Will It Zero-Shot?: Predicting Zero-Shot Classification Performance For Arbitrary Queries

Anonymous

***Abstract*—Vision-Language Models like CLIP create aligned embedding spaces for text and images, making it possible for anyone to build a visual classifier by simply naming the classes they want to distinguish. However, a model that works well in one domain may fail in another, and non-expert users have no straightforward way to assess whether their chosen VLM will work on their problem. We build on prior work using text-only comparisons to evaluate how well a model works for a given natural-language task, and explore approaches that also generate synthetic images relevant to that task to evaluate and refine the prediction of zero-shot accuracy. We show that generated imagery to the baseline text-only scores substantially improves the quality of these predictions. Additionally, it gives a user feedback on the kinds of images that were used to make the assessment. Experiments on standard CLIP benchmark datasets demonstrate that the image based approach helps users predict, without any labeled examples, whether a VLM will be effective for their application.**

***Index Terms*—vision-language models, zero-shot learning, multimodal AI**

I. Introduction

Large-scale vision-language models (VLMs) that integrate text and imagery, such as CLIP [1], are accelerating the development of many types of visual algorithms across numerous application domains. These models are appealing because their behavior can be specified using natural language, making them accessible even to users without deep computer vision expertise. Creating classifiers simply by naming the classes that need to be distinguished—without any training or fine-tuning on new data—lowers the barrier to entry for visual classification. The initial CLIP paper showed that zero-shot performance can beat the state of the art across many domains. But just because a VLM can be defined easily with class names and works well in some cases, it doesn't guarantee effectiveness for every classification problem.

Approaches to evaluating VLM effectiveness for specific problems were recently proposed using a method comparing different VLM models to select the best model. The language only visual model selection (LOVM) [2] work predicts how well a collection of VLM models will perform on an arbitrary zero-shot task; their approach starts with a baseline prediction for each model based on measuring model accuracy on imagenet, and then modifies that baseline score based on how well it classifies generated captions (used as a proxy for images). By comparing many models this way, LOVM can pick the best VLM for a task given only a text description (e.g., distinguishing similar butterflies). The primary motivation for LOVM is to select among a family of models to choose the best one for a problem. We consider the scenario where users have a specific model, and need to answer the question: "For the model I have access to, how well will it work on my task?". Figure 1 shows motivating examples; across a wide variety of problem domains, subject matter experts may want to deploy zero-shot learning approaches and need to know if a model is likely to be effective.

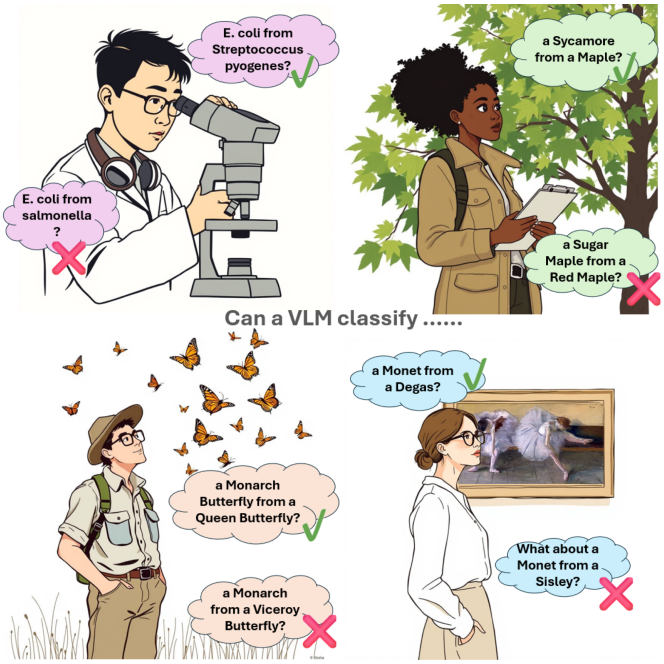


Fig. 1: Vision-Language Models (VLMs) enable users without machine learning expertise to rapidly deploy visual classifiers. These users need practical tools to determine if a VLM will effective for their tasks. This paper introduces a simple approach for predicting zero-shot classification performance by analyzing the internal consistency of CLIP's text and image embedding spaces.

Let's consider one possible task: "I have a picture of a butterfly, is it a Monarch or a Viceroy?". VLMs may struggle with such a task for several reasons: (a) both the words "Monarch" and "Viceroy" have meanings unrelated to the butterfly, (b) Monarch and Viceroy butterflies are quite similar in appearance, (c) the data that the VLM is trained on may itself be mis-labeled about these classes. These are reasons why

a user might want to ask if a given model would be effective for their task. But making specific accuracy predictions may also be hard based on task descriptions: the user might have in mind butterflies pinned in a case where they can take well lit, high-resolution pictures, or they may have in mind smartphone images captured in scenarios with bad lighting and motion blur.

In this paper, we try to realistically address this scenario. First, we improve the baseline approach to understanding how well a model understands concepts related to a given task using text captions as image proxies. We demonstrate that evaluating VLM performance on task-relevant synthetic images improves that understanding compared to text only analysis, with evaluation across many datasets from the CLIP benchmark suite. However comparisons to existing datasets are not useful to users so we also built a user interface that can accept natural language queries and show users the generated images used in our calculations. Showing the user the synthetic images allows them to iterate over their queries to make them more specific while also confirming that the images look like the data they intend to use in their application. This precision and model feedback is not necessarily possible with text as image proxies. Our specific contributions are:

- an improved algorithmic approach for predicting zero-shot VLM accuracy on a given class using only the class description;
- a quantification of the performance advantage from using generated imagery in addition to text for prediction accuracy across CLIP benchmark datasets;
- a web interface that deploys this zero-shot accuracy prediction approach, enabling users to compare models and refine queries without needing labeled data.

