Active Prompt Learning in Vision Language Models

Jihwan Bang[1] Sumyeong Ahn[2] Jae-Gil Lee[1*]
KAIST[1] Michigan State University[2]
{jihwan.bang, jaegil}@kaist.ac.kr, sumyeong@msu.edu

2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) | 979-8-3503-5300-6/24/$31.00 ©2024 IEEE | DOI: 10.1109/CVPR52733.2024.02550

Abstract

Pre-trained Vision Language Models (VLMs) have demonstrated notable progress in various zero-shot tasks, such as classification and retrieval. Despite their performance, because improving performance on new tasks requires task-specific knowledge, their adaptation is essential. While labels are needed for the adaptation, acquiring them is typically expensive. To overcome this challenge, active learning, a method of achieving a high performance by obtaining labels for a small number of samples from experts, has been studied. Active learning primarily focuses on selecting unlabeled samples for labeling and leveraging them to train models. In this study, we pose the question, "how can the pre-trained VLMs be adapted under the active learning framework?" In response to this inquiry, we observe that (1) simply applying a conventional active learning framework to pre-trained VLMs even may degrade performance compared to random selection because of the class imbalance in labeling candidates, and (2) the knowledge of VLMs can provide hints for achieving the balance before labeling. Based on these observations, we devise a novel active learning framework for VLMs, denoted as `PCB`*. To assess the effectiveness of our approach, we conduct experiments on seven different real-world datasets, and the results demonstrate that* `PCB` *surpasses conventional active learning and random sampling methods. Code is available at https://github.com/kaist-dmlab/pcb.*

1. Introduction

In the past, as emerging research in deep neural networks (DNNs) progressed, there was a substantial focus on studying specific types of datasets, including image/video (vision) [1, 10, 16], natural language [5, 54, 55], graph [63], table [58], and more. However, recent research has raised the question: *"can we develop DNNs capable of understanding multiple types of datasets interactively?"* Among various candidates for multi-modality models, vision language models (VLMs) [31–33, 46, 59] have garnered significant attention due to not only to their wide domain knowledge but also to their superior performance on various tasks.

Most of VLMs, for instance CLIP [46], comprises two encoders: image and text encoders. They have consistently shown impressive zero-shot performance across a wide range of tasks without fine-tuning. For example, CLIP is well-known for its remarkable zero-shot classification performance on various benchmarks, even if the model has not encountered the datasets previously. Despite these notable zero-shot performances, many researchers are focusing on developing adaptation methods for new target tasks because of necessity to make the model aware of the target tasks. Since updating all parameters can be computationally expensive, a key research focus lies in reducing the adaptation computing cost [23, 66, 67]. For example, CoOp [66] takes the approach of freezing both encoders and only allowing a small number of trainable parameters (with a size ranging from 4 to 16) to serve as prompts. This strategy has demonstrated substantial improvements in classification performance with only a small number of trainable parameters and a limited amount of data for each class.

Even though we can reduce the adpation cost, the barrier of high labeling costs still persists. To mitigate this inefficiency, there have been extensive studies in an area of active learning [48, 51]. The central objective of active learning is to select samples for labeling so that the model performance is significantly improved, making a noticebale gap compared to random samples of the same quantity. These active learning methods can be roughly divided into two categories: (1) uncertainty-based sampling [12, 18, 19, 25, 47] and (2) diversity-based sampling [43, 50] which leverages feature embeddings from the image encoder. In a hybrid perspective, BADGE [2] was introduced by combining uncertainty and diversity through the use of k-means++ clustering within the gradient embedding space.

Under these two researches, our initial inquiry pertains to the determination of whether the implementation simply combining active learning with VLMs can effectively lead to enhanced classification performance. If it does not result in such improvement, what constitutes the critical in-

* indicates corresponding author.

2575-7075/24/$31.00 ©2024 IEEE
DOI 10.1109/CVPR52733.2024.02550

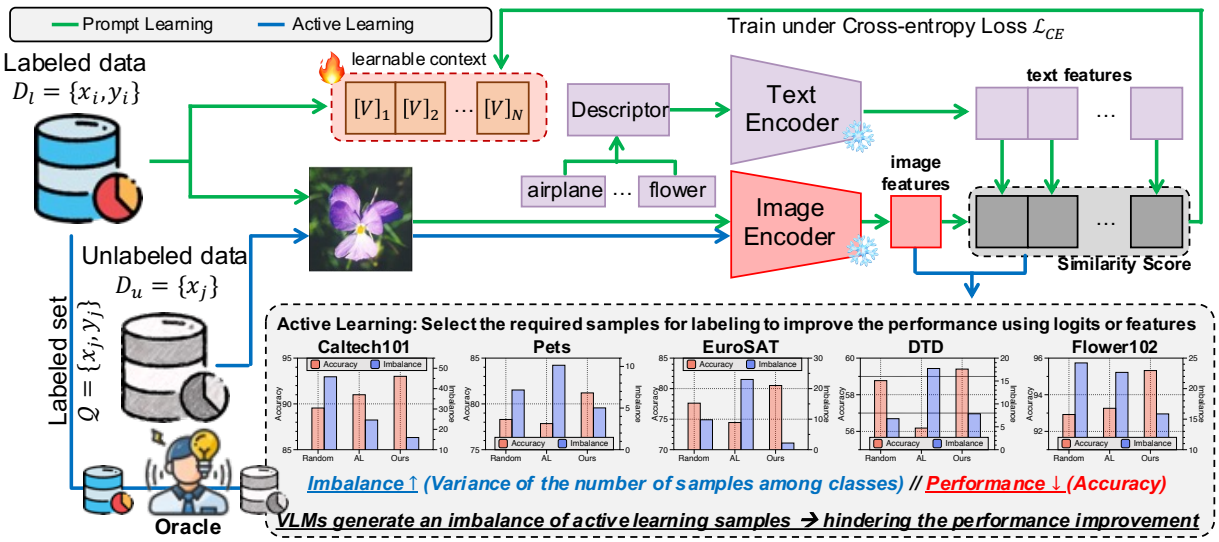


Figure 1. **Key motivation and complete process behind active prompt learning.** When we emply a traditional active learning framework for adapting prompt learning to a new target task, the active learning sampler incurs a significant imbalance (indicated by red bars). Thus, this imbalance results in an inability to enhance the ultimate performance (as indicated by blue bars). In this paper, we introduce a novel algorithm named PCB that rectifies this imbalance by harnessing the knowledge of VLMs, enabling effective utilization of the oracle.

congruity in integrating these two methodologies? To address this question, we observe two phenomena: (1) naïvely applying active learning to VLMs does not consistently demonstrate improvements compared to random selection-based labeling (depicted as red bars in Figure 1); (2) this lack of improvement comes from the imbalanced class labels misled by an active learning framework (illustrated as blue bars in Figure 1). The imbalanced behavior of active learning algorithms is due to the imbalanced pre-trained knowledge of VLMs. We verify that pre-trained CLIP has different knowledge of each class by showing the class-wise accuracy (see Appendix A). Therefore, it is imperative to investigate how VLMs can effectively collaborate with active learning frameworks, particularly given that VLMs exacerbate the issue of class imbalance.

