This ICCV paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Verbalized Representation Learning for Interpretable Few-Shot Generalization

Cheng-Fu Yang[1*] Da Yin[1*] Wenbo Hu[1] Heng Ji[2] Nanyun Peng[1] Bolei Zhou[1] Kai-Wei Chang[1]
[1]University of California, Los Angeles [2]University of Illinois Urbana-Champaign
cfyang@cs.ucla.edu

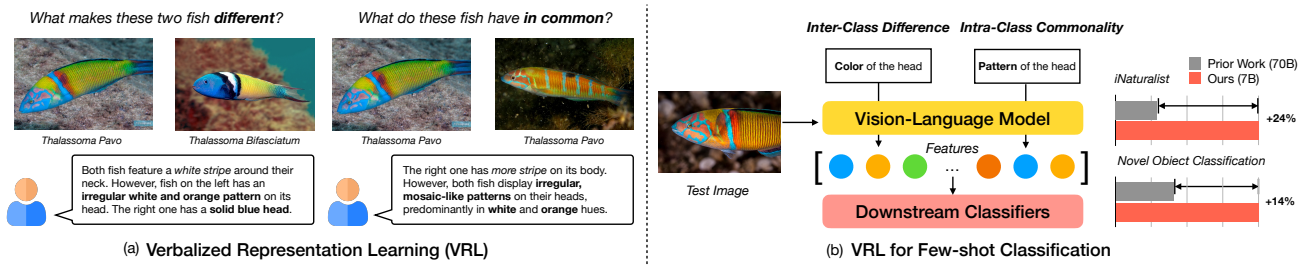


Figure 1. (a) Humans can identify key differences between similar species and recognize common traits within a class, even when appearance varies, and express these insights in concise language. Similarly, our Verbalized Representation Learning (VRL) extracts meaningful features by querying Vision-Language Models (VLMs) to capture inter-class discriminative features and intra-class commonalities. (b) These verbalized features are mapped to numerical vectors through the VLM and can be used by downstream classifiers. Our VRL demonstrates superior performance in few-shot classification compared to prior work, achieving a 24% absolute gain on iNaturalist and a 14% on Novel Object Classification, even with a much smaller model.

Abstract

Humans recognize objects after observing only a few examples, a remarkable capability enabled by their inherent language understanding of the real-world environment. Developing verbalized and interpretable representation can significantly improve model generalization in low-data settings. In this work, we propose Verbalized Representation Learning (VRL), a novel approach for automatically extracting human-interpretable features for object recognition using few-shot data. Our method uniquely captures inter-class differences and intra-class commonalities in the form of natural language by employing a Vision-Language Model (VLM) to identify key discriminative features between different classes and shared characteristics within the same class. These verbalized features are then mapped to numeric vectors through the VLM. The resulting feature vectors can be further utilized to train and infer with downstream classifiers. Experimental results show that, at the same model scale, VRL achieves a 24% absolute improvement over prior state-of-the-art methods while using 95% less data and a smaller model. Furthermore, compared to human-labeled attributes, the features learned by VRL exhibit a 20% absolute gain when used for downstream classification tasks. [1]

[1]The code is available at: https://github.com/joeyy5588/VRL.

1. Introduction

Humans have a remarkable capability to recognize certain objects after seeing only a few examples. As suggested by [33], it is significantly enhanced by inherent language understanding. Language, with its representational strength, serves as a primary resource for conveying knowledge about visual objects. A single, precise description can effectively capture visual distinctions observed across different object categories. For example, in Fig. 1 (a), a human can identify the key difference between two fish species based on the pattern of their heads, even if both share a white stripe around the neck and a yellow body. As a result, we argue that developing verbalized features offers a valuable complement to improve model's generalization under low-resource conditions. In addition, incorporating language into image classification models also increases the interpretability of the visual system, facilitating transparent and reliable decision making and effective model auditing.

Prior works [15, 16] have explored integrating language into visual classifiers with a bottleneck of textual attributes, where the model first predicts relevant attributes and subsequently uses them to classify the image. However, these attributes often require human annotations or predefined ontologies. Development of Large-scale Vision-and-Language Models (VLMs) like CLIP [39] and GPT-4V [1] have enabled recent bottleneck methods [12, 32, 38, 48, 50]

to generate relevant attributes by leveraging the prior knowledge of VLMs learned during the pre-training stage. However, these attributes are often ungrounded for the testing images and rely on VLM's prior knowledge, resulting in low precision when applied to fine-grained or novel concept recognition where objects are underrepresented in the pre-training datasets.

To address the aforementioned challenges, we propose *Verbalized Representation Learning* (VRL) for **automatic**, **human-interpretable** feature extraction using only **few-shot** data. It applies to *fine-grained* and *novel* objects and could work with local models such as LLaVA [28]. Specifically, inspired by self-supervised representation learning (SSL) in computer vision, such as SimCLR [8], MoCo [19], SwAV [6], SimSiam [10], BYOL [18], CLIP-Event [29], and SOLO [11], we propose to leverage a VLM to capture the *inter-class difference* and *intra-class commonality* and articulate these findings in natural language, as illustrated in Fig. 1. Specifically, we cast the VLM to describe the key difference between two images from different classes, which would preserve the discriminative features and remove the redundant ones, such as the yellow body shared by both species on the left side of Fig. 1 (a). Conversely, we employ the VLM to list the key features that are shared by two images within the same class, extracting features that are robust to intra-class variance, such as the orange facial pattern observed in all of the *Thalassoma Pavo*. Notably, this process can be applied to any two images, whether from different classes or the same class, which allows us to exponentially scale the number of useful verbalized features with the image samples, enabling effective generalization even in few-shot settings.

Consequently, to obtain the numerical feature embedding of an image using our VRL, we employ a VLM to determine whether the image possesses the characteristics described by the verbalized features, as depicted in Fig. 1 (b). Each dimension in the resulting embedding represents a scalar value indicating the presence of the described feature. This approach effectively transforms the verbalized features into numeric representations that can serve as inputs to any classification methods, such as logistic regression, random forest, or MLP classifiers, allowing flexible and robust modeling based on the extracted interpretable features.