II. Related Work

a) Zero-Shot Learning: Zero-shot learning (ZSL) seeks to classify images into classes that are described in some way, but for which no training data is provided. The foundational work by [3] introduced and defined the ZSL problem. Early efforts to solve this problem involved the identification of attribute vectors and composing attributes by linear combinations of those vectors, as demonstrated in studies [4]–[6]. These attribute vectors were then applied in the inference phase to facilitate class prediction. With the development of deep learning, non-linear approaches for extracting attributes from images became possible, and ZSL approaches started to be more successful [7]. Related zero-shot learning prediction methodologies to ours include LOVM [2] which attempted to create a benchmark for ranking VLMs for any given zero-shot task, and ModelGPT [8] which is a hypernetwork intended to generate tailored models faster than fine-tuning with just a user description.

On predicting zero-shot and open-vocabulary performance of VLMs, Zero-Shot Out-of-Distribution (OOD) Detection [9] uses a set of generated seen and unseen labels to develop a score that predicts if a sample image is more similar to the unseen labels than the set of seen labels by doing cosine similarity for all labels and the image embedding. These unseen labels are generated by a pre-trained image description generator that takes the CLIP image embedding as an input and outputs labels that are closer to the image in CLIP space.

b) CLIP: Vision-language models (VLMs) like the Contrastive Language–Image Pre-training (CLIP) model [1] try to unify image and text space. Formally, CLIP tries to optimize an image encoder to map images into a set of features I, and a text encoder to map text into features T, maximizing the conditional log-probabilities for images and captions that are associated. A huge number of pairs of images and their associated captions are leveraged to train this model to have aligned image and text representations.

The CLIP framework can be used for classification, where the class name serves as the text prompt (e.g., "an image of a 2003 Corvette"). In the standard approach, the predicted class for an image is determined by finding the class with the highest similarity between its text embedding and the embedding for the image. Building on this, [10] improved the capabilities of VLMs for zero-shot learning by utilizing more descriptive prompts rather than class names. By searching external vision-language databases with CLIP, CaSED [11] tries to solve the image zero-shot task without actual class names. Tip-Adapter [12] enhances CLIP's performance in few-shot settings while preserving the training-free advantage of zero-shot classification. There has also been work to improve zero-shot classification performance by improving the CLIP training data quality, including [13] which used a large language model to rewrite the text descriptions for each CLIP training image.

Beyond these improvements in using CLIP for zero and few-shot learning, there are a large number of extensions of CLIP (as well as other VLMs) that support visual question answering and captioning, like SigLIP [14], FLAVA [15], and tools like DALL-E and its successors [16] which focus on image generation.

c) VLM Modality Gap: In order to be able to accurately predict whether CLIP represents a concept well or not, it is necessary to understand how the CLIP embedding space is organized. While the CLIP training objective tries to align related image and text representations, empirical observations from [17] and [18] reveal a persistent modality gap within the shared latent space, where the image representations for a particular concept and the text representations for that concept are well clustered in their respective embedding spaces, but not well aligned across the embedding spaces.

The existence of this modality gap suggests that approaches to evaluating whether CLIP understands a visual concept cannot be based simply on a threshold of how similar images and text descriptions of those images are. Our proposed approach instead analyzes the internal consistency of the image and text model representations. The approach does not require labeled data, making it accessible to a variety of real-world end users, use cases, and problem domains.

III. METHOD

In this section, we describe our method to predict how well a CLIP-based model can classify an arbitrary class or concept defined in natural language. First, for a query class or concept, we create a list of related classes or concepts. This can be generated in several ways: provided by the end user (e.g., a medical team may already have a list of image types they care about); sourced from WordNet synsets; or generated by a large language model prompted with the query to generate related classes or concepts. For instance, in a medical imaging context, we might be interested in how way a CLIP-variant classifies "early-stage tumors." The related concepts for that class might be "benign growths" or "inflammation." The ideal model would be able to differentiate between these concepts, while models less suited to the task would not be.

Next, we consider the query text. Intuitively, if the VLM "understands" a concept, it should accurately distinguish examples of this query class from related but incorrect classes. While we don't have varying real-world examples of the query class – it is only expressed in natural language – we can simulate examples by using image and text generation tools to create diverse examples of the class.

This idea draws inspiration from cycle consistency losses used in unsupervised learning, which ensure that transformations applied to data can be reversed with minimal information loss. If we generate example images using a CLIP-based image generator, such as Stable Diffusion (which was trained using the OpenCLIP-ViT/H text-encoder) [19], map that image into the CLIP image embedding space, and then find similar features in the text embedding space, we expect that the textual description most closely related to the original query should be among the top matches. This consistency between the generated images and the text embeddings reflects the model's ability to correctly capture the essence of the query. If the model is truly capable of understanding the specific concept, such as "early-stage tumor," the generated images should map closely to relevant textual descriptions, and unrelated or incorrect classes (like "benign growth" or "inflammation") should be less similar in the embedding space. By leveraging this cycle of generating and embedding, we can effectively gauge the model's accuracy in interpreting and distinguishing the specified concept, even in the absence of labeled data.

We explore various image- and text-based scoring methods and analyze their correlation with true zero-shot accuracies on established labeled datasets. The sections below detail each step in this process.

This approach is largely targeted for scenarios where labeled data is unavailable, difficult to obtain, or costly to generate. For well-labeled datasets, it is possible to simply measure the zero-shot classification accuracy. However, in many real-world applications, end users may need to determine whether there exists a model that can support their task before investing in annotation efforts. Additionally, the way a classification problem is framed in natural language can significantly impact a model's performance. Subtle differences in wording – such as describing a category as "early-stage tumor" versus "pre-cancerous growth" – can lead to variations in how well the model distinguishes between relevant concepts. By enabling users to explore different articulations of their task and assess how the model responds, our method provides a powerful tool for refining label definitions, even in already labeled datasets.