In this study, we introduce our approach, called PCB, which is designed to address the class imbalance issue and improve the classification performance of VLMs using only a limited amount of labeled data from experts. Our contributions are summarized as follows:

- This study represents the first exploration of synergistic approaches to active learning and VLMs, marking a novel contribution to the field. We establish that a straightforward combination of these two approaches does not consistently lead to an improvement, highlighting the need for enhancing active learning methods in this context.
- We delve into the underlying reasons for performance degradation of conventional active learning methods when combined with VLMs. Our investigation reveals that the selection of samples to be labeled by experts is imbalanced, thereby making VLMs biased.
- We introduce an algorithm named PCB, which harnesses the valuable pre-trained knowledge of VLMs to address the issue of class imbalance. This algorithm seamlessly integrates with conventional active learning techniques, resulting in a substantial enhancement of active prompt learning within VLMs.

2. Problem Formulation

2.1. Active Learning

The objective of active learning is to facilitate the learning of a multi-class classification model with K classes while minimizing the labeling budget. The model undergoes an iterative training process through interactions with an oracle, who provides correct annotations. In each iteration, the model learns from an annotated dataset, denoted as $\mathcal{D}_l = \{(x_i, y_i)\}_{i=1}^{L}$, where $x_i \in \mathcal{X}$ is the input (*e.g.* images), and $y_i \in \{1, ..., K\}$ is the corresponding label, and L is the number of labeled samples. Upon sufficiently training the model with the given dataset $\mathcal{D}_l$, the active learning algorithm selects N samples from the unlabeled dataset, $\mathcal{D}_u$. The oracle then provides the labels for these selected samples, and they are subsequently incorporated into the annotated dataset $\mathcal{D}_l$ for use in the training.

2.2. Vision Language Models and Prompt Learning

Vision language models (VLMs). VLMs typically consist of two encoders: an image encoder and a text encoder. When presented with an image as an input, the image encoder transforms the image into an embedding vector. In the case of CLIP [46], one of the most representative VLMs,

it employs the ResNet [16] and ViT [10] architectures as its image encoder. On the other hand, the objective of the text encoder is to map a sentence into an embedding vector with the same dimension as the output of the image encoder. CLIP employs the Transformer [56] architecture for its text encoder. The CLIP model is trained using an Image-Text constrastive loss to align the embeddings of image and text pairs, enabling it to learn meaningful associations between images and corresponding text descriptions.

Indeed, the classification process in the CLIP model relies on the similarity score between image and text pairs. Here is a summary of how the CLIP model performs classification. Given an image x_i, the embeddings for the image and text are formulated as

$$e_{\text{img}} = \text{CLIP}_{\text{img}}(x_i), \quad e^k_{\text{txt}} = \text{CLIP}_{\text{txt}}(T(\text{CLS}_k)).$$

Here, T represents the text template (*e.g. A photo of* $\{\text{CLS}_k\}$), and $\{\text{CLS}_k\}$ denotes the class name of each class index $k \in \{1, 2, ..., K\}$. Using e_{img} and e_{txt}, the prediction probability of each class k is formulated as

$$\text{P}(y = k|x) = \frac{\exp(\cos(e_{\text{img}}, e^k_{\text{txt}})/\tau)}{\sum_{i=1}^{K} \exp(\cos(e_{\text{img}}, e^i_{\text{txt}})/\tau)}$$

where τ represents a temperature parameter, and $\cos(\cdot, \cdot)$ is the cosine similarity.

Prompt learning (PL). PL is an efficient adaptation method that allows partial parts of prompts to be trainable [14, 21–23, 28, 34, 36, 52, 65–67]. It improves performance by updating the following criteria. Suppose that the class name $\{\text{CLS}\}$ is tokenized as $[\text{CLS}]$, and the text template function T generates tokens as follows:

$$T(\text{CLS}_k) = [V]_1[V]_2 \dots [V]_M[\text{CLS}_k].$$

Here, $[V]_i$ represent trainable tokens, and $[\text{CLS}_k]$ is a fixed token for each class name $\{\text{CLS}_k\}$. Note that the position of trainable tokens can be changed, *e.g.* by placing them right after the class token; we simply notate it as the *front case* for simplification. These trainable parameters are trained using the cross-entropy loss function defined as

$$\mathcal{L}_{\text{CE}}(x_i, y_i) = -\sum_{k=1}^{K} \mathbb{1}\{y_i = k\} \log P(y = k|x_i).$$

3. Method: PCB

As depicted in Figure 1, two key motivations can be derived: (1) the traditional approach to select unlabeled samples in active learning leads to an imbalance under pre-trained VLMs, and (2) achieving a balance is imperative for enhancing overall performance, but it is challenging with unlabeled data; it can rather cause deeper imbalance by using false knowledge. Building upon the insights from these findings, we recognize the significance of balancing in improving performance within the active prompt learning problem. To address this objective, we introduce a novel algorithm named PCB: **P**seudo-**C**lass **B**alance for Active Prompt Learning in VLMs. In the following section, we delve into a detailed explanation of the entire workflow encompassed by the proposed algorithm.

Algorithm 1: `Balance_sampler`

Input: Labeled dataset $\mathcal{D}_l$, Pseudo-labeled dataset $\tilde{P}$, Budget N
Init: $\mathcal{Q} = \emptyset$ (Query set), $\tilde{\mathcal{D}}_l = \mathcal{D}_l$ (Estimated $\mathcal{D}_l$)
for $n = 1, 2, ..., N$ **do**
  # Select class k, the smallest # of class samples in $\tilde{\mathcal{D}}_l$
  $k = \arg\min_{k \in \{1,...,K\}} |c_k|$
    where $c_k = \{(x_i, y_i) | y_i = k \text{ and } (x_i, y_i) \in \tilde{\mathcal{D}}_l\}$
  # Select one sample pseudo-labeled as k from $\tilde{P}$
  $(x_j, \tilde{y}_j) \in \tilde{P}$, where $\tilde{y}_j = k$
  # Update query set $\mathcal{Q}$ by adding the selected sample
  $\mathcal{Q} = \mathcal{Q} \cup [(x_j)]$
  # Update estimated labeled set $\tilde{\mathcal{D}}_l$
  $\tilde{\mathcal{D}}_l = \tilde{\mathcal{D}}_l \cup [(x_j, \tilde{y}_j)]$
end
Output: Query set $\mathcal{Q}$

3.1. Pseudo-Class Balance for Active Prompt Learning in VLMs

Balance sampler. To satisfy the class balance while selecting informative samples for improving ultimate performance, we propose the two-stage active learning method on VLMs. First, we select a subset of informative samples, $P \subset \mathcal{D}_u$ where the size of P is $\gamma \times |\mathcal{D}_u|$. Here $\gamma \in [0, 1]$ is the hyperparameter that controls how progressively allow the uncertain samples to be labeled. After selecting P, we pseudo-label the selected samples by using VLMs' classification ability, *i.e.* $\tilde{P} = \{(x_i, \tilde{y}_i)\}_{i=1}^{\gamma|\mathcal{D}_u|}$. Then, we build the query set $\mathcal{Q}$ by utilizing the balance sampler, as described in Algorithm 1, randomly selecting samples from $\tilde{P}$ so that the expected number of samples of each class is balanced.