We conduct experiments on iNaturalist [47] and Kiki-Bouba dataset [2], where the former includes objects that only have subtle differences between classes, while the latter contains novel objects that are nearly not present in the web-scale datasets. Compared to previous state-of-the-art (SoTA) baselines using 70B models, we achieve a **24**% absolute improvement while using **95**% less data and a much smaller model with 7B parameters. Against supervised fine-tuned baselines, we observe a **15**% absolute improvement. Lastly, when compared to human annotated attributes, the attributes learned by our VRL demonstrate a **20**% absolute gain. These results illustrate that the features extracted by VRL not only offer superior effectiveness but also exhibit strong robustness, providing a generalizable solution when adapting to fine-grained or novel classification tasks with limited data.

2. Related Work

Natural Language for Image Classification. A common approach for integrating language into visual classifiers involves creating a concept bottleneck [26]. Early bottleneck methods have been extensively applied in few-shot or zero-shot classification models [7, 15–17, 23, 26, 27, 42]. However, these methods typically rely on manually annotated or predefined attributes, which can limit their adaptability to novel classes or fine-grained tasks. Recent advancements in VLMs have enabled researchers to directly sample interpretable descriptive features from these models [12, 32, 38, 45, 48–51]. This approach benefits from incorporating external knowledge embedded in the pre-trained datasets. However, it often relies heavily on the prior knowledge encoded within these models, which can lead to the generation of ungrounded features and difficulty generalizing when the target data is not well-represented in the training datasets. In contrast, our method directly learns grounded, verbalized features from the visual data, allowing for automatic interpretable feature extraction that remains adaptable even for novel classes.

Interpreting Model's Decision Process. There has been a long line of research aiming to improve the interpretability and explainability of deep models. Pioneering methods [3, 4, 13, 24, 43, 52] have tackled this challenge by visualizing learned features, categorizing maximally-activating inputs, and identifying key neurons that drive model decisions. More recently, efforts have been made to interpret model behavior using natural language [20, 21, 36, 44], enabling explanations that are more accessible and editable. However, the aforementioned methods are generally post-hoc, meaning they are applied after the model has been trained with abundant data. While post-hoc explanations can provide valuable insights, they may lack the ability to influence or shape the model's internal representations during training. Instead, our methods embed interpretability directly into the learning process by learning verbalized features. This enables the model to produce inherently explainable and contextually grounded representations without relying solely on retrospective analysis.

Self-Supervised Learning Models. Self-supervised learning (SSL) has emerged as a powerful approach in computer vision, enabling models to learn robust feature representations from unlabeled data. Contrastive learning [34] methods, such as SimCLR [8] and MoCo [19], achieve this goal by learning to discriminate between different samples (neg-

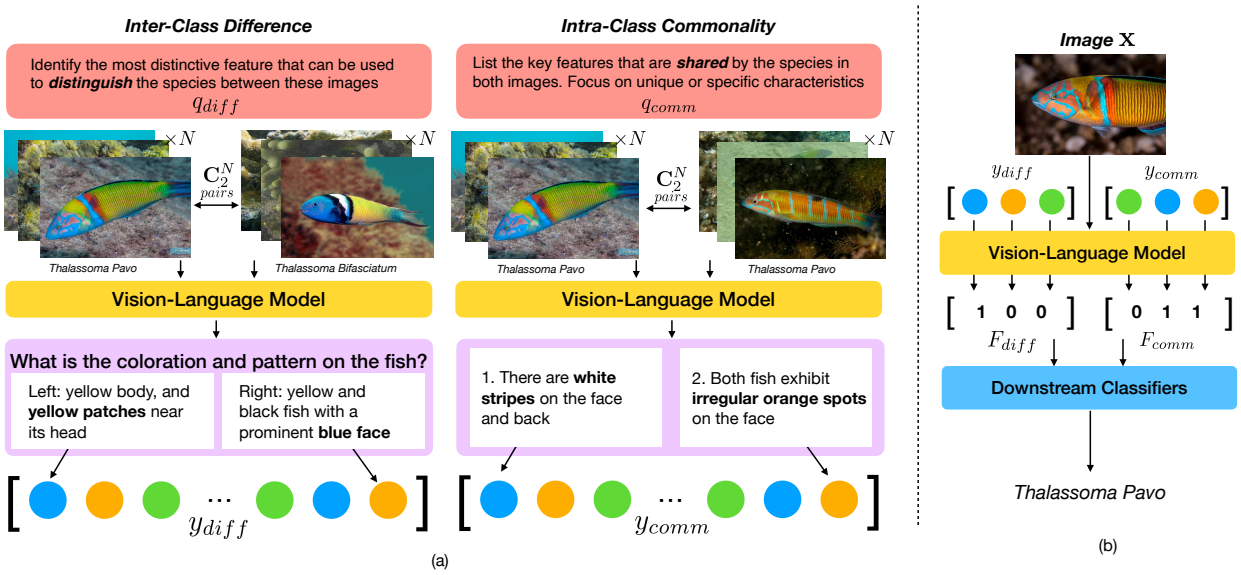


Figure 2. The overview of our Verbalized Representation Learning (VRL) framework. (a) Given N samples per class from C different classes, VRL is able to generate a diverse set (exponentially scaling with $N \times C$) of verbalized features by: 1) extracting key differences between samples from different classes, and 2) identifying commonalities shared among objects within the same class. (b) Given an image, a Vision-and-Language model (VLM) is employed to evaluate whether the image contains the characteristics described by the verbalized features. This process can map a set of verbalized features into numeric representations which can then be used for downstream tasks.

ative pairs) and augmentations of the same sample (positive pairs). Chen et al. [10] have found that the key function of negative pairs is to prevent the model from learning collapsing features, where models produce constant or trivial outputs. As a result, subsequent works have relieved the need for negative samples by employing techniques like clustering [5, 6], momentum update [18], or stop gradient [10]. These methods focus on learning the underlying shared representations between the augmentations of the same sample. Our approach can be viewed as a variant of supervised contrastive learning [25], where the positive samples are drawn from different images from the same class. However, our method introduces two distinct advantages. First, it is data-efficient, as it does not require large datasets for gradient updates. Second, the expressiveness of verbalized features inherently prevents the model from collapsing to constant outputs, thereby ensuring the resulting features are meaningful and representative.