A. Models

In this paper, we evaluate our proposed methods on various Vision-Language Models (VLMs) inspired by the 'CLIP' framework, which combines an image encoder and a text encoder to align visual and textual representations within a shared embedding space. Specifically, we consider:

- **CLIP** [1]: The most popular initial VLM, integrating an image encoder and a text encoder. CLIP is trained to align image and text representations within a shared embedding space using a softmax-based contrastive learning loss.
- **SigLIP** [14]: An adaptation of the CLIP framework, but trained with a sigmoid-based loss instead of softmax, improving robustness to out-of-distribution data and long-tail categories.
- **FLAVA** [15]: A multi-modal framework that integrates an image encoder, a text encoder, and a multi-modality encoder designed for joint vision-language understanding.

B. Datasets

While our proposed approach is designed for use cases with no labeled training data, we use labeled training data for evaluation purposes. With the intent to cover a broad range of image classes and settings, we selected 9 CLIP-benchmark datasets, as well as CUB-200 [20] for more fine-grained classes. The CLIP-benchmark datasets that were selected are ImageNet [21] and ObjectNet [22], for their scale; CIFAR100 [23], for its focus on small images; Food101 [24], FGVC-Aircraft [25], Flowers-102 [26], Stanford Cars [27], and Oxford-IIIT Pets [28] for their fine-grained classes; and RESISC-45 [29], for its focus on non-standard (satellite) imaging.

C. Generated Images

For a given natural language query, we use SDXL-Lightning [30] trained on LAION-5B [31] to generate twenty 1024×1024 pixel synthetic images of the class. The prompt for the image generation is generated by GPT-4o to be a realistic caption of an image of a {class_name} from the {domain}. For example, the domain for CUB-200 is 'birds'. We first generated images with the simple prompt of "a photo of a {class_name}, a type of {domain}", but found that the generated captions yielded more realistic and diverse pictures, whereas the same image prompt yielded very similar images. For a new query from an unknown domain, the user could specify the domain or it could be generated by an LLM (an approach we use on our web interface discussed below). Each image's corresponding text label is "a photo of a {class_name}".

D. Scores

After generating the text descriptions the list of related classes, and the text and/or images for the query, we use CLIP-ViT-B/32 (https://huggingface.co/openai/clip-vit-base-patch32) to compute feature embeddings, and consider different scores that attempt to capture relevant information for predicting the zero-shot accuracy of the CLIP model. We describe each of the different scores in depth below.

a) Consistency Score: The first score we propose uses the generated images and compares their embeddings to the text embeddings for the list of related classes. The point of this score is to measure the consistency of embedding shifts for image and text embeddings of one class to a similar neighbor class. To get a single score for class i, we minimize over all other classes, to focus on the most inconsistent (confusing) alternative class. Specifically, we define a consistency score, s_k:

$$s_k = \min_{j \neq i} \cos\left(I_{i_k} - \overline{I}_j, T_i - T_j\right) \tag{1}$$

where I_{i_k} represents the image embedding of the k-th generated image belonging to class i, while $\overline{I}_j$ denotes the mean vector of all generated image embeddings for class j. We define $S_{i_{sil}}$ to be the consistency score for class i computed as the average over all the images in the class. Each difference computed represents the shift of either the text or image embeddings from class i to class k.

b) Multimodal Silhouette Score.: The second score we propose focuses on how classification is often a task of distinguishing a class from its nearest neighbor. This score takes inspiration from the silhouette score [32] which was used for the granularity scores by LOVM [2] while also updating it to account for the multimodal nature of vision-language models. The generated images are treated as a cluster in a class's image space. The multimodal silhouette score evaluates that cluster, its modality gap, and its distance from its nearest neighbor (weighted by $\lambda = 2.5$). For each class i, let $\{I_{i_k}\}_{k=1}^{N_i}$ be its images, and T_i its text embedding. We define the intra-class distances for each image I_{i_k} as:

$$a\big(I_{i_k}\big) = 1 - \cos\big(I_{i_k}, \overline{I}_i\big) + \lambda \Big[\, 1 - \cos\big(I_{i_k}, T_i\big)\Big], \tag{2}$$

and the inter-class distances as:

$$b\big(I_{i_k}\big) = \min_{j \neq i} \Big\{ \Big[\, 1 - \cos\big(I_{i_k}, \overline{I}_j\big)\Big] + \lambda \Big[\, 1 - \cos\big(I_{i_k}, T_j\big)\Big] \Big\}. \tag{3}$$

Here, λ balances how much we weigh image-text proximity versus image-image cohesion. The multimodal silhouette for I_{i_k} is then defined in a silhouette-like manner:

$$\mathrm{sil}_{\mathrm{mm}}\big(I_{i_k}\big) \;=\; \frac{b(I_{i_k}) - a(I_{i_k})}{\max\big\{\, a(I_{i_k}),\, b(I_{i_k})\big\}}$$

A larger value means I_{i_k} is well separated from other classes in both image and text spaces. We define $S_{i_{sil}}$ to be the multimodal silhouette score for class i computed as the average over all the images in the class. This score acts as a classification margin that is more sensitive to the the intra-class distances of a group of images. The final multimodal silhouette score and consistency scores are presented as a coefficients to evaluate the performance of the VLM on this specific class i.

c) Compound Score: Finally, we propose a compound approach that considers both the internal consistency of image and text embeddings as well as the intra- and inter-class distances. The compound score uses a scaling factor (α = 4) chosen to approximately scale the effect of the silhouette score to be the same as the consistency score. This score measures the consistency for every alternative class, and penalizes a small or negative silhouette score:

$$S_i \;=\; S_{i_{cs}} + \alpha S_{i_{sil}}.$$

d) Text-Based Scores: We also consider a purely text-based score that does not require the generation of any synthetic images. This method leverages detailed textual descriptions as proxies for images to eliminate the need for image generation. To formulate our text-only consistency score, we define two types of prompts:

1) **Plain Text:** "a photo of a {*class_name*}"
2) **Descriptive Text:** "a photo of a {*likely descriptive caption for class_name including the exact class_name*}"