Proposed method. Based on the balancing module, we introduce a method called PCB, which is briefly outlined in Algorithm 2. This algorithm takes the inputs as a labeled dataset $\mathcal{D}_l$, an unlabeled dataset $\mathcal{D}_u$, the number of active learning rounds R, a query budget N, a progressive hyperparameter γ, an active learning algorithm $\mathcal{A}$, an oracle labeler `Oracle`, and a VLM model f.

The initial round randomly selects a query set due to insufficient information about the target dataset. From the second round, an active learning algorithm builds an informative subset P and assigns pseudo-labels to its samples. Algorithm 1 then aims to create a balanced labeled dataset. After obtaining true labels for the query set $\mathcal{Q}$ from an oracle, the procedure proceeds to train the parameters $[V]_i$.

Algorithm 2: PCB

Input: $\mathcal{D}_l$, $\mathcal{D}_u$, R, N, γ, $\mathcal{A}$, $\texttt{Oracle}(\cdot)$, f.
for $r = 1, 2, ..., R$ **do**
  if $r = 1$ **then**
    # Initial query set $\mathcal{Q}$
    $\mathcal{Q} = \texttt{random\_sample}(\mathcal{D}_u, N)$
  end
  else
    # Select informative subset P
    $P = \mathcal{A}(\mathcal{D}_u, \gamma|\mathcal{D}_u|, f)$
    # Pseudo labeling
    $\tilde{P} = \{(x_i, \tilde{y}_i)|(x_i, y_i) \in \mathcal{D}_u \text{ and } \tilde{y}_i = f(x_i)\}$
    # Balance_sampler (Algorithm 1)
    $\mathcal{Q} = \texttt{Balance\_sampler}(\mathcal{D}_u, \tilde{\mathcal{P}}, N)$
  end
  # Labeling by Oracle
  $\hat{\mathcal{Q}} = \{(x_i, y_i)|x_i \in \mathcal{Q} \text{ and } y_i = \texttt{Oracle}(x_i)\}$
  # Update both sets and train prompts (Section 3.2)
  $\mathcal{D}_l = \mathcal{D}_l \cup \hat{\mathcal{Q}}$, $\mathcal{D}_u = \mathcal{D}_u \backslash \mathcal{Q}$
  Train learnable prompts $[V]_i$ on $\mathcal{D}_l$
end
Output: Final model f

3.2. Description Augmentation

In order to improve the classification performance of VLM-based models, numerous studies have explored the integration of external knowledge, as demonstrated in previous research works [39, 44]. These studies have contributed to show how models (*e.g.* GPT-3 [5]) can help VLMs by generating visual description for each class. For instance, the authors of [39] introduced the following templates:

```
Q: What are useful features for
distinguishing a {CLS} in a photo?

A: There are several useful visual
features to tell there is a {CLS} in
a photo:
```

where `{CLS}` indicates the class name. Here, note that we can obtain δ_k descriptions for class k, *i.e.* $\Delta_k = \{\mathrm{d}_k^i\}_{i=1}^{\delta_k}$, where d_k^i denote i-th description for class k. See Appendix B for the detailed prompt template.

By following the results of [39, 44], we adopt their prompts for training the model. In other words, we utilize the new text template function T as follows:

$$T(\mathrm{CLS}_k, i) = [V]_1 ... [V]_M [\mathrm{CLS}_k] \ [\mathrm{which}] \ [\mathrm{is}] \ [\mathrm{d}_k^i],$$

Based on this new text template function, we can use two possible prediction probabilites.

(1) Average Similarity (AS):

$$\mathrm{P}(y = k|x) = \frac{1}{\delta_k} \sum_{i=1}^{\delta_k} \mathrm{P}(y = k|x, \mathrm{d}_k^i),$$

where

$$\mathrm{P}(y = k|x, \mathrm{d}_k^i) = \frac{\exp(\cos(e_{\mathrm{img}}, e_{\mathrm{txt}}^{k,i})/\tau)}{\sum_{i=1}^{K} \sum_{j=1}^{\delta_k} \exp(\cos(e_{\mathrm{img}}, e_{\mathrm{txt}}^{k,i})/\tau)}.$$

(2) Average Embedding (AE):

$$\mathrm{P}(y = k|x) = \frac{\exp(\cos(e_{\mathrm{img}}, e_{\mathrm{txt}}^{k})/\tau)}{\sum_{i=1}^{K} \exp(\cos(e_{\mathrm{img}}, e_{\mathrm{txt}}^{i})/\tau)},$$

where

$$e_{\mathrm{txt}}^{k} = \frac{1}{\delta_k} \sum_{i=1}^{\delta_k} e_{\mathrm{txt}}^{k,i}.$$

Note that the primary distinction between two probability scores lies in their respective averaging timeframes. AS calcuates individual embeddings and then computes the average similarity, whereas AE first averages the embeddings and then assesses the similarity.

4. Experiment

4.1. Implementation Details

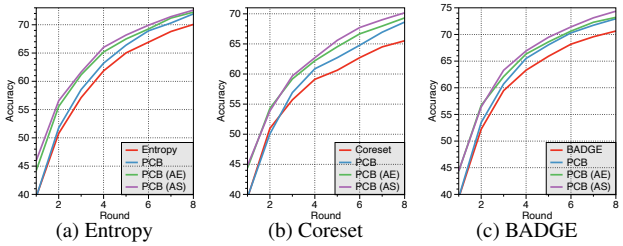
Datasets. For image classification in downstream tasks, we select seven publicly available image classification datasets that have been previously utilized in the CLIP model [46], specifically EuroSAT [17], Oxford Pets [42], DTD [6], Caltech101 [11], Flowers102 [41], StanfordCars [27], and FGVC-Aircraft [37]. These benchmarks span diverse categories, encompassing classification tasks involving common objects, scenes, patterns, and fine-grained categories. For more details regarding the datasets, see Appendix C.

