) provides a degree of explainability because a prediction must be based in natural language attributes $d_i \in \mathcal{D}(c)$. At inference, we perform multi-class classification by scoring Eq. (1) for each class, and picking the highest scoring one, i.e. $\arg\max_c f_c(x; \mathcal{D})$.

Concept bottleneck models have traditionally been challenging to implement in computer vision because we need to instantiate a discriminative set of discrete attributes $\mathcal{D}(c)$, which pose challenges for gradient-based optimization methods that are now ubiquitous in deep learning. Prior work has relied on manual annotation of $\mathcal{D}(c)$ for each class [6], which does not efficiently scale, or relied on other knowledge bases [5], which cannot generalize to specialized categories.

3.2 Learning and Optimization

Given C categories (without semantic labels), we want to learn discriminative attributes $\mathcal{D}$ where $\mathcal{D}(c)$ are the attributes for class c. We optimize the objective:

$$\min_{\mathcal{D}} \ \mathbb{E}_{(x,y)} \left[\mathcal{L}\left(\hat{y}, y\right) \right] \quad \text{for} \quad \hat{y} = \left[f_1(x; \mathcal{D}), \ldots, f_C(x; \mathcal{D}) \right], \tag{2}$$

where $\mathcal{L}$ is a loss function that measures the error of the predictions $\hat{y}$ to the label y of each training example. We use the cross-entropy loss for $\mathcal{L}$.

Since attributes for each class are discrete and f_c is not differentiable, we optimize Eq. (2) with evolutionary search. We maintain a bank of hypotheses B for potential $\mathcal{D}(c)$, mutate them, and keep the best-scoring hypotheses according to the objective function. Typically, evolutionary search creates heuristics to mutate the attributes $\mathcal{D}(c) \in B$, for example by randomly merging attributes together (called crossover) or by randomly injecting new words from a vocabulary. However, these heuristics are not efficient for two reasons. First, the search space of natural language descriptions is large. Second, most heuristics do not leverage patterns between attributes and their performance that could drive the optimization to convergence rapidly.

We propose to replace the mutation step with a large language model and its in-context learning capabilities. Given k past hypotheses $\{\mathcal{D}_t(c), \ldots, \mathcal{D}_{t-k}(c)\}$ for a class c, and their loss, we "mutate" the next hypotheses through:

$$\hat{\mathcal{D}}_{t+1}(c) = \text{LLM}\left(\mathcal{D}_t(c), \ldots, \mathcal{D}_{t-k}(c)\right) \tag{3}$$

where $\mathcal{D}_t$ is an attribute set from iteration t. By using an LLM, the mutated $\hat{\mathcal{D}}_{t+1}(c)$ benefit from natural language priors, allowing us to efficiently search for descriptors that obey the semantics and syntax of natural language. Secondly, the in-context learning ability of the LLM means they will be able to find patterns in the past hypothesis to guide the search towards attributes that are likely to minimize the objective function. After mutation, we add $\hat{\mathcal{D}}_{t+1}(c)$ to the bank of classifiers if it improves the objective and iterate.

We use the open-source Llama-2-70B-Instruct [67] for the LLM$(\cdot)$ and CLIP ViT-B/32 [1] as our vision-language model $\phi(\cdot)$. The starting attributes are initialized randomly, with no prior knowledge of the class name or prior information.

Algorithm 1: Discriminative Attribute Set Learning

```
Input: Number of categories $C$, Loss function that scores attribute set $\mathcal{L}(\mathcal{D})$
       on the training set, Scalar hyper-parameters $N$, $M$
Output: Discriminative attributes $\mathcal{D}_{\text{best}}$, where $\mathcal{D}_{\text{best}}(c)$ is the set of
        attributes to recognize class $c$
// Randomly initialize the classifier bank $B$
$B \leftarrow \{\}$
for $i = 1, \ldots, N$ do
    for $c = 1, \ldots, C$ do
        $\mathcal{D}_i(y) \leftarrow$ randomly sampled attributes
    $B \leftarrow B \cup \{\mathcal{D}_i\}$
// Learn a $\mathcal{D}$ that discriminates the training set $\{(x, y)\}$
while not converged do
    // Biased sampling of $M$ classifiers by expected loss $\mathbb{E}[\mathcal{L}]$
    $S = \{\mathcal{D}_1, ..., \mathcal{D}_M\}$ where $\mathcal{D}_i \sim B$ s.t. $p(\mathcal{D}_i) \propto \mathbb{E}[\mathcal{L}(\mathcal{D}_i)]$
    // Sort by the expected loss
    $S = (\mathcal{D}_1, ..., \mathcal{D}_M)$ where $\mathbb{E}[\mathcal{L}(\mathcal{D}_i)] \geq \mathbb{E}[\mathcal{L}(\mathcal{D}_j)] \quad \forall_{i<j}$
    // One step of evolutionary search for each category
    for $c = 1, \ldots, C$ do
        // Mutate the attributes with the LLM
        $S_c = (\mathcal{D}_i(c) : \mathcal{D}_i \in S)$
        $\hat{\mathcal{D}}(c) \leftarrow \text{LLM}(S_c)$
        $S' = (\mathcal{D}'_1, \ldots, \mathcal{D}'_M)$ where $\forall_k \; \mathcal{D}'_i(k) = \begin{cases} \hat{\mathcal{D}}(c) & \text{if } c = k \\ \mathcal{D}(k) & \text{otherwise} \end{cases}$
        // Keep best attribute set
        $B \leftarrow B \cup \arg\min_{\mathcal{D} \in S'} \{\mathbb{E}[\mathcal{L}(\mathcal{D})]\}$
return $\arg\min_{\mathcal{D} \in B} \{\mathbb{E}[\mathcal{L}(\mathcal{D})]\}$
```

3.3 Evolutionary Search

Algorithm 1 shows the search procedure to optimize the attribute sets for classification. We initialize the classifier bank B with random words to create N initial hypotheses for $\mathcal{D}_i$. We sample sets of attributes from B to construct the in-context examples for mutation and repeat this process until convergence. We bias the samples with the loss of each attribute set. For each sampled set, we construct a prompt per class by concatenating the attributes from class c in increasing order of performance. We prompt the LLM separately per class, which generates the novel, mutated attributes, $\hat{\mathcal{D}}(c)$ for them. We evaluate the newly mutated attribute sets and add the best-mutated classifier to the classifier bank. We provide more implementation details in the supplementary material.