Here, the descriptive text is generated using GPT-4o. For instance, for the class "Golden Retriever," examples of the descriptive text are "a photo of a large golden retriever with short, silky fur" and "a photo of a medium golden retriever with long, wavy fur." Using these descriptions allows us to generate diverse images for a given class. More examples can be found in the Appendix.

e) Text-Only Consistency Score.: To mirror the image-based consistency score, we replace synthetic images with descriptive text embeddings. For each class i, let $T_{i_k}^d$ be the CLIP embedding of the k-th descriptive text, and let T_i be the embedding of the plain text prompt (e.g., "a photo of a {`class_name`}"). We compute:

$$s_k^T = \min_{j \neq i} \cos\big(T_{i_k}^d - \overline{T}_j^d,\, T_i - T_j\big), \tag{4}$$

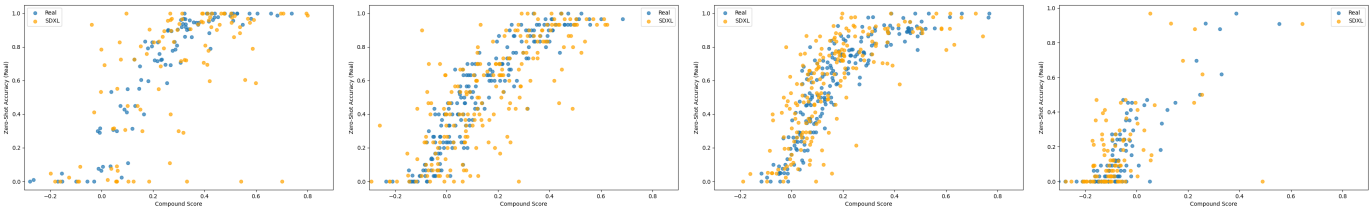
where $\overline{T}_j^d$ is the mean of the descriptive text embeddings for class j, and T_j is the plain text embedding for class j. We define $S_{i_{\mathrm{cs}}}^T$ to be the text consistency score for class i computed as the average over all the images in the class.

f) Text-Based Silhouette Score.: We also adapt the multimodal silhouette score for text-only data. For class i, let $\{T_{i_k}^d\}_{k=1}^{N_i}$ be the descriptive text embeddings, and let T_i be the plain text embedding. We define the intra-class term

$$a(T_{i_k}^d) \;=\; 1 - \cos\big(T_{i_k}^d, \overline{T}_i^d\big) \;+\; \lambda \Big[1 - \cos\big(T_{i_k}^d, T_i\big)\Big], \tag{5}$$

and the nearest inter-class term:

$$b(T_{i_k}^d) \;=\; \min_{j \neq i} \Big\{ \big[1 - \cos\big(T_{i_k}^d, \overline{T}_j^d\big)\big] \;+\; \lambda \Big[1 - \cos\big(T_{i_k}^d, T_j\big)\Big] \Big\}. \tag{6}$$



(a) Flowers-102 (b) CUB-200 (c) Stanford Cars (d) FGVC-Aircraft

Fig. 2: Each scatter plot shows the relationship between zero-shot classification accuracy on the real dataset and our compound score in real (blue) or generated (orange) data. Each dot represents a class in datasets.

where $\overline{T}_i^d$ is the mean of the descriptive text embeddings for class i, and $\lambda = 2.5$ balances plain-descriptive distances. Then the text-based silhouette is

$$\text{sil}_{\text{text}}(T_{i_k}^d) = \frac{b(T_{i_k}^d) - a(T_{i_k}^d)}{\max\{a(T_{i_k}^d),\, b(T_{i_k}^d)\}}, \tag{7}$$

We define $S_{i_{\text{sil}}}^T$ to be the text silhouette score for class i computed as the average over all the images in the class.

g) Text-Based Compound Score.: Finally, we combine the text-only consistency score and the text-based silhouette score:

$$S_i^T = S_{i_{\text{cs}}}^T + \alpha\, S_{i_{\text{sil}}}^T, \tag{8}$$

where we set $\alpha = 4$, so that its scale is matched to the image-based compound score.

IV. Experimental Results

In this section, we demonstrate the efficacy of our proposed method and report results using the ten labeled datasets discussed in the methods section for evaluation purposes.

A. Comparison of Scores and Zero-shot Accuracy

For each class in the datasets, we compute our image and text-based scores. We evaluate using labeled datasets, so we can report on the zero-shot accuracy metric for each of these datasets.

a) Image-Based Scores: Figure 2 shows the relationship between our compound score and the true accuracy for each class in four of the datasets. The y-axis represents the zero-shot classification accuracy on the real dataset, and the x-axis represents the compound score for each class. To gain a qualitative understanding of the reasonableness of the generated images, each class in each dataset is represented with two points. . Blue markers denote the scores computed based on embeddings of the real images in the test datasets. while orange markers denote generated images (that would be available in arbitrary circumstances). The figure shows a positive correlation between the classification accuracy and the scores computed with both the real image embeddings and the embeddings of the generated images, showing that the generated images serve as a good (but not perfect) proxy for the real images.

To quantify the relationship, we compute the Spearman's correlation coefficient between the scores for the generated images and the zero-shot accuracy performance on real datasets to assess how strongly our scores are related to the true zero-shot classification accuracy. Table I, shows the zero-shot classification accuracy for the real images in each of the datasets, when using the CLIP, SigLIP and FLAVA models. These are the actual accuracies for Zero-Shot classification with these VLMs on imagery from these standard datasets. The next three tables capture our prediction performance, how well our scores correlate with the per-class classification accuracy.

b) Compound Score vs. Zero-Shot: In Table IV, we compare our compound score with a baseline approach of simply measuring the zero-shot classification performance on the generated images. We observe that the compound score has higher correlation with the actual zero-shot learning accuracy of the real world data on a significant majority of the datasets, and also that computing these scores based on image embeddings is much better than computing these scores based on text embeddings. We believe the improvements from the compound score arise because the consistency score and silhouette score capture useful measurements of the image clustering and local organization of similar classes in the embeddings spaces, which generalizes better than the classification accuracy of the generated images.