Training details. Our active learning setup consists of eight rounds (*i.e.* R=8), and in each round, we select a subset whose size is the number of classes, *i.e.* N=K . To serve as the backbone for our image encoder, we adopt ViT-B/32. The size of the context vectors M is set to 16, and they are initialized using a zero-mean Gaussian distribution with a standard deviation of 0.02. Throughout the training process for all rounds, we employ the SGD optimizer with a learning rate 0.002, which is decayed by the cosine annealing scheduler. We also set the maximum epoch as 200. All methods are implemented with PyTorch 2.0.1 and executed on a single NVIDIA A5000 GPU.

Active learning methods. To validate the effectiveness of PCB, we select three representative active learning methods: (1) Entropy [18] selects the most uncertain examples with the highest entropy value from logits in the prediction; (2) Coreset [50] queries the most diverse examples using embeddings from the model (*i.e.* image encoder); and (3) BADGE [2] considers both uncertainty and diversity by selecting the examples via k-means++ clustering in the gradient space. By adding PCB into those active learning methods, we study the synergy of active learning and PCB, and

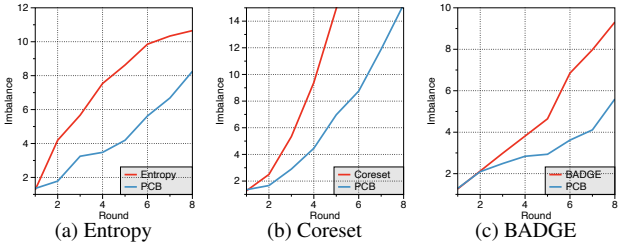
Method	Flowers102	DTD	Oxford Pets	EuroSAT	Caltech101	Stanford Cars	Aircraft	Avg Acc (↑)
CLIP (zero-shot)	66.7	44.5	87.0	49.4	87.9	59.4	21.2	59.44
Random	92.92±0.61	58.77±1.94	78.30±0.74	77.62±1.12	89.55±1.00	65.96±0.08	30.69±0.30	70.54
Entropy [18]	94.80±0.75	59.18±1.31	76.81±1.38	75.46±3.39	91.67±0.09	66.68±0.91	25.80±0.78	70.06
+ AE	96.06±0.63	60.80±1.18	78.35±1.30	79.97±2.70	92.87±0.20	65.99±0.26	26.69±1.34	71.53
+ AS	95.67±1.19	59.34±0.81	79.88±1.43	79.50±0.60	93.28±0.55	68.54±0.09	26.04±1.27	71.75
+ PCB	96.16±0.45	59.73±1.96	80.44±1.24	80.80±2.88	92.41±0.50	67.18±0.28	26.78±0.87	71.93
+ PCB(AE)	96.33±0.06	60.07±1.69	80.87±0.60	81.72±0.53	93.14±0.51	66.42±0.86	27.09±0.13	72.23
+ PCB(AS)	**96.94**±0.19	59.50±1.99	80.94±1.05	80.75±1.15	93.48±0.26	68.93±0.86	27.58±0.43	72.59
Coreset [50]	88.65±0.68	50.39±0.54	76.70±0.52	68.09±1.54	88.78±0.49	61.75±0.60	24.32±0.45	65.53
+ AE	89.06±0.62	51.89±1.38	78.08±1.07	68.02±2.86	88.99±0.82	60.65±0.33	25.88±0.70	66.08
+ AS	89.73±0.93	52.76±1.21	78.89±0.84	68.07±1.04	90.63±0.54	64.15±0.77	26.11±0.86	67.19
+ PCB	91.30±0.90	55.77±1.33	76.84±1.10	77.50±4.64	89.96±0.03	63.63±0.27	25.38±0.64	68.63
+ PCB(AE)	91.70±0.29	57.09±0.63	78.60±1.14	79.28±0.14	90.29±0.30	62.08±0.35	26.19±1.40	69.31
+ PCB(AS)	92.33±0.84	56.38±0.73	79.50±0.91	79.28±1.42	91.70±0.48	65.75±0.55	26.22±0.47	70.17
BADGE [2]	96.33±0.39	58.98±1.30	80.03±1.19	79.79±0.94	92.54±0.01	68.07±0.61	31.25±0.45	72.43
+ AE	96.24±0.29	59.97±0.71	81.94±0.55	80.57±1.40	92.93±0.02	67.10±0.47	31.04±0.32	72.83
+ AS	96.44±0.16	61.52±1.25	82.33±0.72	81.66±0.41	93.79±0.25	70.56±0.31	31.79±0.74	74.01
+ PCB	96.12±0.12	60.28±0.75	80.22±1.69	**81.98**±0.81	92.21±0.92	68.50±0.26	31.35±0.21	72.95
+ PCB(AE)	96.35±0.27	61.92±1.60	81.93±0.88	80.70±3.67	92.52±0.32	67.70±0.84	31.80±0.08	73.27
+ PCB(AS)	96.71±0.29	**62.33**±1.06	**83.16**±0.18	81.50±1.11	**93.85**±0.37	**70.70**±0.79	**32.27**±0.66	**74.36**
Full data	97.9	74.7	89.3	94.5	94.4	80.8	43.4	82.14

Table 1. **Final accuracy on seven downstream tasks with the ViT-B/32 image encoder.** Final Accuracy is the the accuracy after eight rounds, and Avg Acc is the average of the final accuracies of seven datasets. AS and AE are the average score and the average embedding, respectively, as described in Section 3.2. Also, CLIP (zero-shot) is the accuracy of each task from pretrained CLIP as reported in [46], and Full data is the accuracy when exploiting the whole dataset while prompt learning. Please note that the boldface and underscore represent the best performance overall and within the same active learning, respectively. For large datasets, see Appendix D.



(a) Entropy (b) Coreset (c) BADGE

Figure 2. **Learning curve.** Average accuracy on downstream tasks with the ViT-B/32 image encoder for each round.



(a) Entropy (b) Coreset (c) BADGE

Figure 3. **Imbalance curve.** Average variance of the number of labeled samples for each class on downstream tasks with the ViT-B/32 image encoder for each round.

compare the results with random sampling (*i.e.* instead of active learning) and zero-shot CLIP. Furthermore, we show the results when using the descriptions presented in Section 3.2. To illustrate the room for performance enhancement, we also measure the performance when prompt learning the model with the whole dataset (see "Full data").

Metrics. To validate the effectiveness of our method, we use the final accuracy that indicates the accuracy at the last round. As in previous analysis about imbalance, we use the variance value of the number of samples among classes. Note that all experiments are conducted three times, and all the results are reported as averages.

4.2. Overall Results

PCB improves performance. We evaluate our methodology by integrating it with three active learning methods—Entropy, Coreset, and BADGE—and compare it with the Random approach and the pre-trained zero-shot CLIP model. As shown in Table 1, the proposed algorithm mostly improves performance in each case of its integration. For example, in the DTD dataset with the BADGE algorithm, applying PCB (AS) results in a 3.35% improvement compared to the case without our algorithm. Furthermore, on average across datasets, leveraging our algorithm shows a performance improvement up to 4.64%. Additionally, across all active learning algorithms, PCB (AS) cases typically exhibit the highest accuracy.