3.4 Classifier Bank Initialization

During fine-grained classification, the optimization favors attributes that are highly discriminative between visually similar classes, causing attributes common to multiple classes to be discouraged. We want to encourage common attributes at the beginning of learning so that the optimization discovers class-specific details, instead of spurious unrelated attributes arising from noise.

To achieve this, we use a pre-training strategy that discovers common attributes, to serve as an initialization for joint multi-class training. We implement the pre-training step by learning a per class binary classifier, using the same algorithm as before, but with an objective to separate one class from all others, including significantly unrelated classes. We use the following objective:

$$\min_{\mathcal{D}} \mathbb{E}_{x_p}\left[f_c\left(x_n;\mathcal{D}\right)\right] - \mathbb{E}_{x_n}\left[f_c\left(x_p;\mathcal{D}\right)\right] \tag{4}$$

where x_p is a positive images of the class c, and x_n denotes the negative images of all the other classes. Crucially, we include classes outside of the fine-grained dataset in the negative classes here as well. The generated attributes across the first 200 iterations become the initialization of the set of attributes for the program bank initialization of the joint training. We randomly initialize the attribute bank for binary-pre-training with a large pool of attributes generated by an LLM about generic visual categories. Fig. 3 shows an example of the initialization, and we include further implementation details in the supplementary.

4 Experiments

4.1 Datasets

We validate our method by evaluating on two datasets. First are subsets of iNaturalist – a fine-grained classification dataset with rare species that are rarely discussed online. Second is an image dataset of novel concepts, created with new words that are even less likely to be discussed online, as they are invented.

iNaturalist: iNaturalist [9] is a dataset for fine-grained species classification. It contains images and annotations obtained from citizen scientists for a large number of animal, plant, and fungus species. We experiment with five different families and classify between five to six species with each family. We chose families whose features varied in more complex ways than color.

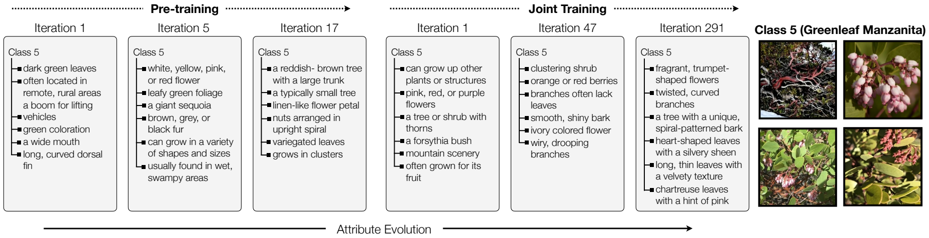
Pre-training
Joint Training
Iteration 1
Class 5
dark green leaves
often located in remote, rural areas
a boom for lifting
vehicles
green coloration
a wide mouth
long, curved dorsal fin
Iteration 5
Class 5
white, yellow, pink, or red flower
leafy green foliage
a giant sequoia
brown, grey, or black fur
can grow in a variety of shapes and sizes
usually found in wet, swampy areas
Iteration 17
Class 5
a reddish- brown tree with a large trunk
a typically small tree
linen-like flower petal
nuts arranged in upright spiral
variegated leaves
grows in clusters
Iteration 1
Class 5
can grow up other plants or structures
pink, red, or purple flowers
a tree or shrub with thorns
a forsythia bush
mountain scenery
often grown for its fruit
Iteration 47
Class 5
clustering shrub
orange or red berries
branches often lack leaves
smooth, shiny bark
ivory colored flower
wiry, drooping branches
Iteration 291
Class 5
fragrant, trumpet-shaped flowers
twisted, curved branches
a tree with a unique, spiral-patterned bark
heart-shaped leaves with a silvery sheen
long, thin leaves with a velvety texture
chartreuse leaves with a hint of pink
Class 5 (Greenleaf Manzanita)
Attribute Evolution

Fig. 3: Attribute evolution. We show examples of the attribute evolution for both the pre-training and joint-training stages of learning. At the beginning, the first generated set of attributes have little to do with the class, and by the end of the joint-training, the learned attributes are specific to the Greenleaf Manzanita.