For some classes, the real image classification variations between classes have low correlations with both the text and image based scores. For some datasets like Food and PETS, the zero-shot classification is very accurate for most classes, making the rank-based correlation that Spearman correlation not a stable measure. The other classes with very poor performance are in ObjectNet, which explicitly has images of objects that are more obscured in the scene and in unnatural positions (like a "rocking chair" image showing a small rocking chair laying sideways on a bed, taking up a small part of the image). In this case, correlations are low because synthetic images (which are reasonable images of tjhe class) are not representative of the images in the task.

c) Choice of image generator: Table III shows the correlation of our image-based scores for images generated using the DALL-E 3 image generator [16] (top), and the open-source SDXL-Lightning image generator [30] (bottom). We run this experiment on a subset of the datasets due to the cost (in both dollars and time) of the DALL-E 3 image generator. In all of our result tables, entries that have correlations greater than 0.5 are color-coded in green, with higher correlations having

TABLE I: Zero-Shot Classification Accuracy for Real Images

Dataset	CLIP	SigLIP	FLAVA
CUB	.52	.73	.45
Cars	.58	.93	.29
Aircraft	.17	.41	.12
Food	.82	.94	.79
ImageNet	.62	.79	.55
ObjectNet	.43	.65	.33
CIFAR	.64	.71	.64
Flowers	.65	.71	.64
Pets	.80	.92	.63
RESISC	.51	.61	.44

TABLE II: Spearman Correlations for Compound Score & Real Zero Shot Accuracy for Text and Image Methods (CLIP+SDXL-Lightning)

Dataset	Text Score		Image Score	
	Compound	Zero Shot	Compound	Zero Shot
CUB	.52	.33	.77	.57
Cars	.52	.27	.79	.74
Aircraft	.36	.32	.54	.55
Food	.34	.15	.25	.28
ImageNet	.29	.09	.50	.43
ObjectNet	.01	.01	.25	.25
CIFAR	-.10	-.04	.60	.56
Flowers	.60	.31	.60	.65
Pets	.37	.39	.44	.49
RESISC	.34	.25	.55	.59

TABLE III: Spearman Correlation for Image-based Scores (Model × Score × Image Generator)

	Generated with DALL-E 3		
Dataset	CLIP	SigLIP	FLAVA
CUB	.68	.72	.63
Aircraft	.55	.79	.59
CIFAR	.70	.63	.68
RESISC	.59	.35	.65
	Compound Score		
CUB	.77	.74	.73
Aircraft	.54	.66	.60
CIFAR	.60	.53	.59
RESISC	.55	.25	.51
	Generated with SDXL		

TABLE IV: Spearman Correlation for Image-based Scores

	Consistency			Silhouette			Compound		
Dataset	CLIP	SigLIP	FLAVA	CLIP	SigLIP	FLAVA	CLIP	SigLIP	FLAVA
CUB	0.73	0.73	0.62	0.75	0.68	0.70	0.77	0.74	0.73
Cars	0.69	0.54	0.65	0.79	0.59	0.81	0.79	0.60	0.80
Aircraft	0.57	0.56	0.40	0.49	0.62	0.62	0.54	0.66	0.60
Food	0.19	0.27	0.15	0.26	0.20	0.39	0.25	0.22	0.36
ImageNet	0.47	0.47	0.62	0.49	0.46	0.66	0.50	0.48	0.68
ObjectNet	0.19	0.08	0.21	0.25	0.21	0.35	0.25	0.16	0.34
CIFAR	0.55	0.43	0.53	0.60	0.57	0.59	0.60	0.53	0.59
Flowers	0.63	0.30	0.44	0.58	0.27	0.64	0.60	0.27	0.62
Pets	0.45	0.48	0.49	0.42	0.50	0.41	0.44	0.56	0.49
RESISC	0.40	0.27	0.27	0.53	0.23	0.54	0.55	0.25	0.51

brighter coloring. The figure shows that both image generation models give scores with reasonably high correlations. While Dall-E 3 images are often more compelling and realistic, the results for our purposes are similar. Unless otherwise noted, the remainder of the paper uses the SDXL-Lightning model.

Text-Based Scores.: Table VI shows the correlation for each of the different text-based scores, using each of the different models, across each of the datasets. These correlations are significantly lower than using the image-based approaches seen in Table IV, and in some cases are even negative. This suggests that simply using text prompts to ascertain how well a model can perform zero-shot learning for image datasets is insufficient.

d) Outlier Classes: While the tables above present aggregate information computed over all of the classes in a dataset, we can also investigate the performance for specific classes. In particular, there are some exceptional outlier classes, where our score and the actual accuracy diverge. These are points that are in the bottom-right or top-left corners of Figure 2.

One such outlier was the "Tornado" class from the Aircraft dataset, shown in Figure 3a, where the top row shows generated images and the bottom row shows real images. Although the real images had very high zero-shot classification accuracy, our prediction on the generated images was quite low. The reason for this is that the image generation process confused the meaning of Tornado the aircraft, with the natural disaster. This suggests a potential limitation of our method: it may be challenging to generate images for classes that have somewhat ambiguous meanings, but that is irrelevant to the classification performance for real world images of the class. This can, however, be mitigated by a user visually confirming that the generated images are acceptable and regenerating them if not, potentially using modified image generation prompts (e.g., 'a tornado, a type of aircraft').