3. Method

In Sec. 3.1, we first introduce the motivation and concept of *verbalized representation learning* (VRL) and then discuss how VRL automatically extracts interpretable, compact representations with few-shot data. Then, in Sec. 3.2, we outline the process of building a visual classifier using the derived features. Furthermore, we demonstrate that these extracted features are versatile and can be applied to arbitrary classification models, including but not limited to logistic regression, decision trees, and multi-layer perceptron (MLP) classifiers. The overall framework of our VRL is presented in Fig. 2.

3.1. Verbalized Representation Learning

Humans can recognize objects after seeing only a few examples, a skill boosted by language understanding [33]. Language effectively conveys visual distinctions, even without many extra visual cues. Incorporating language also enhances image classification interpretability, enabling clearer, more reliable decisions. To this end, we propose a method that verbalizes key visual features in natural language. In this section, we focus on identifying which visual features are crucial for few-shot image classification and how to extract and verbalize them effectively, and how to map these verbalized features into vectors which would be later utilized during training and inference.

What visual features should be verbalized? Inspired by self-supervised representation learning methods, where model learns robust features through contrasting positive and negative pairs [8, 19] or maximizing the similarity of augmented views of the positive samples [6, 10, 18], we propose to verbalize the objective functions of these self-supervised learning methods with the help of VLMs such as LLaVA. Specifically, given a classification task with C classes and N-shot examples per class, each data point is defined as (x, l), where x represents an image, and $l \in \{1, 2, \ldots, C\}$ is the categorical label indicating the class to which the image belongs. Our method leverages two types of paired images: positive pairs, defined as image pairs (x_i, x_j) with the same label ($l_i = l_j$), and negative pairs, which consist of images (x_i, x_k) from different

classes ($l_i \neq l_k$). Notably, with this pairing strategy, we can form $\mathbf{C}_2^C \times \mathbf{C}_2^N$ distinct negative pairs, and $C \times \mathbf{C}_2^N$ positive pairs, which significantly increase the data utilization under the few-shot setting.

Inspired by contrastive learning methods [8, 19], for each negative pair sample, our approach emphasizes capturing *inter-class differences* by tasking the VLM to describe key distinguishing features between images from different classes, denoted as $y_{diff} = \text{VLM}(\mathbf{x}_i, \mathbf{x}_k, q_{diff})$, where $l_i \neq l_k$ and q_{diff} denotes the query that captures the visual differences between the two input images $\mathbf{x}_i$ and $\mathbf{x}_k$. For instance, in Fig. 2 (a), the model learns to distinguish two fish species by the coloration and the pattern of the fish. Conversely, for each positive pair, we draw inspirations from negative sample-free SSL methods [6, 10, 18], where we employ the VLM to capture *intra-class commonalities*, generating descriptions of the key shared features between two images from the same class, i.e., $y_{comm} = \text{VLM}(\mathbf{x}_i, \mathbf{x}_j, q_{comm})$, where $l_i = l_j$ and q_{comm} denotes the query that captures the visual commonalities between $\mathbf{x}_i$ and $\mathbf{x}_j$. For example, in Fig. 2 (a), VRL identifies that both fish possess irregular orange spots around their face. We include the detailed prompt templates used in the above process in Appendix A.1.

Intuitively, y_{diff} and y_{comm} learn complimentary features. y_{diff} captures the most discriminative features while filtering out redundant ones, such as the yellow body present in both types of fish, leading to less noisy features. While y_{comm} is robust to the intra-class variance where it learns to identify the fish based on the shared features, like the pattern on its face, rather than the number of stripes on its back. In the later experimental section, we will further verify this assumption and show that these two features can yield the best performance when combined. In addition, as discussed in the earlier paragraph, even with only N-shot samples, our VRL can sample a diverse set of features from $\mathbf{C}_2^C \times \mathbf{C}_2^N$ negative and $C \times \mathbf{C}_2^N$ positive pairs, making our method excel under low-resource setting. Moreover, we empirically discover that sampling $C \times N$ pairs for both negative and positive samples is sufficient to collect a robust set of features.

How to map the verbalized features to numeric feature embeddings? To obtain feature vectors that can be later used to build visual classifiers, we transform the verbalized features into numeric vectors with the help of VLMs such as CLIP [39] and LLaVA [28]. Given an image and a set of verbalized features, which are in the form of language descriptions, we employ a VLM to determine whether the image possesses those features, as illustrated in Fig. 2 (b), resulting in a feature embedding F, i.e., $F = \text{VLM}(y, x)$. Each dimension of F is produced by the VLM, indicating the presence of a certain verbalized feature from y_{diff} and y_{comm} or assessing the degree that the image has a certain feature. Concretely, for generative VLMs like LLaVA, each verbalized feature is mapped to 0 or 1, based on whether the model infers the presence of the feature in the image. For VLMs like CLIP, feature embeddings are derived by calculating the similarity between each verbalized feature and the image using CLIP's visual encoders. This results in a continuous vector indicating the likelihood that the image contains each feature, with an optional similarity threshold to convert it into binary feature vectors.

3.2. Training and Inference

Training. Once the numeric feature vectors are generated, they are used as input for training various visual classifiers, including, but not limited to, logistic regression, random forests, Naive Bayes, K-nearest neighbors (KNN), and MLP classifiers. These models learn to predict the class labels based on the interpretable feature vectors. This flexibility enables robust modeling that can be tailored to different applications. Moreover, this approach not only enables the construction of classifiers for novel concepts with few-shot data, but also provides insight into which visual features are most important for decision-making. For instance, we can extract feature importance from logistic regression or visualize decision paths in decision tree classifiers, enhancing the interpretability of our method.