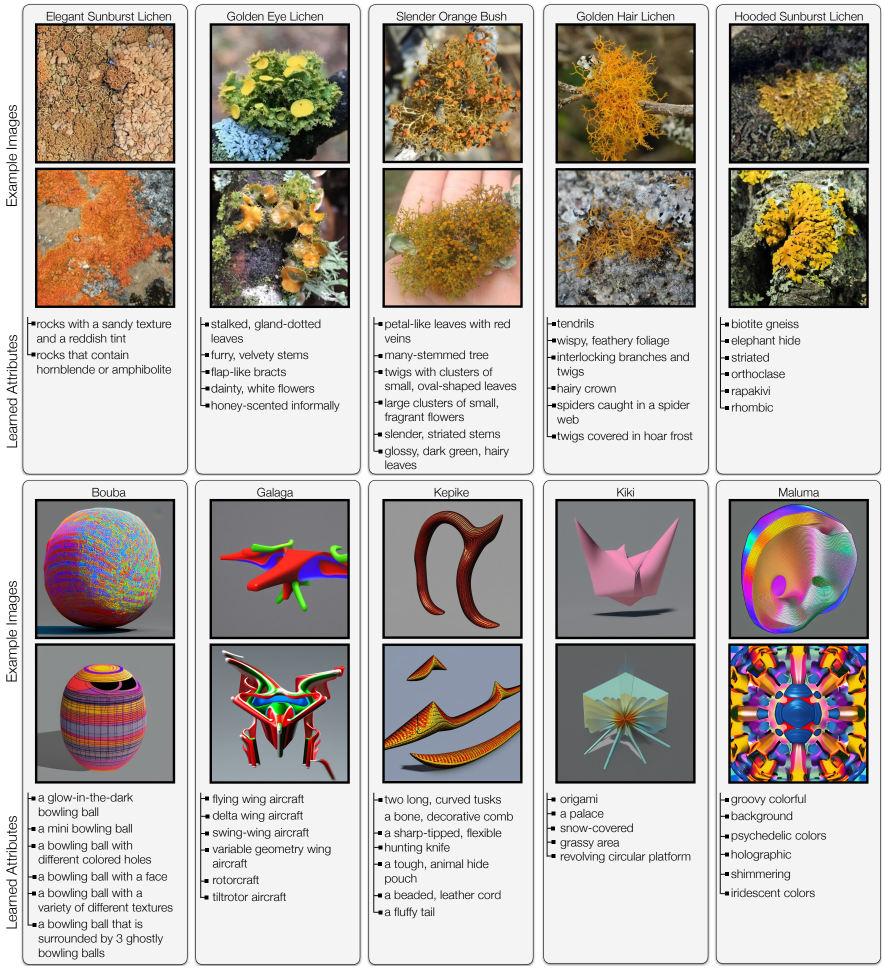

Fig. 4: Qualitative Results. We show qualitative results for the iNaturalist Lichen family and a KikiBouba dataset. The results illustrate two sample images per class and the learned attributes. The learned attributes for the Lichen hardly refer to color, as this is a common feature to all Lichen, and instead focus on structural properties.

Lichen (fungi) has 6 species: elegant sunburst, golden-eye, slender orange-bush, golden-hair, hooded sunburst, and maritime sunburst lichen. Wrasse (fish) has 5 species: caribbean bluehead, six-bar, cortez-rainbow, moon, and ornate wrasse. Wild rye (grass) has 5 species: squirreltail, bottlebrush, quack grass, canada wild rye, and virginia wild rye. Manzanita (berry shrubs) has 5 species: big-berry, pinemat, greenleaf, point-leaf, and pine-mat manzanita. Bulrush (herb) also has has 5 species: dark-green, woolgrass, panicled, rufous, and wood bulrush.

Kiki-Bouba: We also validate our method on completely novel concepts that do not appear on the internet and thus language models lack familiarity.

In a surprising study, the Kiki-Bouba experiment [68] showed that people tend to associate specific symbols with different sounds, even though the words and the physical objects do not exist. We create a dataset corresponding to such concepts with generative models trained to create images from such words [10]. Such a dataset with novel concepts is a strong testbed for attribute discovery.

4.2 Baselines

We compare our method against several baselines. Every baseline that is not our own has access to privileged information that we do not. Specifically, the baselines with zero-shot attributes were generated by prompting GPT3 [69] with the class name. The baselines that contain class names have an evident advantage. Lastly, the gradient-based approach we constructed does not have access to privileged information such as class names. Nonetheless, our evolutionary method significantly outperforms this baseline for every dataset, thus justifying our evolutionary approach. Our results show that our method outperforms all of the baselines across all datasets, demonstrating that it can learn attributes for specialized and undocumented visual categories.

Class Name (CLIP [1]): Our first baseline is the simple method of classifying with the class name. For the iNaturalist dataset, which has both a common and a scientific name per species, we report the best accuracy between using the common name, the scientific name, and both names.

Classification by Description [5]: We implement the Classification by Description method [5], which generates zero-shot attributes for a class by prompting GPT3 [69], and joins the class name to each of the zero-shot attributes. Similar to the "Class Name" baseline, for the iNaturalist dataset, we report the best accuracy between using the common, the scientific, and both names.

Zero-shot Attributes: We also use a variant of Classification by Description where the class name is no longer appended to the attributes.

Gradient-based Approach: Instead of using an LLM to search for attributes, we instead search for optimal input tokens to the text encoder of the VLM using gradient descent. For a class, we find tokens that highly discriminate it from other classes. The model optimizes for probability values over the complete token list. However, since the explanations have to be tokens and not a probability distribution over them, we enforce the probability distribution to be more selective to fewer tokens as training progresses by using a temperature parameter that decreases over the training period.

Varying Prompt Length: We report the results of our method with two different prompt lengths: one and ten. In the former, the LLM prompt only has one example of a set of attributes for a particular class, and therefore doesn't see the increasingly better sets of attributes. We notice a significant drop in accuracy when in-context learning is prohibited.

Engineered Text Templates: For each method, we report the best accuracy per dataset between the accuracy computed using scores averaged over the engineered text templates proposed in CLIP [1], and without averaging over the