Active learning can be poor than Random. As shown in Table 1, active learning algorithms exhibit lower performance than the Random approach in some cases. For example, especially in EuroSAT and Aircraft, both Entropy and Coreset strategies perform worse than the Random approach, because these active learning algorithms select imbalanced datasets for labeling, leading to performance degradation. See Appendix A for the detailed evidences.

Learning curve. Figure 2 illustrates the learning curve of the avearge accuracy among various datasets for each algo-

Model	Method	Flowers102	DTD	Oxford Pets	EuroSAT	Caltech101	Stanford Cars	Aircraft	Avg Acc (↑)
RN50	CLIP (zero-shot)	65.9	41.7	85.4	41.1	82.1	55.8	19.3	55.9
	Random	92.06±0.54	56.62±0.97	74.65±0.50	79.10±2.31	84.11±0.75	61.34±0.57	29.15±0.32	68.18
	BADGE [2]	95.56±0.54	58.35±1.20	75.06±0.50	80.94±0.55	89.67±0.30	63.96±0.53	28.12±1.03	70.24
	+ PCB	95.66±0.28	57.41±0.17	76.51±1.83	80.06±0.97	89.06±0.21	63.18±0.77	29.23±0.35	70.16
	+ PCB(AE)	95.72±0.31	**59.20**±1.25	76.77±0.65	**81.96**±0.60	89.57±0.19	62.62±0.26	28.85±1.59	70.67
	+ PCB(AS)	**96.18**±0.07	59.14±1.08	**80.09**±0.85	81.60±2.89	**90.76**±0.34	**66.20**±0.69	**29.61**±0.78	**71.94**
	Full data	97.6	71.6	88.0	93.6	92.8	78.8	42.6	80.71
RN101	CLIP (zero-shot)	65.7	43.9	86.2	33.1	85.1	62.3	19.5	56.54
	Random	92.87±0.43	58.29±1.24	79.08±1.39	77.21±4.13	87.55±0.75	70.02±0.36	32.76±0.29	71.11
	BADGE [2]	96.26±0.07	59.93±1.25	80.77±1.31	78.23±2.22	91.35±0.32	71.43±0.97	32.56±0.64	72.93
	+ PCB	95.79±0.38	60.20±1.89	80.94±0.42	79.55±1.37	91.75±0.44	71.35±0.39	32.62±1.48	73.17
	+ PCB(AE)	**96.49**±0.26	**62.59**±0.84	83.02±0.89	**81.50**±0.69	92.51±0.32	71.42±0.77	32.76±0.76	74.33
	+ PCB(AS)	96.47±0.18	62.17±1.04	**83.48**±2.13	81.14±1.57	**92.87**±0.18	**74.04**±0.39	**32.84**±0.85	**75.43**
	Full data	97.8	74.2	91.1	92.9	94.7	83.7	46.0	82.91
ViT-B/16	CLIP (zero-shot)	70.4	46.0	88.9	54.1	88.9	65.6	27.1	63.0
	Random	94.98±0.06	62.63±1.81	84.36±1.34	81.14±1.83	90.95±0.85	73.62±0.30	38.88±0.25	75.22
	BADGE [2]	97.97±0.41	62.84±2.17	85.54±1.30	82.22±1.94	93.77±0.51	76.55±0.78	39.64±0.14	76.93
	+ PCB	**98.32**±0.21	64.89±1.45	86.22±0.71	81.53±3.11	93.75±0.28	76.36±0.27	40.20±0.30	77.32
	+ PCB(AE)	98.21±0.21	**65.25**±1.28	87.23±0.35	**84.04**±2.92	94.51±0.29	75.84±0.44	39.93±0.21	77.86
	+ PCB(AS)	98.19±0.17	64.95±1.47	**88.10**±1.49	83.85±2.45	**95.12**±0.26	**78.19**±0.48	**40.56**±0.51	**78.42**
	Full data	99.0	77.7	92.7	95.1	95.3	85.3	53.6	85.53

Table 2. **Various architectures of an image encoder.** We report the performance on various types of architectures, such as ResNet-50/101 and ViT-B/16, under the BADGE active learning algorithm. The performance under Entropy and Coreset is described in Appendix F.

rithm with or without the proposed algorithm. In all cases, applying PCB with AS shows the best performance. Furthermore, utilizing PCB improves the performance compared to the active learning algorithms without PCB. In particular, all figures represent that applying PCB demonstrates an increasing gap between with PCB and without PCB as the training progresses. It indicates that, based on our imbalance analysis described in Figure 3 and detailed in Appendix A, reducing the imbalance when constructing the query set $\mathcal{Q}$ is crucial in active prompt learning.

Additional analysis. In the case of Oxford Pets, PCB exhibits lower performance than CLIP (zero-shot). This result is consistent with the results from the original CoOp paper [67]. To further analyze this phenomenon, we increase the number of samples (*i.e.* N) selected at each round by the active learning algorithm from K to $16K$. When N=$4K$, PCB combined with BADGE outperforms zero-shot CLIP, and detailed results are described in Appendix E. We can conclude that the reason of performance degradation in Oxford Pets is due to the lack of samples involved in training.

4.3. Detailed Analyses

In this section, we answer the following questions: (1) other architectures of the image encoder in the CLIP family, (2) class imbalance analysis over different γ values, (3) analysis on various prompt learning methods, and (4) hyperparameter sensitivity analysis.

Other types of image encoder. We assess the efficacy of our method across various image encoder models, as described in Table 2. Given the superior performance of PCB coupled with BADGE, as evidenced in Table 1, our subsequent analysis in Table 2 is confined to this particular setup. Our method shows a similar trend across different encoder architectures, as supported by these observations: (1) the accuracy of zero-shot CLIP can be significantly enhanced with finetuning randomly sampled data, (2) the combination of BADGE with PCB yields better results than random sampling, and (3) description augmentation (*i.e.* AS and AE) noticeably enhances the accuracy. Regardless of the encoder used, the overall accuracy increases, with the difference between AS and AE decreasing as the model size grows.