Inference. During inference, given a testing image, we generate its feature vector by following the approach described in Sec. 3.1. Specifically, we query the VLMs with the learned verbalized features to determine whether the image contains specific features, producing a numeric feature vector. This vector can either be continuous, representing the likelihood of a feature's presence, or binary, indicating the feature's absence or presence. The resulting vector is then passed to the trained classifier to make a prediction. Notably, to enhance the model robustness, we ensemble results from classifiers trained with different algorithms. Combining predictions from multiple classifiers is beneficial because it reduces the likelihood of overfitting to any specific algorithm's biases or weaknesses. The ensemble can use either hard or soft voting: In hard voting, each classifier makes a discrete prediction, and the final prediction is determined by a majority vote. In soft voting, the prediction logits from each classifier are averaged to produce the final output, allowing for more nuanced decision-making.

4. Experiments

Our experiments are designed to answer the following research questions: (i) How well does VRL generalize to tasks requiring fine-grained recognition with few-shot data? (ii) Can VRL effectively adapt to tasks involving novel concepts that were not part of the VLM's pre-training resources, using only few-shot data? (iii) How effective is

VRL compared to conventional few-shot adaptation algorithms like LoRA fine-tuning or in-context learning? (iv) Do different types of features extracted by VRL mutually benefit each other, and do the extracted features further improve on top of the other commonly used features? (v) How does the performance of automatically features extracted with VRL compare to human-labeled features?

4.1. Dataset and Evaluation Protocols

We conduct experiments on iNaturalist and Kiki-Bouba to validate our method's effectiveness under varied scenarios, with all reported numbers representing classification accuracy (%). Although our primary analysis centers on these datasets, additional results for general few-shot classification on mini-ImageNet [41] are provided in the Appendix B.3.

Fine-Grained Species Classification. For fine-grained classification, we utilize the iNaturalist 2021 dataset [47], which comprises a diverse collection of images and annotations contributed by citizen scientists, spanning numerous species of animals, plants, and fungi. Each species in the training split contains between 200 and 300 images, while the validation split includes 10 images per species.

Following [12], we experiment on images from five different families, each containing five to six species. These families are selected due to their challenging nature: distinguishing between species within the same family requires the model to identify complex features such as shapes and patterns, rather than simple color variations. The families used in our experiments are as follows: *Lichen* (fungi), *Wrasse* (fish), *Wild rye* (grass), *Manzanita* (berry shrubs), and *Bulrush* (herbs). For a detailed list of the species, please refer to the appendix. Notably, unlike [12], which utilizes a full training set containing between 200 and 300 images per species, our method uses on only **10** images per species to achieve generalization in a low-resource setting. We will report the baseline performance from their original paper as well as our reproduced results using the same limited data as our method for comparison.

Novel Concept Classification. To test the model's generalizability on objects that have been rarely seen in the pre-training image resources, we evaluate our method using the Kiki-Bouba [2] dataset. The Kiki-Bouba experiment, originally introduced by [40], illustrates that people often associate specific shapes with different sounds. The dataset was constructed by prompting generative models trained to create 3D-rendered images from non-existent, meaningless words [2], providing a unique testbed for evaluating generalization to novel and abstract concepts. Following the approach in [12], we evaluate our method on two distinct splits, each containing five different classes. The first split consists of the classes *bamaba, duludu, gaduga, lomulo, nomano*, while the second split includes *bouba, galaga, kepike, kiki, maluma*. In [12], the training set comprises 800 images per class, with 200 images per class for validation. In contrast, we adopt a similar low-resource setting as used for the iNaturalist dataset, with our method accessing only **10** training images per class. Baseline results for both settings will be reported in the table for comparison.

4.2. Baseline Methods and Implementation Details

To demonstrate the effectiveness of our method, we compare it against several baselines, including the CLIP-based approach proposed in [12] and other baseline methods that utilize LLMs, such as LaBo [50], CBD [32] and LLM-Mutate [12], which leverage LLM's pre-trained knowledge to generate attributes. Additionally, since our method leverages LLaVA-OneVision [28] to generate verbalized features, we perform quantitative comparisons with common approaches used to adapt LLaVA for downstream tasks, such as LoRA fine-tuning [22] and in-context learning [14]. We describe the details of each baseline method as follows.

CLIP-Based Methods. For CLIP-based baselines, we consider two variants. *CLIP Class Name* is a naive baseline where CLIP is employed to compute the similarity between the species' scientific name and the images. The best accuracy is reported as the highest value achieved among using the common name, the scientific name, or both names combined. The second variant involves *Prompt Tuning* with a CLIP encoder, where the text embeddings of specific class-related tokens are optimized through gradient descent. For both baselines, we present results as reported in [12].

Language Bottleneck Methods. *Classification by Description* [32], *LaBo* [50], and *LLM-Mutate* [12] generate a list of attributes by prompting LLMs like GPT [35] and Llama [46] with the class name of the object. While this allows them to leverage the prior knowledge learned by LLMs, the generated features often lack grounding in the actual images. *LaBo* and *LLM-Mutate* alleviate this issue by using submodular optimization and evolutionary algorithm, respectively, to filter out ungrounded attributes. However, these methods still rely heavily on the capabilities and learned knowledge of the pre-trained LLMs. For *LaBo*, we adapt the official implementation for our datasets; for *LLM-Mutate*, we not only report the results from the original paper but also evaluate the method using the same limited data as our approach, ensuring a fair comparison.

LLaVA-Based Methods. We include two common approaches for adopting LLaVA. The first baseline involves performing LoRA [22] fine-tuning (*LLaVA-SFT*), which has been demonstrated to be effective in scenarios with limited data. The second baseline is in-context learning [14] (*LLaVA-ICL*), where one exemplar image for each species or object is included in the prompt to guide the model's predictions. We study which of the LLaVA-based methods and VRL better utilizes LLaVA's capabilities.