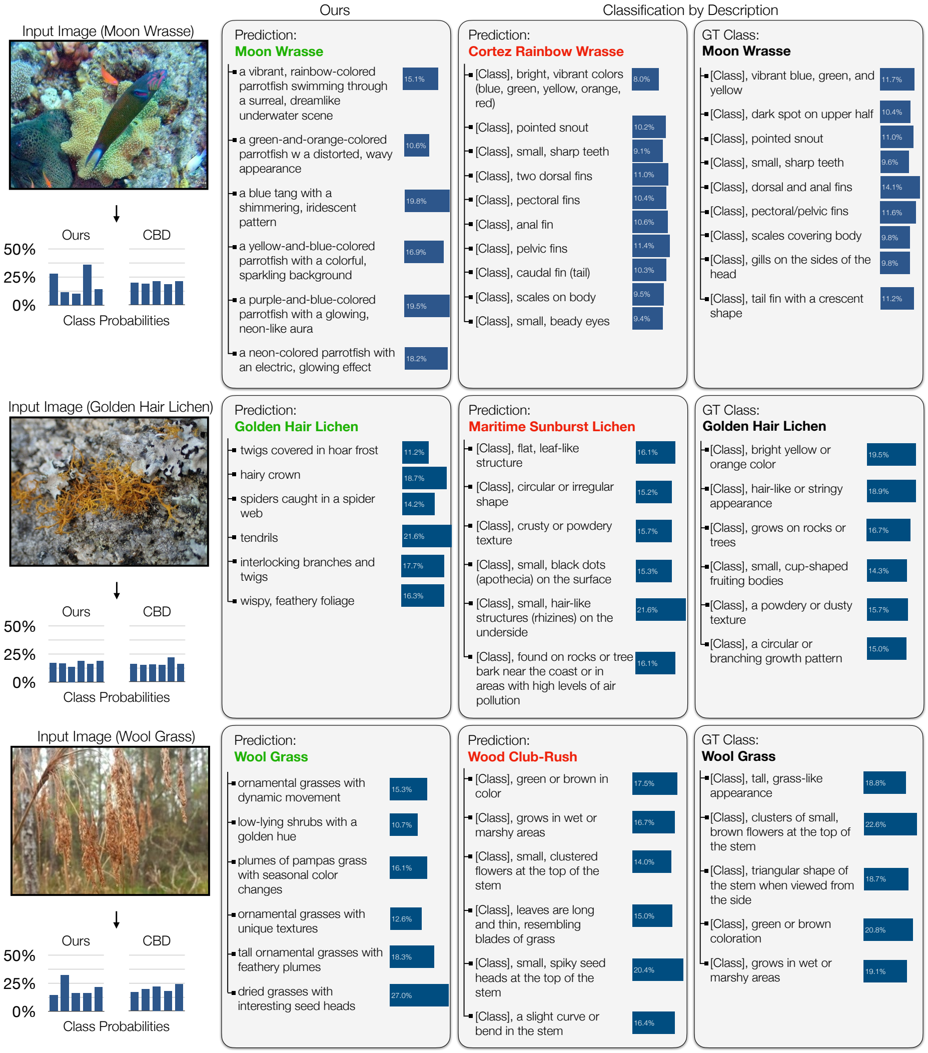
Ours
Classification by Description
Input Image (Moon Wrasse)
Ours
CBD
50%
25%
0%
Class Probabilities
Prediction:
Moon Wrasse
a vibrant, rainbow-colored parrotfish swimming through a surreal, dreamlike underwater scene 15.1%
a green-and-orange-colored parrotfish w a distorted, wavy appearance 10.6%
a blue tang with a shimmering, iridescent pattern 19.8%
a yellow-and-blue-colored parrotfish with a colorful, sparkling background 16.9%
a purple-and-blue-colored parrotfish with a glowing, neon-like aura 19.5%
a neon-colored parrotfish with an electric, glowing effect 18.2%
Prediction:
Cortez Rainbow Wrasse
[Class], bright, vibrant colors (blue, green, yellow, orange, red) 8.0%
[Class], pointed snout 10.2%
[Class], small, sharp teeth 9.1%
[Class], two dorsal fins 11.0%
[Class], pectoral fins 10.4%
[Class], anal fin 10.6%
[Class], pelvic fins 11.4%
[Class], caudal fin (tail) 10.3%
[Class], scales on body 9.5%
[Class], small, beady eyes 9.4%
GT Class:
Moon Wrasse
[Class], vibrant blue, green, and yellow 11.7%
[Class], dark spot on upper half 10.4%
[Class], pointed snout 11.0%
[Class], small, sharp teeth 9.6%
[Class], dorsal and anal fins 14.1%
[Class], pectoral/pelvic fins 11.6%
[Class], scales covering body 9.8%
[Class], gills on the sides of the head 9.8%
[Class], tail fin with a crescent shape 11.2%
Input Image (Golden Hair Lichen)
Ours
CBD
50%
25%
0%
Class Probabilities
Prediction:
Golden Hair Lichen
twigs covered in hoar frost 11.2%
hairy crown 18.7%
spiders caught in a spider web 14.2%
tendrils 21.6%
interlocking branches and twigs 17.7%
wispy, feathery foliage 16.3%
Prediction:
Maritime Sunburst Lichen
[Class], flat, leaf-like structure 16.1%
[Class], circular or irregular shape 15.2%
[Class], crusty or powdery texture 15.7%
[Class], small, black dots (apothecia) on the surface 15.3%
[Class], small, hair-like structures (rhizines) on the underside 21.6%
[Class], found on rocks or tree bark near the coast or in areas with high levels of air pollution 16.1%
GT Class:
Golden Hair Lichen
[Class], bright yellow or orange color 19.5%
[Class], hair-like or stringy appearance 18.9%
[Class], grows on rocks or trees 16.7%
[Class], small, cup-shaped fruiting bodies 14.3%
[Class], a powdery or dusty texture 15.7%
[Class], a circular or branching growth pattern 15.0%
Input Image (Wool Grass)
Ours
CBD
50%
25%
0%
Class Probabilities
Prediction:
Wool Grass
ornamental grasses with dynamic movement 15.3%
low-lying shrubs with a golden hue 10.7%
plumes of pampas grass with seasonal color changes 16.1%
ornamental grasses with unique textures 12.6%
tall ornamental grasses with feathery plumes 18.3%
dried grasses with interesting seed heads 27.0%
Prediction:
Wood Club-Rush
[Class], green or brown in color 17.5%
[Class], grows in wet or marshy areas 16.7%
[Class], small, clustered flowers at the top of the stem 14.0%
[Class], leaves are long and thin, resembling blades of grass 15.0%
[Class], small, spiky seed heads at the top of the stem 20.4%
[Class], a slight curve or bend in the stem 16.4%
GT Class:
Wool Grass
[Class], tall, grass-like appearance 18.8%
[Class], clusters of small, brown flowers at the top of the stem 22.6%
[Class], triangular shape of the stem when viewed from the side 18.7%
[Class], green or brown coloration 20.8%
[Class], grows in wet or marshy areas 19.1%

Fig. 5: Predictions. We show three different prediction examples. For each example, we show our method's prediction (first column), as well as classification by description [5] (CBD)'s prediction (second column), and CBD's attributes for the ground truth class (third column). For each column, we show the normalized probability per attribute. Below the input image, we show the probability distributions across classes for both our method and CBD. The results show that our learned attributes are more detailed and discriminative of the species within the family, compared to the description by classification (CBD) baseline. Furthermore, our method's class probability distributions tend to be more concentrated than CBD's.