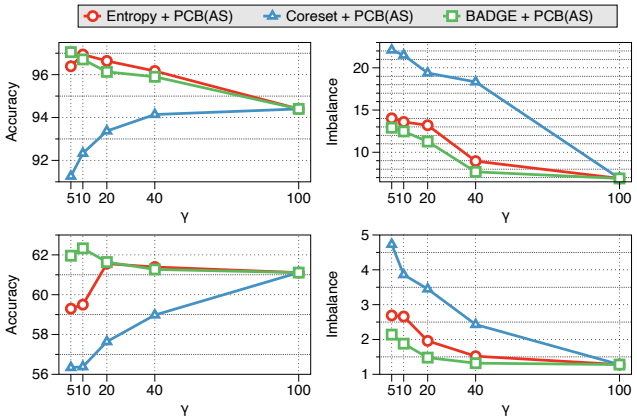
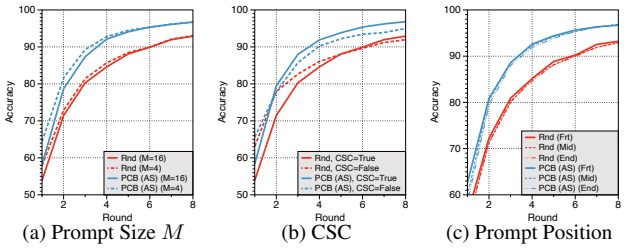


Figure 4. **Accuracy and imbalance in terms of various γ on Flowers102 (Upper) and DTD (Bottom).**

Class imbalance analysis over different γ. We also study the effectiveness of our method as γ increases and summa-



(a) Prompt Size M (b) CSC (c) Prompt Position

Figure 5. **CoOp case analysis of BADGE on Flowers102.**

rize the results in terms of accuracy and imbalance in Figure 4. As shown in the figure, the accuracy increases as the imbalance decreases. In particular, Coreset+PCB (AS) has higher accuracy and lower imbalance as γ increases due to a larger number of unlabeled examples to balance classes. On the contrary, it is noteworthy that the accuracies of Entropy+PCB (AS) and BADGE+PCB (AS) do not tend to improve as γ increases despite of more unlabeled samples for balancing. It indicates that getting informative (*i.e.* uncertain) data is very important to improve the accuracy after achieving a certain level of balance.

Moreover, we compare the accuracies and imbalances on two different datasets: Flowers102, which lacks class balance, and DTD, which exhibits class balance. As shown in Figure 4, the value of imbalance in Flowers102 is larger than that in DTD. Interestingly, in Flowers102, the imbalance value drops most dramatically in the range of 20%–40% of γ, whereas in DTD, it drops most dramatically in the range of 10%–20% of γ. This observation indicates that achieving a class balance is obviously harder in an imbalanced dataset than in a balanced dataset.

Hyperparameter sensitivity. We examine various variants of CoOp training methods, including different prompt sizes (M), cases where class-wise different tokens are not allowed (CSC=False), and variants in the position of trainable parameters (Front, Middle, and End).

In Figure 5a, it is observed that the accuracy with a small M is higher than that with a large M, but this gap decreases as the rounds progress. Since the number of trainable parameters with a large M is greater than that with a small M, the model with a large M can be easily overfitted by a small labeled dataset at the initial round.

We analyze the performance gap when context vectors are shared for all classes (*i.e.* CSC=False) versus when different context vectors are used per class (*i.e.* CSC=True), and report the results in Figure 5b. The accuracy is shown to be initially higher when CSC=False, but it is beaten by CSC=True as the rounds progress. This phenomenon is attributed to the difference in the number of trainable parameters similarly to Figure 5a.

Last, we measure the accuracy by changing the position of context vectors with Front (Frt), Middle (Mid), and End. As shown in Figure 5c, the accuracies with the Front position of context vectors are slightly better than those with

Visual	Text	Method	N	PCB	Entropy	Coreset	BADGE
RN50	None	LP	102	✗	79.78±1.01	70.66±1.16	81.23±0.40
		FFT	102	✗	47.67±1.08	48.19±2.35	53.43±0.61
		FFT	250	✗	77.48±3.45	78.91±0.77	79.51±0.32
	Transformer	CoCoOp [66]	102	✗	74.54±1.28	71.58±0.73	78.60±0.52
		CoOp [67]	102	✗	94.74±0.40	85.61±1.36	95.56±0.54
	Transformer	CoCoOp [66]	102	$\mathcal{O}$	76.18±1.55	72.74±1.29	80.06±1.53
		CoOp [67]	102	$\mathcal{O}$	95.89±0.32	91.34±1.00	95.66±0.28
ViT-B/32	None	LP	102	✗	94.19±0.77	86.76±0.55	95.57±0.15
		FFT	102	✗	37.01±1.69	35.90±1.26	43.14±0.89
		FFT	250	✗	58.21±2.89	58.63±2.87	60.10±0.47
	Transformer	CoCoOp [66]	102	✗	76.41±1.29	73.54±0.68	78.94±0.36
		MaPLe [23]	102	✗	84.72±2.56	80.98±0.80	87.86±1.84
		CoOp [67]	102	✗	94.80±0.75	88.65±0.68	96.33±0.39
	Transformer	CoCoOp [66]	102	$\mathcal{O}$	77.28±1.71	73.91±0.97	80.39±0.48
		MaPLe [23]	102	$\mathcal{O}$	87.60±1.93	82.51±0.22	88.14±0.73
		CoOp [67]	102	$\mathcal{O}$	96.16±0.45	91.30±0.90	96.12±0.12
ViT-B/16	Transformer	CoCoOp [66]	102	✗	84.62±1.95	78.44±1.91	86.85±1.21
		MaPLe [23]	102	✗	92.66±1.20	85.54±1.73	93.29±0.39
		CoOp [67]	102	✗	97.32±0.23	92.22±2.03	97.97±0.41
	Transformer	CoCoOp [66]	102	$\mathcal{O}$	85.61±1.63	80.44±0.56	87.41±1.42
		MaPLe [23]	102	$\mathcal{O}$	93.72±0.95	87.58±0.48	93.34±1.02
		CoOp [67]	102	$\mathcal{O}$	97.75±0.08	94.79±0.31	98.32±0.21

Table 3. **Results of various training methods on Flowers102.**

the others over all rounds, but the gap is within the standard deviation. As such, it is hard to conclude that the position of context vectors affects the performance in active learning.

Various prompt learning methods. There have been various types of prompt learning algorithms. Specifically, CoCoOp [66] and MaPLe [23] are popular among recent approaches, and they mainly focus on transferring to unseen novel classes. We evaluate the Flower102 performance of PCB and active learning algorithms on these other prompt learning algorithms, even though they do not mainly target the case where all classes are visible at the training phase. Furthermore, we examine the performance of full fine-tuning (FFT), which tunes all parameters and place a linear classifier on top of the model, and linear probing (LP), which trains the linear layer to adapt to the new task.

First of all, without considering transferability, *i.e.* CoCoOp and MaPLe, CoOp shows better performance than LP and FFT. Also, we observe that the performance of CoCoOp and MaPLe is lower than that of CoOp. This observation aligns with the results reported in each paper, specifically concerning the base class performance, which pertains to the seen class during the training phase. Regardless of their performance superiority, when we compare the performance of ✗ and $\mathcal{O}$ indicating the setups without and with PCB, we find that PCB consistently improves performance. For instance, in the case of CoCoOp ViT-B/32, it enhances performance by 1.45% point.