Method	Lichen	Wrasse	Wild Rye	Manzanita	Bulrush	Average
Zero-Shot Methods						
CLIP Class Name [12]	23.3	32.0	32.0	26.0	26.0	27.86
Full Dataset Methods (200+ Images per Species)						
CLIP Prompt Tuning [12]	23.3	20.0	40.0	20.0	20.0	24.66
Classification by Description [32]	30.0	34.0	36.0	28.0	20.0	29.60
LLM-Mutate-70B [12] (1-prompt)	31.6	24.0	44.0	40.0	22.0	32.32
LLM-Mutate-70B [12] (10-prompt)	48.3	44.0	58.0	58.0	42.0	50.06
Few-Shot Methods (10 Images per Species)						
LLM-Mutate-7B† (10-prompt)	35.0	48.0	38.0	44.0	26.0	38.20
LLM-Mutate-70B† (10-prompt)	46.6	44.0	46.0	44.0	40.0	44.13
LaBo-7B† [50]	45.0	48.0	44.0	42.0	36.0	43.00
LaBo-70B† [50]	56.6	46.0	64.0	52.0	40.0	51.72
LLaVA-ICL-7B	16.6	28.0	22.0	18.0	30.0	22.92
LLaVA-SFT-7B	41.6	50.0	58.0	42.0	28.0	43.92
LLaVA-VRL-7B (**Ours**)	58.3	48.0	74.0	**66.0**	46.0	58.46
LLaVA-VRL-72B (**Ours**)	**71.6**	**72.0**	**74.0**	56.0	**66.0**	**67.92**

Table 1. Comparison of classification accuracy (%) across different methods for fine-grained classification on iNaturalist. The table presents results for zero-shot, full-dataset (200+ images per species), and few-shot (10 images per species). Results marked with † denote values reproduced using the official implementation but restricted to the same few-shot data as our method.

Our Method. We implement VRL with LLaVA-OneVision for capturing visual difference and commonality features. We also select LLaVA-OneVision as the feature mapping model that assists to convert an image into a feature vector, and the base model for all the LLaVA-based baseline methods. For the few-shot training stage, we experiment with various classification methods including logistic regression or MLP. Unless otherwise specified, we utilize both the difference features F_{diff} and commonality features F_{comm}. We report the best performance achieved across different classifiers, selecting the optimal results from a *single* classifier.

4.3. Fine-Grained Species Classification

We present the results for fine-grained species classification on the iNaturalist dataset in Table 1. We find that compared to previous SoTA methods, even with 95% less data (10 images per species compared to 200+ images per species), and using a significantly smaller model (7B v.s. 70B parameters), our method can already surpass the previous SoTA by 8%. When we increase the scale of our method to a comparable 72B model, we can further increase the improvements to nearly 18%. Moreover, when baseline methods are limited to the same data availability as our approach, our 7B and 72B models achieve improvements of 20% and 23%, respectively. These results highlight the advantage of using grounded descriptions to capture the subtle visual cues, especially when the model needs to discover visual nuances to discriminate species within the same family.

We also compare VRL against LLaVA-based baselines. Using the same backbone LLaVA model, our method showcases a 15% gain to *LLaVA-SFT* and a 36% advantage over the in-context learning method, *LLaVA-ICL*. In addition, VRL is easier to be scaled up than the baselines. For instance, VRL with the 72B model requires 8 GPUs with 48 GB of RAM, while fine-tuning the 13B model with LoRA already requires 8 GPUs with 80 GB of RAM. For in-context learning, including image examples for every category in the prompt significantly increases the memory usage of the method, making it infeasible to perform in-context learning on the 72B LLaVA with the 8 GPUs (48 GB). The model performance may also be constrained by the context window of LLaVA model. These results and analysis further validate the scalability and robustness of VRL comparing to those existing adaptation algorithms.

4.4. Novel Concept Classification

For novel concept classification, we report the results on the Kiki-Bouba dataset in Table 2. Similar to the performance trend observed in the previous experiment, our method surpasses the previous SoTA by 14% with a smaller model, but with 99% less data (10 images per species compared to 800+ images per species). Surprisingly, we observe that increasing the number of model parameters does not lead to better performance on this task. One possible explanation for this is that, rather than relying on prior knowledge learned during pre-training, the model needs to focus on learning new, critical features specific to these novel objects with unfamiliar shapes and patterns. Since the objects in this task are unseen during LLM training, existing models struggle to generate relevant attributes and fail to ground visual differences necessary for distinguishing objects from different classes. This hypothesis is further supported by the larger performance gap between our method and those that rely on LLM's pre-trained knowledge [32]. On this dataset, our method exceeds their performance by 50%, compared

Method	v1	v2	Avg.
Zero-Shot Methods			
CLIP Class Name	38.7	38.8	38.75
Full Dataset Methods (800+ Images per Object)			
CLIP Prompt Tuning	16.7	55.6	36.15
Classification by Description	28.8	36.8	32.80
LLM-Mutate-70B (1-prompt)	50.3	47.8	49.05
LLM-Mutate-70B (10-prompt)	79.2	59.4	69.30
Few-Shot Methods (10 Images per Object)			
LLM-Mutate-7B† (10-prompt)	63.3	59.2	61.25
LLM-Mutate-70B† (10-prompt)	67.3	59.2	63.25
LLaVA-ICL-7B	24.5	27.2	25.85
LLaVA-SFT-7B	72.1	50.6	61.35
LLaVA-VRL-7B (**Ours**)	**89.4**	**76.6**	**83.00**
LLaVA-VRL-72B (**Ours**)	89.1	74.7	81.90

Table 2. Accuracy (%) across different methods for novel concept classification on Kiki-Kouba dataset. v1 and v2 indicates two different splits described in Sec. 4.1. The table presents results for zero-shot, full-dataset, and few-shot. Results marked with † denote values reproduced using the official implementation but restricted to the same few-shot data as our method.

to a 37% advantage on the previous dataset.