Another interesting outlier class was "Apple Pie" from the Food-101 dataset, shown in Figure 3b. For this dataset, the generated apple pie pictures were all very clear depictions of the class and, since there are no other pies among the food classes, the apple pie class scores very well. The real apple pie pictures, on the other hand, score very poorly. When reviewing the images it is clear that the actual dataset images are more often deconstructed pastry plates, smaller handheld apple pastries, or just more focused on apple pie dish accessories than the actual pie itself. This is not an issue of the generated images not being realistic depictions of apple pie, but more of the real images having a much looser interpretation of what counts as apple pie. In this way our method can be used to evaluate the quality of class depictions comparatively to the

(a) Aircraft:"Tornado"

(b) Food-101:"Apple Pie"

Fig. 3: Example images from classes where the compound scores do not match the CLIP Zero-Shot Accuracy. These outliers can be caused by real dataset images being weaker depictions of the class than the generated images (a) or image generation misunderstanding the class (b).

generated images. This might be useful to someone wanting to evaluate the labels of a dataset or understand why some set on data does not work well for their retrieval queries.

V. Interactive Tool

We finally present a web accessible interactive tool, "Will CLIP Zero-Shot?" This tool demonstrates our methods by allowing users to input any natural language query and see our prediction of the zero-shot classification accuracy for that query with CLIP, using (we describe a simple approach to convert our scores to a prediction of classification accuracy in Appendix Section B). This interactive tool can be accessed at https://brave-sand-0df30360f.4.azurestaticapps.net. The tool works by taking the user's input query, and reasonable alternative classes that the user thinks could cause classification confusion. If the user does not know what alternatives to select, we provide an option to generate them using the following prompt to OpenAI's GPT-4o model to generate alternative labels: "Create 10 realistic alternatives to the following input label by suggesting alternatives that are somewhat similar. I don't want the same label reworded or a subclass of the input label. The given label is: {input}". The user can also input a domain so that the LLM can disambiguate the class more accurately. We generate images of the input query as well as the alternatives, compute the compound scores, and map the score into an accuracy prediction.

A. Case Study

There are many potential use cases for wanting to predict the zero-shot capabilities of a VLM on any arbitrary query. Here we consider a specific case study that highlights how it might be used — differentiating between species of related looking insect.

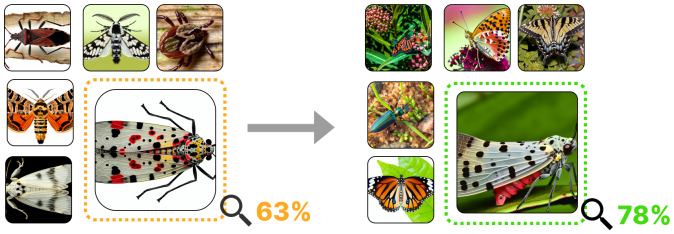


Fig. 4: Our tool predicts that CLIP would be able to correctly classify images of spotted lanternflies relative to images of similar-looking insects about 63% predicted accuracy. However, when the alternative classes are endangered and other threatened species in the northeast United States, CLIP is predicted to correctly classify a spotted lanternfly relative to the other insects about with 78% predicted accuracy.

The spotted lanternfly is an invasive species northeast United States which threatens to devastate local vegetation [33]. The USDA recommends killing, photographing and reporting the insect [34]. But how easy is it to automatically recognize these insects? Our tool suggests that CLIP has reasonable accuracy in differentiating Spotted Lanternflies from similar looking insects. Figure 4 (left) shows the spotted lanternfly compared to five similar looking insects.

An alternative app, however, might seek to ensure that someone does not accidentally kill endangered insects. Figure 4 (right) shows our tool predicting that CLIP can differentiate Spotted Lanternflies from these endangered species even more accurately. This shows how different users might iterate over queries to get more of less specific depending on their use case. They could also specify the habitat of each insect for the photos making the classification problem even more specific. Most importantly, they are getting feedback from the model in the form of the predicted accuracy and the images which add context to how our method got to its predicted accuracy, potentially increasing user trust in the system.

VI. Conclusions

In this paper, we presented an approach to understand how well a VLM model will perform on a users zero-shot classification task. We believe this approach can have practical benefits for non-experts deciding if a VLM is an appropriate tool for their problem domain. Our work shows that there is substantial benefit gained from analyzing the embeddings of artificially generated imagery related to the task.

We also find that there are a few causes where our scores are not good predictors of performance. The most interesting of these are cases like ObjectNet, where the image data is not well described by a text description like "an image of a rocking chair". Future work should explore effective approaches to allow the user to describe more about the context and background and circumstances of their image in order, and then take that information into account when creating a prediction.

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 *Int. Conf. Mach. Learn.*, 2021, pp. 8748–8763.

[2] O. Zohar, S.-C. Huang, K.-C. Wang, and S. Yeung, "Lovm: Language-only vision model selection," 2023. [Online]. Available: https://arxiv.org/abs/2306.08893

[3] C. H. Lampert, H. Nickisch, and S. Harmeling, "Learning to detect unseen object classes by between-class attribute transfer," in *IEEE Conf. Comput. Vis. Pattern Recog.* IEEE, 2009, pp. 951–958.

[4] B. Romera-Paredes and P. Torr, "An embarrassingly simple approach to zero-shot learning," in *International conference on machine learning*. PMLR, 2015, pp. 2152–2161.

[5] E. Kodirov, T. Xiang, and S. Gong, "Semantic autoencoder for zero-shot learning," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017, pp. 3174–3183.

[6] Y. Fu, T. M. Hospedales, T. Xiang, and S. Gong, "Transductive multi-view zero-shot learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 11, pp. 2332–2345, 2015.

[7] W. Wang, Y. Pu, V. Verma, K. Fan, Y. Zhang, C. Chen, P. Rai, and L. Carin, "Zero-shot learning via class-conditioned deep generative models," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, 2018.