Table 1: We report the accuracy per dataset, for our method and baselines.

	iNaturalist					Kiki-Bouba	
Method	Lichen	Wrasse	Wild Rye	Manzanita	Bulrush	KB1	KB2
Zero-shot Attributes	21.67	24.0	30.0	22.0	22.0	21.4	20.1
Class Name (CLIP [1])	21.67	28.0	32.0	20.0	26.0	40.7	43.7
Classification by Desc. [5]	23.33	36.0	30.0	26.0	22.0	28.9	41.7
Gradient-based Approach	23.3	20.0	40.0	20.0	20.0	16.7	55.6
Ours (1-prompt)	31.6	24.0	44.0	40.0	22.0	50.3	47.8
Ours (10-prompt)	**46.67**	**42.0**	**52.0**	**50.0**	**36.0**	**78.9**	**59.2**

	Zeroshot	CLIP	CBD	Ours
Zeroshot	13.6	10.0	12.4	28.8
CLIP	17.6	20.8	20.8	40.8
CBD	17.6	14.4	20.0	41.2
Ours	18.0	12.8	19.6	38.8

Table 2: Accuracy across the hierarchical iNaturalist 26-class dataset. Rows represent the method for coarse-level classification, and column represents the method for fine-grained level classification.

engineered text templates. We outperform all baselines with or without engineered text templates, and report the full numbers in the supplementary.

4.3 Fine-Grained Classification (iNaturalist)

Quantitative Result Discussion: We present the performance (accuracy) for our method and baselines in Tab. 1. Crucially, our method starts with zero prior knowledge about what is in a class of images, beyond the fact that those images are grouped together. The initialized best classifier at the beginning of optimization has no correspondence to the class name or quality, and only

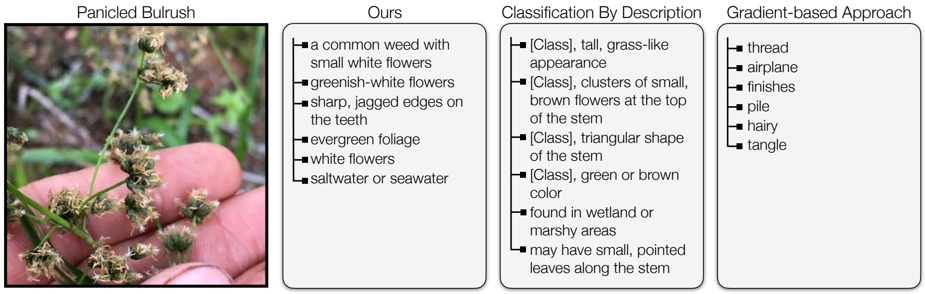
Panicled Bulrush
Ours
a common weed with small white flowers
greenish-white flowers
sharp, jagged edges on the teeth
evergreen foliage
white flowers
saltwater or seawater
Classification By Description
[Class], tall, grass-like appearance
[Class], clusters of small, brown flowers at the top of the stem
[Class], triangular shape of the stem
[Class], green or brown color
found in wetland or marshy areas
may have small, pointed leaves along the stem
Gradient-based Approach
thread
airplane
finishes
pile
hairy
tangle

Fig. 6: Comparison of Attributes by Method. We show qualitative examples of our learned attributes, classification by description's attributes (CBD), and our gradient-based approach attributes. CBD often produces reasonable attributes, but they are not discriminative, resulting in poor recognition accuracy. Gradient-based methods often produce poor attributes due to optimization difficulties.

through optimization our method learns to discover interpretable and reflective attributes. The first three baselines of zero-shot descriptors, class name, and class name with zero-shot descriptors on the other hand have prior knowledge of the class from the start and therefore are at a significant advantage as they wouldn't work for any concepts that the internet isn't already familiar with.

Additionally, we show results for a hierarchical model of classification. Using the taxonomy from iNaturalist, we train our method to classify objects at coarse levels first, i.e. the family, then fine-grained levels, i.e., the species, resulting in a 26-way multi-class classification problem across the five families we previously reported on. We report these results in Tab. 2 for different choices of method for the coarse and fine-level classification decision. While class names are sufficient for coarse-level classification, our method always outperforms class names and all other baselines for fine-grained classification.

Qualitative Result Discussion: In Fig. 4, we show the fine-grained classification results on the Lichen family. We observe that while all the Lichen are orange and yellow in color, the only references to color within the attributes are when the color diverges from the mean color, i.e. "reddish tint". The majority of the attributes reflect the Lichen's structure, which is due to the fact that all the Lichen are roughly the same color and grow in similar environments, the principal discriminating feature is the structure. In Fig. 6, we compare our learned attributes to the zero-shot attribute in CBD, as well as the learned attributes with our gradient-based approach. The gradient-based approach has far less descriptive and interpretable attributes. We provide more such qualitative examples of the three methods in the supplementary material.

In Fig. 5, we explicitly visualize the predictions across for three of the classifiers, one per row. Each example illustrates how each method compares to the top-performing baseline, classification by description (CBD). We show our method's prediction as well as CBD's prediction, along with the relative contribution of each attribute to the mean score. We additionally denote CBD's attributes for the ground truth class, along with the relative contribution to the ground truth mean score, to visualize why CBD may have incorrectly predicted the class. Across the three examples, we notice that CBD's attributes are less specific and detailed compared to ours, and that there are many shared attributes across different classes. Shared attributes are not useful in fine-grained classification, since the goal is to discriminate between classes.

4.4 Novel Objects (Kiki-Bouba)

Quantitative Result Discussion: We outperform all baselines on the two KikiBouba datasets. The gradient-based baseline has a large variation in performance. We hypothesize that this is due to the intra-class variation changing quite significantly across classes. Across all experiments, the gradient-based approach produces poor attributes due to optimization difficulties, as illustrated in Tab. 1.