4.5. Additional Analysis

Fusing Features Learned by VRL and Commonly Used Features. Table 3 summarizes the classification accuracy on iNaturalist achieved by fusing different types of visual features. Specifically, we study the two feature vectors: F_{diff} and F_{comm} learned via our VRL, which focusing on capturing regional nuances of the object. F_{diff} captures the inter-class difference while F_{comm} represents the shared features within the same class. We also consider the image features directly encoded by the vision encoder of CLIP, which often encode high-level semantics such as the object type, context, and other conceptual associations. As discussed in Sec. 3.1, these features are robust to different kinds of variance and could be benefited when combined for downstream tasks. This is evident from the 2% improvements when concatenating F_{diff} and F_{comm}, and an additional 6% gain after including the CLIP feature.

We also discover that ensemble classifiers trained separately on each feature type are more effective than simply concatenating all features to build a single classifier. This is primarily because these features are heterogeneous. For example, F_{diff} and F_{comm} are generally binary, with each element indicating the presence of a specific verbalized feature. Therefore, they tend to perform best when used with logistic regression models. Meanwhile, the features encoded by CLIP are continuous, which makes them more suitable for mapping to class labels through MLP models. Empirically, we demonstrate that the best configuration of ensembling can further improves the performance by 3% comparing to feature concatenation. We find that with three

Method	F_{diff}	F_{comm}	CLIP	Avg.
Concat	✓			65.26
Concat		✓		58.32
Concat			✓	63.26
Concat	✓	✓		67.92
Concat	✓	✓	✓	73.92
Ensemble (hard)	LR	LR	LR	68.32
Ensemble (hard)	MLP	MLP	MLP	74.26
Ensemble (hard)	LR	LR	MLP	71.60
Ensemble (soft)	LR	LR	LR	67.06
Ensemble (soft)	MLP	MLP	MLP	74.92
Ensemble (soft)	LR	LR	MLP	**76.52**

Table 3. Accuracy (%) when incorporating feature vectors learned from different methods. F_{diff} and F_{comm} denote the difference and commonality feature vectors learned from VRL. CLIP refers to the image features encoded by CLIP visual encoder. We compare two approaches: concatenating all features to train a single classifier and ensembling classifiers trained separately on each feature type. The visual classifiers include Logistic Regression (LR) and Multi-Layer Perceptron (MLP) models.

classifiers, hard-voting is prone to high variance, while soft-voting is more robust since it leverages the confidence of each classifier by averaging their probabilities. These experiments showcase that our approach is flexible and can be seamlessly integrated with existing representations to build a robust classification model through ensembling.

Comparison with Human-Labeled Features. To evaluate the performance of automatically extracted features compared to human-labeled features, we conduct experiments on the Kiki-Bouba dataset using the human-annotated attributes provided by [12]. From Table 4, we observe that VRL extracted features outperform human-labeled attributes across all configurations. Our best configuration, which combines both difference feature vector F_{diff} and commonality feature vector F_{comm}, surpasses the human-derived features by 20%. This underscores the effectiveness of our automatic feature extraction approach in capturing relevant and distinguishing characteristics for classification tasks. These results also suggest that, in scenarios with limited resources or new visual concepts, we can automate the process of feature extraction, reducing the need for manual human annotation.

Classifier Choices and Feature Mapping Models. We investigate the model performance using different choices for visual classifiers and feature mapping models. From Table 5, we notice that LLaVA, as a feature mapping model for obtaining binary feature vectors, achieves the best overall performance when paired with the logistic regression.

As mentioned in Sec. 3.1, we can also leverage CLIP as a feature mapping model to convert images into features. When predicting, one can either set a threshold to determine if the image possesses a certain attribute, resulting in a binary feature, or preserve the similarity between the image and each attribute to form continuous features. Although the performance of using CLIP features is slightly lower

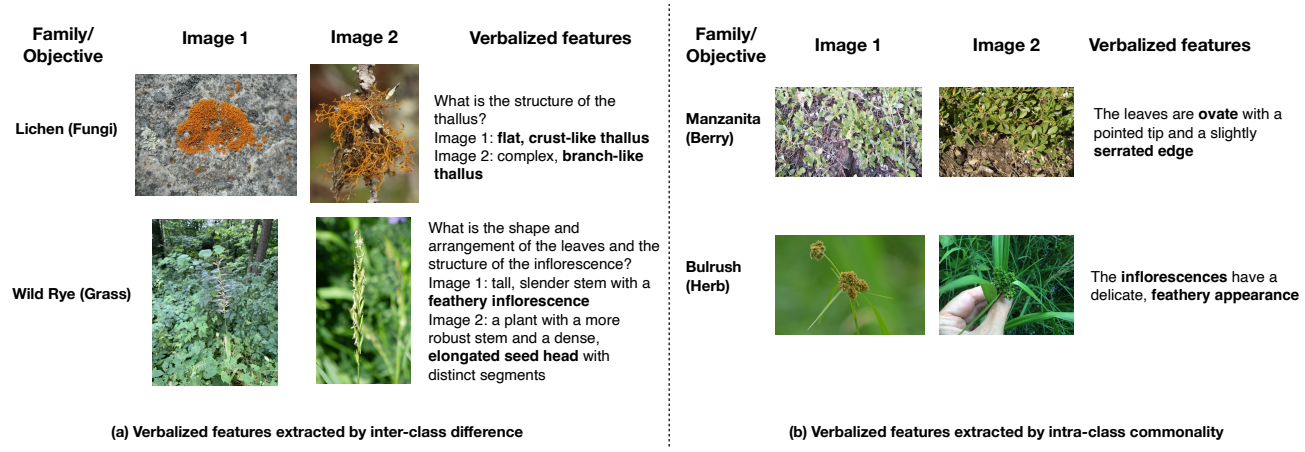


Figure 3. Qualitative examples of the features extracted by VRL. We highlight the key attributes in **bold**. (a) Verbalized features extracted by comparing images from different classes. (b) Verbalized features extracted by comparing the images within the same class.