[8] Z. Tang, Z. Lv, S. Zhang, F. Wu, and K. Kuang, "Modelgpt: Unleashing llm's capabilities for tailored model generation," 2024. [Online]. Available: https://arxiv.org/abs/2402.12408

[9] S. Esmaeilpour, B. Liu, E. Robertson, and L. Shu, "Zero-shot out-of-distribution detection based on the pre-trained model clip," *AAAI*, vol. 36, no. 6, p. 6568–6576, Jun. 2022. [Online]. Available: http://dx.doi.org/10.1609/aaai.v36i6.20610

[10] S. Pratt, I. Covert, R. Liu, and A. Farhadi, "What does a platypus look like? generating customized prompts for zero-shot image classification," in *Int. Conf. Comput. Vis.*, 2023, pp. 15 691–15 701.

[11] A. Conti, E. Fini, M. Mancini, P. Rota, Y. Wang, and E. Ricci, "Vocabulary-free image classification," *Adv. Neural Inform. Process. Syst.*, vol. 36, pp. 30 662–30 680, 2023.

[12] R. Zhang, W. Zhang, R. Fang, P. Gao, K. Li, J. Dai, Y. Qiao, and H. Li, "Tip-adapter: Training-free adaption of clip for few-shot classification," in *Eur. Conf. Comput. Vis.* Springer, 2022, pp. 493–510.

[13] L. Fan, D. Krishnan, P. Isola, D. Katabi, and Y. Tian, "Improving clip training with language rewrites," *Adv. Neural Inform. Process. Syst.*, vol. 36, 2024.

[14] X. Zhai, B. Mustafa, A. Kolesnikov, and L. Beyer, "Sigmoid loss for language image pre-training," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 11 975–11 986.

[15] A. Singh, R. Hu, V. Goswami, G. Couairon, W. Galuba, M. Rohrbach, and D. Kiela, "Flava: A foundational language and vision alignment model," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022, pp. 15 638–15 650.

[16] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, and I. Sutskever, "Zero-shot text-to-image generation," in *International conference on machine learning*, 2021, pp. 8821–8831.

[17] C. Zhou, F. Zhong, and C. Oztireli, "Clip-pae: Projection-augmentation embedding to extract relevant features for a disentangled, interpretable and controllable text-guided face manipulation," in *ACM SIGGRAPH 2023 Conference Proceedings*, 2023, pp. 1–9.

[18] W. Liang, Y. Zhang, Y. Kwon, S. Yeung, and J. Zou, "Mind the gap: Understanding the modality gap in multi-modal contrastive representation learning," in *Adv. Neural Inform. Process. Syst.*, 2022.

[19] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022, pp. 10 684–10 695.

[20] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona, "Caltech-UCSD Birds 200," California Institute of Technology, Tech. Rep. CNS-TR-2010-001, 2010.

[21] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2009, pp. 248–255.

[22] A. Barbu, D. Mayo, J. Alverio, W. Luo, C. Wang, D. Gutfreund, J. Tenenbaum, and B. Katz, "Objectnet: A large-scale bias-controlled dataset for pushing the limits of object recognition models," in *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[23] A. Krizhevsky, "Learning multiple layers of features from tiny images," University of Toronto, Tech. Rep., 2009.

[24] L. Bossard, M. Guillaumin, and L. Van Gool, "Food-101 – mining discriminative components with random forests," in *European Conference on Computer Vision*, 2014.

[25] S. Maji, E. Rahtu, J. Kannala, M. Blaschko, and A. Vedaldi, "Fine-grained visual classification of aircraft," *arXiv preprint arXiv:1306.5151*, 2013.

[26] M.-E. Nilsback and A. Zisserman, "Automated flower classification over a large number of classes," in *Indian Conference on Computer Vision, Graphics and Image Processing*, Dec 2008.

[27] J. Krause, M. Stark, J. Deng, and L. Fei-Fei, "3d object representations for fine-grained categorization," in *2013 IEEE International Conference on Computer Vision Workshops*, 2013, pp. 554–561.

[28] O. M. Parkhi, A. Vedaldi, A. Zisserman, and C. V. Jawahar, "Cats and dogs," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2012.

[29] G. Cheng, J. Han, and X. Lu, "Remote sensing image scene classification: Benchmark and state of the art," *Proceedings of the IEEE*, vol. 105, no. 10, pp. 1865–1883, 2017.

[30] S. Lin, A. Wang, and X. Yang, "Sdxl-lightning: Progressive adversarial diffusion distillation," *arXiv preprint arXiv:2402.13929*, 2024.

[31] 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. 25 278–25 294, 2022.

[32] P. J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *Journal of Computational and Applied Mathematics*, vol. 20, pp. 53–65, 1987.

[33] S. Kundurthy, "Lantern-rd: Enabling deep learning for mitigation of the invasive spotted lanternfly," *arXiv preprint arXiv:2205.06397*, 2022.

[34] USDA Animal and Plant Health Inspection Service, "Spotted lanternfly (slf)," 2025, accessed: 30-Jan-2025. [Online]. Available: https://www.aphis.usda.gov/plant-pests-diseases/slf

APPENDIX

A. Image Captioning

We generated image captions for each class in all datasets for use in image and text experiments. Our captioning approach involved taking the list of class names and the category of a dataset and using that information to prompt GPT-4o to come up with descriptive image captions that we can use to generate diverse realistic images. Our generation is a multi-stage process that gets descriptive content about about a class to generate a realistic, descriptive, and diverse caption then generate the image. Our prompting is as follows. 1st stage: "You are an AI assistant that generates creative and diverse image captions suitable for use with image generation models like DALL-E. Given a subject, provide num_captions distinct, diverse and descriptive captions, considering the following global taxonomical traits when generating captions: {global_traits}." 2nd stage: "Please generate num_captions diverse and creative alternative captions for the subject 'subject'. Each caption should be compatible with the CLIP model so your caption should share the same prefix with the original prompt template provided: 'prompt_template'. An example can be, the template is 'a photo of a c' and the descriptive caption is 'a photo of a c, [descriptive content]'. Here c is the class name of the subject."