More precisely, we can conclude that FFT exhibits lower performance in a few-shot case. This phenomenon has also been reported in previous work [66], and we can attriute it to the few-shot training, as evidenced by the performance increase when we increase the number of samples from 102 to 250. However, it performs less effective than CoOp, indicating that prompt learning is superior to adaptation for new tasks in a few-shot perspective. Furthermore, PCB further enhances this improvement in an active learning setting.

5. Related Work

Vision language models (VLMs). To comprehend the visual and language representations, multiple approaches have been explored [7, 8, 13, 30, 35, 45, 60]. In the stream of trials to understand both modalities at once, several years ago, CLIP [46] emerged, drawing significant attention due to its remarkable zero-shot performance across various tasks. In a similar vein, ALIGN [20] was introduced, employing a comparable training methodology but featuring distinct architectural and training dataset characteristics. Unlike CLIP, ALBEF [31] introduced multi-modal transformer operations applied to the outputs of two separate image and text encoders. BLIP [32] introduced a captioning module aimed at improving model performance by rectifying noisy captions. LiT [64] and BLIP-2 [33] enhanced training efficiency by freezing specific encoder parameters. The authors of FILIP [59] endeavored to enable the model to discern finer image details through a fine-grained, *i.e.* patch-level, matching training approach. Florence [62], on the other hand, sought to expand representations from various perspectives, such as image-to-video.

Prompt learning in VLMs. In the realm of natural language processing, there has been numerous works [14, 21, 23, 28, 34, 52, 65] aimed at enhancing the performance of language models through the optimization of prompts. The primary motivation behind these works lies in the huge size of models for fine-tuning, and it is also prevalent in the VLM area. Consequently, a considerable amount of research has been dedicated to prompt learning as a means to enhance classification accuracy. CoOp [67] is one of representative methods and has demonstrated that a minimal number of trainable parameters suffice to adapt to a given classification task. The authors of [36] introduced a methodology that leverages estimated weight distributions to assign weights aimed at minimizing classification errors.

Within this framework, several studies have aimed to enhance the generalization performance in prompt learning for VLMs [23, 24, 61, 66]. The primary task of these studies is to showcase a small number of classes and evaluate unseen ones. The same authors in [67] introduced CoCoOp [66], which incorporates a meta-network module to improve transferability. In the work presented in [61], the authors elucidated transferability in prompt learning from a VLM perspective. Moreover, MaPLe [23], a branch-aware hierarchical prompt method, was proposed where prompts for the image encoder are influenced by prompts for the text encoder. In PromptSRC [24], the authors highlight that previous prompt learning methods have overlooked the forgetting phenomenon during prompt training and propose an alignment-based self-regularization method to enhance transferability. It is important to note that this paper primarily focuses on active learning not for generalizability to new classes but for enhanced performance on the given task.

Description augmentation. Recently, generating descriptions using large language models (LLMs) has gained popularity, owing to the significant improvements in the performance of VLMs. To generate descriptions for each specific class, we asked the questions based on the specific prompt template to the LLMs. A method was proposed by [39], where the scores obtained from different descriptions of the same class were averaged. In contrast, [44] proposed a method, called `CuPL`, that utilized the averaged embeddings from multiple descriptions for each class.

Active learning. Active learning [15, 40, 48, 51] aims to minimize human labeling costs by identifying informative data to maximize model performance. Most of the work has generally progressed along two trajectories: (1) uncertainty-based sampling and (2) diversity-based sampling. In uncertainty-based sampling, prediction probability-based sampling methods such as soft-max confidence [29], margin [49], and entropy [18] were simple yet effective. In addition, some methods performed multiple forward passes to achieve uncertainty. An intuitive approach was to receive the outputs from multiple experts [3, 26, 38]. Some methods [12, 19, 25] leveraged the Monte Carlo Dropout, which obtains the stochastic results from the same model using a dropout layer. On the other hand, diversity-based sampling methods [43, 50] were introduced using either clustering or coreset selection protocols. The coreset method [50] identified sets of examples with the greatest coverage distance across all unlabeled data. More recently, hybrid methods leveraging both uncertainty and diversity have emerged. One such method, BADGE [2], employed k-means++ clustering within the gradient embedding space.

6. Conclusion

In this paper, we delve into the realm of active prompt learning within vision-language models (VLMs). Initially, we observe a misalignment between previous active learning algorithms and VLMs due to the inherent knowledge imbalance of VLMs. This imbalance consequently leads to a class imbalance of queried samples during the active learning process. To address this challenge, we introduce a novel algorithm named `PCB` which rectifies this imbalance by leveraging the knowledge embedded in VLMs before soliciting labels from the oracle labeler. Through extensive experiments across a range of real-world datasets, we demonstrate that our algorithm outperforms conventional active learning methods and surpasses the performance of random sampling. We believe that this framework opens up new avenues for research in the field of active learning within VLMs.

Acknowledgement. The third author was supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2020-0-00862, DB4DL: High-Usability and Performance In-Memory Distributed DBMS for Deep Learning).

References

[1] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. In *Proceedings of International Conference on Computer Vision*, pages 6836–6846, 2021. 1

[2] Jordan T Ash, Chicheng Zhang, Akshay Krishnamurthy, John Langford, and Alekh Agarwal. Deep batch active learning by diverse, uncertain gradient lower bounds. *ArXiv Preprint*, 2019. 1, 4, 5, 6, 8, 13, 14, 16

[3] William H Beluch, Tim Genewein, Andreas Nürnberger, and Jan M Köhler. The power of ensembles for active learning in image classification. In *Proceedings of IEEE/CVF Computer Vision and Pattern Recognition Conference*, pages 9368–9377, 2018. 8

[4] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101–mining discriminative components with random forests. In *Proceedings of European Conference on Computer Vision*, pages 446–461, 2014. 14

[5] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33:1877–1901, 2020. 1, 4

[6] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Proceedings of IEEE/CVF Computer Vision and Pattern Recognition Conference*, pages 3606–3613, 2014. 4, 12

[7] Abhishek Das, Satwik Kottur, Khushi Gupta, Avi Singh, Deshraj Yadav, José MF Moura, Devi Parikh, and Dhruv Batra. Visual dialog. In *Proceedings of IEEE/CVF Computer Vision and Pattern Recognition Conference*, pages 326–335, 2017. 8

[8] Harm De Vries, Florian Strub, Jérémie Mary, Hugo Larochelle, Olivier Pietquin, and Aaron C Courville. Modulating early visual processing by language. *Advances in Neural Information Processing Systems*, 30, 2017. 8

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of IEEE/CVF Computer Vision and Pattern Recognition Conference*, pages 248–255, 2009. 14