Method	v1	v2	Avg.
Human	73.8	52.5	63.15
VRL-F_{diff}	88.4	75.2	81.80
VRL-F_{comm}	89.2	74.0	81.60
VRL-both	**89.4**	**76.6**	**83.00**

Table 4. Accuracy (%) of using human-labeled attributes and VRL extracted features on Kiki-Bouba dataset. Note that v1 and v2 indicates two different splits described in Sec. 4.1.

Method	RF	LR	MLP
iNaturalist (Prior Works: 50.1)			
LLaVA	64.9	**67.4**	66.0
CLIP (binary)	52.2	58.0	55.2
CLIP (continuous)	52.8	53.6	60.6
Kiki-Bouba (Prior Works: 69.3)			
LLaVA	83.65	**86.65**	85.65
CLIP (binary)	79.25	81.50	81.10
CLIP (continuous)	77.05	77.10	81.85

Table 5. Accuracy (%) of different classifier choices (columns) and models used to map verbalized features to numeric vectors (rows). LLaVA and CLIP are utilized as mapping models to map an image to a feature vector.

compared to LLaVA, CLIP offers a notable advantage in terms of inference speed. This is because CLIP supports batch-wise operations when computing similarities between a set of images and a set of verbalized features, which offers our method the flexibility to scale with a larger dataset or an increasing number of object classes. It is worth noting that using CLIP as the feature mapping model still allows our method to surpass previous state-of-the-art results by significant margins, achieving improvements of 8% and 12% on the iNaturalist and Kiki-Bouba datasets, respectively. This highlights the universal applicability of VRL-extracted features, regardless of the feature mapping approach used.

4.6. Qualitative Analysis

Figure 3 presents qualitative examples of what verbalized features are extracted from image pairs. We list some of the features that have top-5 feature importance in our trained logistic regression model. For inter-class differences (Fig. 3 (a)), verbalized features are used to capture distinguishing characteristics between different species. For example, the Lichen (Fungi) category is distinguished based on the structure of the thallus, where one image describes a flat, crust-like thallus, and the other describes a complex, branch-like thallus. Similarly, in the Wild Rye (Grass) category, the differences between a slender stem with a feathery inflorescence and a more robust stem with a dense, elongated seed head are verbalized. For intra-class commonality (Fig. 3 (b)), verbalized features are employed to capture shared traits within the same species. For example, the leaves of Manzanita (Berry) are consistently described as "ovate with a pointed tip and slightly serrated edges," while the inflorescences of Bulrush (Herb) are uniformly described as having a "delicate, feathery appearance." These verbalized features are used to guide the decision process by helping the model focus on important distinguishing features or shared traits within each class. Due to page limit, see Appendix B.5 for human study on the learned verbalized features.

5. Conclusion

In this paper, we propose Verbalized Representation Learning (VRL), which enables automatic interpretable feature extraction with few-shot samples. By leveraging VLMs, VRL generates verbalized features that capture both inter-class differences and intra-class commonalities. Our method not only enhances the model's adaptability with limited data but also provides transparency in the decision-making process, enabling easier interpretation of the features that influence predictions. Our experiments show that VRL outperforms prior approaches, achieving superior results while using significantly less data. This includes tasks like fine-grained recognition and novel concept adaptation, demonstrating its potential for real-world applications.

Acknowledgements

We thank anonymous reviewers and other members of UCLA-NLP+ group for their helpful comments. This work was partially supported by U.S. DARPA ECOLE Program No. #HR00112390060, ONR grant N00014-23-1-2780, DARPA ANSR program FA8750-23-2-0004, and Apple. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of DARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 1

[2] Morris Alper and Hadar Averbuch-Elor. Kiki or bouba? sound symbolism in vision-and-language models. *Advances in Neural Information Processing Systems*, 36, 2024. 2, 5

[3] David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. Network dissection: Quantifying interpretability of deep visual representations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6541–6549, 2017. 2

[4] David Bau, Jun-Yan Zhu, Hendrik Strobelt, Agata Lapedriza, Bolei Zhou, and Antonio Torralba. Understanding the role of individual units in a deep neural network. *Proceedings of the National Academy of Sciences*, 117(48):30071–30078, 2020. 2

[5] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European conference on computer vision (ECCV)*, pages 132–149, 2018. 3

[6] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in neural information processing systems*, 33:9912–9924, 2020. 2, 3, 4

[7] Chaofan Chen, Oscar Li, Daniel Tao, Alina Barnett, Cynthia Rudin, and Jonathan K Su. This looks like that: deep learning for interpretable image recognition. *Advances in neural information processing systems*, 32, 2019. 2

[8] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020. 2, 3, 4

[9] Wentao Chen, Chenyang Si, Zhang Zhang, Liang Wang, Zilei Wang, and Tieniu Tan. Semantic prompt for few-shot image recognition. *arXiv preprint arXiv:2303.14123*, 2023. 4

[10] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15750–15758, 2021. 2, 3, 4

[11] Yangyi Chen, Xingyao Wang, Hao Peng, and Heng Ji. Solo: A single transformer for scalable vision-language modeling. In *Transactions on Machine Learning Research*, 2024. 2

[12] Mia Chiquier, Utkarsh Mall, and Carl Vondrick. Evolving interpretable visual classifiers with large language models. *arXiv preprint arXiv:2404.09941*, 2024. 1, 2, 5, 6, 7

[13] Fahim Dalvi, Nadir Durrani, Hassan Sajjad, Yonatan Belinkov, Anthony Bau, and James Glass. What is one grain of sand in the desert? analyzing individual neurons in deep nlp models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 6309–6317, 2019. 2

[14] Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Tianyu Liu, et al. A survey on in-context learning. *arXiv preprint arXiv:2301.00234*, 2022. 5

[15] Ali Farhadi, Ian Endres, Derek Hoiem, and David Forsyth. Describing objects by their attributes. In *2009 IEEE conference on computer vision and pattern recognition*, pages 1778–1785. IEEE, 2009. 1, 2

[16] Vittorio Ferrari and Andrew Zisserman. Learning visual attributes. *Advances in neural information processing systems*, 20, 2007. 1