Qualitative Result Discussion: For the novel image results on the first dataset

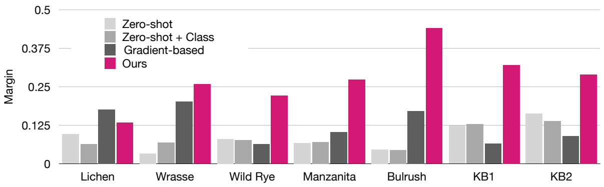

Fig. 7: Confidence. We show the mean margin of the scores between the top and runner-up prediction, which measures the typical confidence of each model.

of Kiki-Bouba. In Fig. 4, we show qualitative results of the learned attribute sets for the second Kiki-Bouba dataset. We notice that compared to the iNaturalist datasets, there is less similarity between classes, and the discovered attributes are more object-oriented. We suspect these observations are not unrelated, and that when an object name can be used to discriminate between classes, the vision-language model scores highly with it.

4.5 Analysis of Learning

Margin Metric: In Fig. 5, below the input image, we show the probability distributions across classes for both our method and CBD. We observe that in general, the probability distribution across classes for CBD is more uniform than ours. We investigate this observation by measuring the *margin* metric per dataset, per method. The margin is defined as $\max_{c \in C} f_c(x_i; \mathcal{D}) - \max_{\bar{c} \notin C} f_{\bar{c}}(x_i; \mathcal{D})$, which measures the difference in scores between the top prediction and the runner-up, indicating the prediction confidence [70]. We plot the margin for each method, per dataset, in Fig. 7. The results illustrate that our method is on average twice as confident as the other methods.

Attribute Evolution: In Fig. 3, we show examples of the iterations for pretraining, followed by the joint training. At initialization, the first sampled set of attributes for Greenleaf Manzanita hardly relate to the class. The closest attributes are “dark greenf leaves” and “green coloration”, with the other attributes having nothing in common with images. However, by the end of the attribute evolution, the color “green” is no longer part of the attributes, as the other Manzanita also have green leaves, making it non-discriminative. The final attributes contain descriptors that are particularly descriptive of the Greenleaf Manzanita. We share more examples of attribute evolution in the supplementary.

4.6 Auditing Dataset Bias

By having explicitly interpretable attributes as the bottleneck for classification, we can directly observe whether the classifier is picking up on dataset bias to perform prediction. An example of dataset bias can be seen in Fig. 8. This is unique to our method, as classifiers with no concept bottlenecks have no way of converting dataset bias into language, and previous work in concept bottlenecks do not discover the attributes from the data itself.

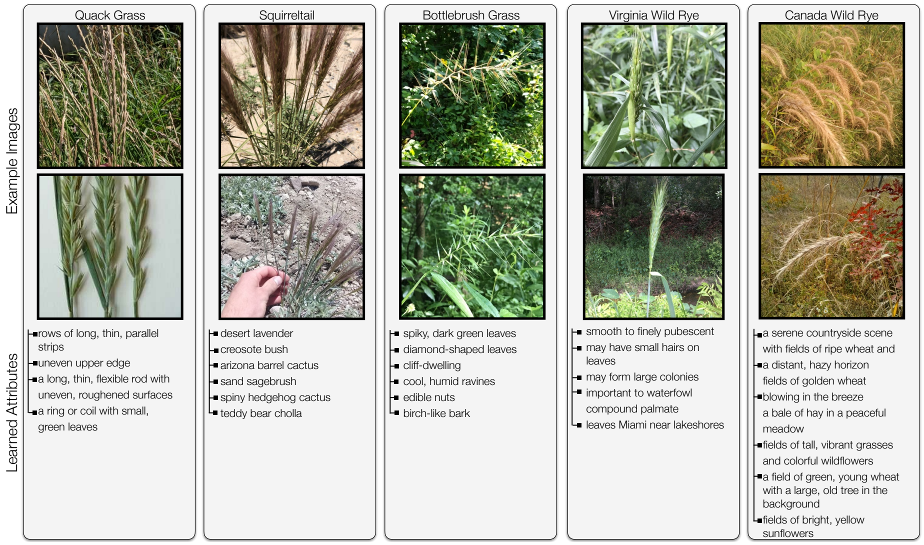
Example Images
Learned Attributes
Quack Grass
rows of long, thin, parallel strips
uneven upper edge
a long, thin, flexible rod with uneven, roughened surfaces
a ring or coil with small, green leaves
Squirreltail
desert lavender
creosote bush
arizona barrel cactus
sand sagebrush
spiny hedgehog cactus
teddy bear cholla
Bottlebrush Grass
spiky, dark green leaves
diamond-shaped leaves
cliff-dwelling
cool, humid ravines
edible nuts
birch-like bark
Virginia Wild Rye
smooth to finely pubescent
may have small hairs on leaves
may form large colonies
important to waterfowl compound palmate
leaves Miami near lakeshores
Canada Wild Rye
a serene countryside scene with fields of ripe wheat and
a distant, hazy horizon fields of golden wheat
blowing in the breeze a bale of hay in a peaceful meadow
fields of tall, vibrant grasses and colorful wildflowers
a field of green, young wheat with a large, old tree in the background
fields of bright, yellow sunflowers

Fig. 8: Dataset Bias. The **squirreltail** is the only species in the family that lives in drought habitats, and the learned attributes are names of plants that live in the desert. The ability to explicitly audit bias is an advantage of our interpretable method.

5 Discussion

Societal Impacts and Limitations. Explainable vision systems have the potential to have significant practical impact, especially in specialized, critical, and scientific domains. Such interpretable classifiers can establish trust, as they provide insight into how a classifier reached a decision. They allow people to audit the decision-making process, which is important for many practical cases. Finally, they also impact education, as the classifiers can report to a person the visual differences it has discovered, thereby helping the person learn about the recognized concept too. However, since our approach uses open-source LLMs, our method inherits known limitations about LLMs in bias and inappropriate generations. As research in LLMs advance, we expect our framework to improve too.