We find that images generated with the same non-descriptive class name prompt are considerably less diverse and look less realistic or only emulate one realistic setting. An example of this can be seen in Figure 5 with the class "Woman". The generative caption images show many diverse scenarios and characters that are all women, whereas the simply generated images ("a photo of a woman") almost look like the same image multiple times.

B. Converting Scores to Accuracy Predictions

Our Experimental Results show that the scores above correlate with zero-shot classification accuracy but do not directly predict it. On our interactive website, however, we want to give end users a prediction of accuracy rather than correlations. In order to do this, we use beta regression to transform predicted scores into accuracy values.

Given N datasets, let $\mathbf{S}_i$ and $\mathbf{A}_i$ represent the scores and true accuracies for dataset i. $\mathbf{S}_i$ may include any relevant scores. We evaluate using leave-one-out cross-validation, training on $N-1$ datasets and testing on the left-out dataset.

In the beta regression model, accuracy $A_j \in (0,1)$ for class j follows a beta distribution:

$$A_j \sim \text{Beta}(\alpha_j, \beta_j),$$

where α_j and β_j are functions of the score S_j:

$$\log(\alpha_j) = \mathbf{X}_j \cdot \boldsymbol{\theta}_1, \quad \log(\beta_j) = \mathbf{X}_j \cdot \boldsymbol{\theta}_2.$$

Here, $\mathbf{X}_j$ is a feature vector derived from S_j, and $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2$ are regression parameters.

Class	Captions
Woman	A photo of the small woman exploring antique shops on a cobblestone street.
Woman	A photo of the big woman effortlessly lifting weights in a gym.
Woman	A photo of a woman wearing a traditional dress at a cultural festival.
Woman	A photo of a woman hiking on a scenic mountain trail with her friend.
Woman	A mysterious photo of a woman in a vintage dress standing in an old mansion.
Black footed Albatross	A photo of a black footed Albatross, soaring gracefully above the ocean waves with its long wings spread wide.
Black footed Albatross	A photo of a black footed Albatross, displaying its striking black and white plumage against the backdrop of a clear blue sky.
Black footed Albatross	A photo of a black footed Albatross, nestled on a rocky cliff during the breeding season, showcasing its fluffy chick.
Black footed Albatross	A photo of a black footed Albatross, expertly gliding over the surface of the water, its keen eyes scanning for fish.
Black footed Albatross	A photo of a black footed Albatross, with its uniquely shaped beak, ready to dive for its next meal in the vast sea.
Lasagna	A photo of a lasagna, a classic Italian dish layered with rich tomato sauce, creamy béchamel, and melted mozzarella.
Lasagna	A photo of a lasagna, a comfort food favorite served hot, oozing with gooey cheese and savory meat.
Lasagna	A photo of a lasagna, a delicious baked pasta dish with layers of succulent ground beef and tangy marinara sauce.
Lasagna	A photo of a lasagna, an indulgent layered casserole with a golden-brown crust and fragrant herbs throughout.
Lasagna	A photo of a lasagna, a rustic Italian family meal traditionally prepared in a large, deep dish.
F-16A/B	A photo of the F-16A/B, a type of aircraft, displayed on a military base with combat-ready preparations.
F-16A/B	A photo of a F-16A/B, a type of aircraft, in a dogfight simulation with other fighter jets.
F-16A/B	A photo of the F-16A/B, a type of aircraft, during sunset, highlighting its silhouette against the vibrant colors of the sky.
F-16A/B	A photo of a F-16A/B, a type of aircraft, equipped with precision-guided munitions, ready for a mission.
F-16A/B	A photo of the F-16A/B, a type of aircraft, soaring above mountainous terrain in a training exercise.

TABLE V: Caption descriptions for different categories.

Fig. 5: A comparison of images generated from diverse text captions (top) and images generated with the simple prompt of the class name ”woman” (bottom). The text captions lead to more detailed and diverse images of the class which matters much more in general classes such as ”woman”.

TABLE VI: Spearman Correlation for Text-based Scores

	Consistency			Silhouette			Compound		
Dataset	CLIP	SigLIP	FLAVA	CLIP	SigLIP	FLAVA	CLIP	SigLIP	FLAVA
CUB	0.50	0.31	0.57	0.50	0.31	0.51	0.52	0.32	0.56
Cars	0.51	0.35	0.55	0.52	0.34	0.59	0.52	0.36	0.57
Aircraft	0.35	0.62	0.21	0.35	0.55	0.21	0.36	0.60	0.21
Food	0.31	0.26	0.20	0.35	0.21	-0.09	0.34	0.22	-0.02
ImageNet	0.27	0.19	0.24	0.30	0.14	0.20	0.29	0.17	0.22
ObjectNet	-0.07	-0.02	0.06	0.08	-0.01	-0.07	0.01	-0.02	-0.02
CIFAR	-0.15	-0.09	-0.14	-0.06	-0.02	0.24	-0.10	-0.03	0.12
Flowers	0.55	0.11	0.42	0.56	0.08	0.26	0.60	0.10	0.34
Pets	0.27	0.21	0.20	0.39	-0.01	-0.25	0.37	0.06	-0.18
RESISC	0.31	0.28	0.14	0.34	0.13	0.29	0.34	0.21	0.24

C. Spearman Correlation for Text-based Scores

Table VI displays the spearman correlations between the text consistency, silhouette, and compound scores to the accuracy of the real images. These score demonstrate how captions are generally weaker image proxies than synthetic images. The scores tend to follow similar patterns to the image results, but with most correlations being far lower. The CIFAR dataset is notably far worse than its corresponding image scores perhaps indicating how the captions fail to capture the more general classes of the dataset compared to the lower quality real images.

D. Reproducibility

Datasets can be accessed at: https://huggingface.co/clip-benchmark. The evaluation of real and generated data for a single dataset took between 10 and 25 minutes running on an Nvidia GeForce RTX 3080 depending on the size of the dataset. The code to reproduce this result is at: https://github.com/Will-It-Zero-Shot/Will-it-Zero-Shot.