[17] Andrea Frome, Greg S Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Marc'Aurelio Ranzato, and Tomas Mikolov. Devise: A deep visual-semantic embedding model. *Advances in neural information processing systems*, 26, 2013. 2

[18] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020. 2, 3, 4

[19] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020. 2, 3, 4

[20] Lisa Anne Hendricks, Ronghang Hu, Trevor Darrell, and Zeynep Akata. Grounding visual explanations. In *Proceedings of the European conference on computer vision (ECCV)*, pages 264–279, 2018. 2

[21] Evan Hernandez, Sarah Schwettmann, David Bau, Teona Bagashvili, Antonio Torralba, and Jacob Andreas. Natural language descriptions of deep visual features. In *International Conference on Learning Representations*, 2021. 2

[22] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021. 5

[23] Shaoli Huang, Zhe Xu, Dacheng Tao, and Ya Zhang. Part-stacked cnn for fine-grained visual categorization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1173–1182, 2016. 2

[24] Andrej Karpathy, Justin Johnson, and Li Fei-Fei. Visualizing and understanding recurrent networks. *arXiv preprint arXiv:1506.02078*, 2015. 2

[25] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Advances in neural information processing systems*, 33:18661–18673, 2020. 3

[26] Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. In *International conference on machine learning*, pages 5338–5348. PMLR, 2020. 2

[27] Christoph H Lampert, Hannes Nickisch, and Stefan Harmeling. Attribute-based classification for zero-shot visual object categorization. *IEEE transactions on pattern analysis and machine intelligence*, 36(3):453–465, 2013. 2

[28] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024. 2, 4, 5

[29] Manling Li, Ruochen Xu, Shuohang Wang, Luowei Zhou, Xudong Lin, Chenguang Zhu, Michael Zeng, Heng Ji, and Shih-Fu Chang. Clip-event: Connecting text and images with event structures. In *Proc. Conference on Computer Vision and Pattern Recognition (CVPR2022)*, 2022. 2

[30] Fan Liu, Wenwen Cai, Jian Huo, Chuanyi Zhang, Delong Chen, and Jun Zhou. Making large vision language models to be good few-shot learners. *arXiv preprint arXiv:2408.11297*, 2024. 4

[31] Mushui Liu, Fangtai Wu, Bozheng Li, Ziqian Lu, Yunlong Yu, and Xi Li. Envisioning class entity reasoning by large language models for few-shot learning. *arXiv preprint arXiv:2408.12469*, 2024. 4

[32] Sachit Menon and Carl Vondrick. Visual classification via description from large language models. *arXiv preprint arXiv:2210.07183*, 2022. 1, 2, 5, 6

[33] Eduardo Navarrete, Michele Miozzo, and Francesca Peressotti. Language can shape the perception of oriented objects. *Scientific reports*, 10(1):8409, 2020. 1, 3

[34] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 2

[35] OpenAI. ChatGPT. 2022. 5

[36] Dong Huk Park, Lisa Anne Hendricks, Zeynep Akata, Anna Rohrbach, Bernt Schiele, Trevor Darrell, and Marcus Rohrbach. Multimodal explanations: Justifying decisions and pointing to the evidence. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8779–8788, 2018. 2

[37] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011. 3

[38] Sarah Pratt, Ian Covert, Rosanne Liu, and Ali 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*, pages 15691–15701, 2023. 1, 2

[39] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 1, 4

[40] Vilayanur S Ramachandran and Edward M Hubbard. Synaesthesia–a window into perception, thought and language. *Journal of consciousness studies*, 8(12):3–34, 2001. 5

[41] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. In *International conference on learning representations*, 2017. 5

[42] Bernardino Romera-Paredes and Philip Torr. An embarrassingly simple approach to zero-shot learning. In *International conference on machine learning*, pages 2152–2161. PMLR, 2015. 2

[43] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017. 2

[44] Tamar Rott Shaham, Sarah Schwettmann, Franklin Wang, Achyuta Rajaram, Evan Hernandez, Jacob Andreas, and Antonio Torralba. A multimodal automated interpretability agent. In *Forty-first International Conference on Machine Learning*, 2024. 2

[45] Chenming Shang, Shiji Zhou, Hengyuan Zhang, Xinzhe Ni, Yujiu Yang, and Yuwang Wang. Incremental residual concept bottleneck models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11030–11040, 2024. 2

[46] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023. 5

[47] Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. The inaturalist species classification and detection dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8769–8778, 2018. 2, 5

[48] An Yan, Yu Wang, Yiwu Zhong, Chengyu Dong, Zexue He, Yujie Lu, William Yang Wang, Jingbo Shang, and Julian McAuley. Learning concise and descriptive attributes for visual recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3090–3100, 2023. 1, 2

[49] Cheng-Fu Yang, Yao-Hung Hubert Tsai, Wan-Cyuan Fan, Russ R Salakhutdinov, Louis-Philippe Morency, and Frank Wang. Paraphrasing is all you need for novel object captioning. *Advances in Neural Information Processing Systems*, 35:6492–6504, 2022.

[50] Yue Yang, Artemis Panagopoulou, Shenghao Zhou, Daniel Jin, Chris Callison-Burch, and Mark Yatskar. Language in a bottle: Language model guided concept bottlenecks for interpretable image classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19187–19197, 2023. 1, 5, 6

[51] Mert Yuksekgonul, Maggie Wang, and James Zou. Post-hoc concept bottleneck models. *arXiv preprint arXiv:2205.15480*, 2022. 2

[52] MD Zeiler. Visualizing and understanding convolutional networks. In *European conference on computer vision/arXiv*, 2014. 2

[53] Lianmin Zheng, Liangsheng Yin, Zhiqiang Xie, Chuyue Sun, Jeff Huang, Cody Hao Yu, Shiyi Cao, Christos Kozyrakis, Ion Stoica, Joseph E Gonzalez, et al. Sglang: Efficient execution of structured language model programs. *arXiv preprint arXiv:2312.07104*, 2023. 1