Conclusion. We propose a framework that integrates large language models and evolutionary search in order to learn interpretable, discrete attributes for visual recognition. In multiple datasets, our method outperforms existing baselines significantly.

Acknowledgements. This research was partially supported by the National Science Foundation AI Institute for Artificial and Natural Intelligence (ARNI) and the DARPA ECOLE program.

References

1. A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, *et al.*, "Learning transferable visual models from natural language supervision," in *ICML*, pp. 8748–8763, PMLR, 2021.
2. C. Schuhmann, R. Beaumont, R. Vencu, C. Gordon, R. Wightman, M. Cherti, T. Coombes, A. Katta, C. Mullis, M. Wortsman, *et al.*, "Laion-5b: An open large-scale dataset for training next generation image-text models," *Advances in Neural Information Processing Systems*, vol. 35, pp. 25278–25294, 2022.
3. V. Ferrari and A. Zisserman, "Learning visual attributes," *Advances in neural information processing systems*, vol. 20, 2007.
4. A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth, "Describing objects by their attributes," in *2009 IEEE conference on computer vision and pattern recognition*, pp. 1778–1785, IEEE, 2009.
5. S. Menon and C. Vondrick, "Visual classification via description from large language models," *ICLR*, 2023.
6. P. W. Koh, T. Nguyen, Y. S. Tang, S. Mussmann, E. Pierson, B. Kim, and P. Liang, "Concept bottleneck models," in *ICML*, 2020.
7. K. Roth, J. M. Kim, A. Koepke, O. Vinyals, C. Schmid, and Z. Akata, "Waffling around for performance: Visual classification with random words and broad concepts," *CoRR*, 2023.
8. J. H. Holland, *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press, 1992.
9. G. Van Horn, E. Cole, S. Beery, K. Wilber, S. Belongie, and O. Mac Aodha, "Benchmarking representation learning for natural world image collections," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12884–12893, 2021.
10. M. Alper and H. Averbuch-Elor, "Kiki or bouba? sound symbolism in vision-and-language models," *Advances in Neural Information Processing Systems*, vol. 36, 2024.
11. S. Huang, Z. Xu, D. Tao, and Y. Zhang, "Part-stacked cnn for fine-grained visual categorization," in *CVPR*, 2016.
12. B. Zhou, Y. Sun, D. Bau, and A. Torralba, "Interpretable basis decomposition for visual explanation," in *ECCV*, 2018.
13. L. Tang, D. Wertheimer, and B. Hariharan, "Revisiting pose-normalization for fine-grained few-shot recognition," in *CVPR*, 2020.
14. C. H. Lampert, H. Nickisch, and S. Harmeling, "Attribute-based classification for zero-shot visual object categorization," 2013.
15. A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, M. Ranzato, and T. Mikolov, "Devise: A deep visual-semantic embedding model," in *NeurIPS*, 2013.
16. Z. Akata, S. Reed, D. Walter, H. Lee, and B. Schiele, "Evaluation of output embeddings for fine-grained image classification," in *CVPR*, 2015.
17. B. Romera-Paredes and P. Torr, "An embarrassingly simple approach to zero-shot learning," in *ICML*, 2015.
18. E. Kodirov, T. Xiang, and S. Gong, "Semantic autoencoder for zero-shot learning," in *CVPR*, 2017.
19. R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *ICCV*, 2017.

20. H. Chefer, S. Gur, and L. Wolf, "Generic attention-model explainability for interpreting bi-modal and encoder-decoder transformers," in *ICCV*, pp. 397–406, 2021.
21. R. Fong, M. Patrick, and A. Vedaldi, "Understanding deep networks via extremal perturbations and smooth masks," in *ICCV*, 2019.
22. B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *CVPR*, 2016.
23. V. Petsiuk, A. Das, and K. Saenko, "Rise: Randomized input sampling for explanation of black-box models," *CoRR*, 2018.
24. V. Shitole, F. Li, M. Kahng, P. Tadepalli, and A. Fern, "One explanation is not enough: structured attention graphs for image classification," *NeurIPS*, 2021.
25. K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," *CoRR*, 2013.
26. Y. Goyal, Z. Wu, J. Ernst, D. Batra, D. Parikh, and S. Lee, "Counterfactual visual explanations," in *ICML*, 2019.
27. V. Prabhu, S. Yenamandra, P. Chattopadhyay, and J. Hoffman, "Lance: Stress-testing visual models by generating language-guided counterfactual images," *NeurIPS*, 2024.
28. S. Vandenhende, D. Mahajan, F. Radenovic, and D. Ghadiyaram, "Making heads or tails: Towards semantically consistent visual counterfactuals," in *ECCV*, 2022.
29. P. Wang and N. Vasconcelos, "Scout: Self-aware discriminant counterfactual explanations," in *CVPR*, 2020.
30. C.-K. Yeh, J. Kim, I. E.-H. Yen, and P. K. Ravikumar, "Representer point selection for explaining deep neural networks," *NeurIPS*, 2018.
31. C.-P. Tsai, C.-K. Yeh, and P. Ravikumar, "Sample based explanations via generalized representers," *CoRR*, 2023.
32. Y. Sui, G. Wu, and S. Sanner, "Representer point selection via local jacobian expansion for post-hoc classifier explanation of deep neural networks and ensemble models," in *NeurIPS*, 2021.
33. G. Pruthi, F. Liu, M. Sundararajan, and S. Kale, "Estimating training data influence by tracking gradient descent," *CoRR*, 2020.
34. P. W. Koh and P. Liang, "Understanding black-box predictions via influence functions," in *ICML*, 2017.
35. A. Silva, R. Chopra, and M. C. Gombolay, "Cross-loss influence functions to explain deep network representations," in *AISTATS*, 2020.
36. H. Guo, N. Rajani, P. Hase, M. Bansal, and C. Xiong, "Fastif: Scalable influence functions for efficient model interpretation and debugging," *CoRR*, 2020.
37. Y. Gandelsman, A. A. Efros, and J. Steinhardt, "Interpreting clip's image representation via text-based decomposition," 2023.
38. A. Dravid, Y. Gandelsman, A. A. Efros, and A. Shocher, "Rosetta neurons: Mining the common units in a model zoo," in *CVPR*, 2023.
39. M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *ECCV*, 2014.
40. J. Frankle and M. Carbin, "The lottery ticket hypothesis: Training pruned neural networks," *CoRR*, 2018.
41. A. M. Nguyen, A. Dosovitskiy, J. Yosinski, T. Brox, and J. Clune, "Synthesizing the preferred inputs for neurons in neural networks via deep generator networks," in *NeurIPS*, 2016.
42. C. Jia, Y. Yang, Y. Xia, Y.-T. Chen, Z. Parekh, H. Pham, Q. Le, Y.-H. Sung, Z. Li, and T. Duerig, "Scaling up visual and vision-language representation learning with noisy text supervision," in *International conference on machine learning*, pp. 4904–4916, PMLR, 2021.

43. M. Cherti, R. Beaumont, R. Wightman, M. Wortsman, G. Ilharco, C. Gordon, C. Schuhmann, L. Schmidt, and J. Jitsev, "Reproducible scaling laws for contrastive language-image learning," in *CVPR*, 2023.
44. M. Tsimpoukelli, J. L. Menick, S. Cabi, S. Eslami, O. Vinyals, and F. Hill, "Multimodal few-shot learning with frozen language models," *NeurIPS*, 2021.
45. J.-B. Alayrac, J. Donahue, P. Luc, A. Miech, I. Barr, Y. Hasson, K. Lenc, A. Mensch, K. Millican, M. Reynolds, *et al.*, "Flamingo: a visual language model for few-shot learning," *NeurIPS*, 2022.
46. Z. Wang, J. Yu, A. W. Yu, Z. Dai, Y. Tsvetkov, and Y. Cao, "Simvlm: Simple visual language model pretraining with weak supervision," *CoRR*, 2021.
47. K. Desai and J. Johnson, "Virtex: Learning visual representations from textual annotations," in *CVPR*, 2021.
48. J. Chen, H. Guo, K. Yi, B. Li, and M. Elhoseiny, "Visualgpt: Data-efficient adaptation of pretrained language models for image captioning," in *CVPR*, 2022.
49. L. H. Li, M. Yatskar, D. Yin, C.-J. Hsieh, and K.-W. Chang, "Visualbert: A simple and performant baseline for vision and language," *CoRR*, 2019.
50. X. Xu, C. Wu, S. Rosenman, V. Lal, W. Che, and N. Duan, "Bridgetower: Building bridges between encoders in vision-language representation learning," in *AAAI*, 2023.
51. H. Tan and M. Bansal, "Lxmert: Learning cross-modality encoder representations from transformers," *CoRR*, 2019.
52. J. Lu, D. Batra, D. Parikh, and S. Lee, "Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks," *NeurIPS*, 2019.
53. Y. Zeng, X. Zhang, and H. Li, "Multi-grained vision language pre-training: Aligning texts with visual concepts," *CoRR*, 2021.
54. L. Yuan, D. Chen, Y.-L. Chen, N. Codella, X. Dai, J. Gao, H. Hu, X. Huang, B. Li, C. Li, *et al.*, "Florence: A new foundation model for computer vision," *CoRR*, 2021.
55. L. H. Li, P. Zhang, H. Zhang, J. Yang, C. Li, Y. Zhong, L. Wang, L. Yuan, L. Zhang, J.-N. Hwang, *et al.*, "Grounded language-image pre-training," in *CVPR*, 2022.
56. L. Li, Z.-Y. Dou, N. Peng, and K.-W. Chang, "Desco: Learning object recognition with rich language descriptions," *NeurIPS*, 2024.
57. J. Li, D. Li, C. Xiong, and S. Hoi, "Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation," in *ICML*, 2022.
58. J. Li, D. Li, S. Savarese, and S. Hoi, "Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models," *CoRR*, 2023.
59. R. Mokady, A. Hertz, and A. H. Bermano, "Clipcap: Clip prefix for image captioning," *CoRR*, 2021.
60. Z. Luo, Y. Xi, R. Zhang, and J. Ma, "A frustratingly simple approach for end-to-end image captioning," *CoRR*, 2022.
61. S. Pratt, I. Covert, R. Liu, and A. Farhadi, "What does a platypus look like? generating customized prompts for zero-shot image classification," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 15691–15701, 2023.
62. A. Yan, Y. Wang, Y. Zhong, C. Dong, Z. He, Y. Lu, W. Y. Wang, J. Shang, and J. McAuley, "Learning concise and descriptive attributes for visual recognition," in *CVPR*, 2023.
63. B. Romera-Paredes, M. Barekatain, A. Novikov, M. Balog, M. P. Kumar, E. Dupont, F. J. Ruiz, J. S. Ellenberg, P. Wang, O. Fawzi, *et al.*, "Mathematical discoveries from program search with large language models," *Nature*, 2023.
64. S. Yu, S. Liu, Z. Lin, D. Pathak, and D. Ramanan, "Language models as black-box optimizers for vision-language models," *CoRR*, 2023.

65. S. Jin, X. Jiang, J. Huang, L. Lu, and S. Lu, "Llms meet vlms: Boost open vocabulary object detection with fine-grained descriptors," *CoRR*, 2024.
66. S. Han, L. Zhuo, Y. Liao, and S. Liu, "Llms as visual explainers: Advancing image classification with evolving visual descriptions," *CoRR*, 2023.
67. H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, *et al.*, "Llama 2: Open foundation and fine-tuned chat models," *CoRR*, 2023.
68. V. S. Ramachandran and E. M. Hubbard, "Synaesthesia–a window into perception, thought and language," *Journal of consciousness studies*, vol. 8, no. 12, pp. 3–34, 2001.
69. B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, *et al.*, "Language models are few-shot learners," *CoRR*, 2020.
70. P. Melville and R. J. Mooney, "Diverse ensembles for active learning," in *Proceedings of the twenty-first international conference on Machine learning*, p. 74, 2004.