[10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *ArXiv Preprint*, 2020. 1, 3

[11] Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *Proceedings of IEEE/CVF Computer Vision and Pattern Recognition Conference Workshop*, pages 178–178, 2004. 4, 13

[12] Yarin Gal, Riashat Islam, and Zoubin Ghahramani. Deep bayesian active learning with image data. In *Proceedings of International Conference on Machine Learning*, pages 1183–1192, 2017. 1, 8

[13] Zhe Gan, Yen-Chun Chen, Linjie Li, Chen Zhu, Yu Cheng, and Jingjing Liu. Large-scale adversarial training for vision-and-language representation learning. *Advances in Neural Information Processing Systems*, 33:6616–6628, 2020. 8

[14] Tianyu Gao, Adam Fisch, and Danqi Chen. Making pre-trained language models better few-shot learners. *ArXiv Preprint*, 2020. 3, 8

[15] Yonatan Geifman and Ran El-Yaniv. Deep active learning with a neural architecture search. *Advances in Neural Information Processing Systems*, 32, 2019. 8

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of IEEE/CVF Computer Vision and Pattern Recognition Conference*, pages 770–778, 2016. 1, 3

[17] Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(7):2217–2226, 2019. 4, 12

[18] Alex Holub, Pietro Perona, and Michael C Burl. Entropy-based active learning for object recognition. In *Proceedings of IEEE/CVF Computer Vision and Pattern Recognition Conference Workshop*, pages 1–8, 2008. 1, 4, 5, 8, 13, 14, 16

[19] Neil Houlsby, Ferenc Huszár, Zoubin Ghahramani, and Máté Lengyel. Bayesian active learning for classification and preference learning. *ArXiv Preprint*, 2011. 1, 8

[20] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *Proceedings of International Conference on Machine Learning*, pages 4904–4916, 2021. 8

[21] Zhengbao Jiang, Frank F Xu, Jun Araki, and Graham Neubig. How can we know what language models know? *Transactions of the Association for Computational Linguistics*, 8: 423–438, 2020. 3, 8

[22] Chen Ju, Tengda Han, Kunhao Zheng, Ya Zhang, and Weidi Xie. Prompting visual-language models for efficient video understanding. In *Proceedings of European Conference on Computer Vision*, pages 105–124, 2022.

[23] Muhammad Uzair Khattak, Hanoona Rasheed, Muhammad Maaz, Salman Khan, and Fahad Shahbaz Khan. Maple: Multi-modal prompt learning. In *Proceedings of IEEE/CVF Computer Vision and Pattern Recognition Conference*, pages 19113–19122, 2023. 1, 3, 7, 8

[24] Muhammad Uzair Khattak, Syed Talal Wasim, Muzammal Naseer, Salman Khan, Ming-Hsuan Yang, and Fahad Shahbaz Khan. Self-regulating prompts: Foundational model adaptation without forgetting. In *Proceedings of International Conference on Computer Vision*, pages 15190–15200, 2023. 8

[25] Andreas Kirsch, Joost Van Amersfoort, and Yarin Gal. Batchbald: Efficient and diverse batch acquisition for deep bayesian active learning. *Advances in Neural Information Processing Systems*, 32, 2019. 1, 8

[26] Christine Körner and Stefan Wrobel. Multi-class ensemble-based active learning. In *Proceedings of European Conference on Machine Learning*, pages 687–694, 2006. 8

[27] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of International Conference on Computer Vision Workshop*, pages 554–561, 2013. 4, 13

[28] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. *ArXiv Preprint*, 2021. 3, 8

[29] David D Lewis and Jason Catlett. Heterogeneous uncertainty sampling for supervised learning. In *Machine Learning Proceedings 1994*, pages 148–156. Elsevier, 1994. 8

[30] Gen Li, Nan Duan, Yuejian Fang, Ming Gong, and Daxin Jiang. Unicoder-vl: A universal encoder for vision and language by cross-modal pre-training. In *Proceedings of AAAI Conference on Artificial Intelligence*, pages 11336–11344, 2020. 8

[31] Junnan Li, Ramprasaath Selvaraju, Akhilesh Gotmare, Shafiq Joty, Caiming Xiong, and Steven Chu Hong Hoi. Align before fuse: Vision and language representation learning with momentum distillation. *Advances in Neural Information Processing Systems*, 34:9694–9705, 2021. 1, 8

[32] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *Proceedings of International Conference on Machine Learning*, pages 12888–12900, 2022. 8

[33] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *ArXiv Preprint*, 2023. 1, 8

[34] Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. *ArXiv Preprint*, 2021. 3, 8

[35] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *Advances in Neural Information Processing Systems*, 32, 2019. 8

[36] Yuning Lu, Jianzhuang Liu, Yonggang Zhang, Yajing Liu, and Xinmei Tian. Prompt distribution learning. In *Proceedings of IEEE/CVF Computer Vision and Pattern Recognition Conference*, pages 5206–5215, 2022. 3, 8

[37] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *ArXiv Preprint*, 2013. 4, 13

[38] Prem Melville and Raymond J Mooney. Diverse ensembles for active learning. In *Proceedings of International Conference on Machine Learning*, pages 584–591, 2004. 8

[39] Sachit Menon and Carl Vondrick. Visual classification via description from large language models. In *Proceedings of International Conference on Learning Representations*, 2023. 4, 8

[40] Prateek Munjal, Nasir Hayat, Munawar Hayat, Jamshid Sourati, and Shadab Khan. Towards robust and reproducible active learning using neural networks. In *Proceedings of IEEE/CVF Computer Vision and Pattern Recognition Conference*, pages 223–232, 2022. 8

[41] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *Proceedings of Indian Conference on Computer Vision, Graphics & Image Processing*, pages 722–729, 2008. 4, 12

[42] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *Proceedings of IEEE/CVF Computer Vision and Pattern Recognition Conference*, pages 3498–3505, 2012. 4, 12

[43] Amin Parvaneh, Ehsan Abbasnejad, Damien Teney, Gholamreza Reza Haffari, Anton Van Den Hengel, and Javen Qinfeng Shi. Active learning by feature mixing. In *Proceedings of IEEE/CVF Computer Vision and Pattern Recognition Conference*, pages 12237–12246, 2022. 1, 8

[44] 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 International Conference on Computer Vision*, pages 15691–15701, 2023. 4, 8

[45] Di Qi, Lin Su, Jia Song, Edward Cui, Taroon Bharti, and Arun Sacheti. Imagebert: Cross-modal pre-training with large-scale weak-supervised image-text data. *ArXiv Preprint*, 2020. 8

[46] 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 *Proceedings of International Conference on Machine Learning*, pages 8748–8763, 2021. 1, 2, 4, 5, 8, 12

[47] Vineeth Rakesh and Swayambhoo Jain. Efficacy of bayesian neural networks in active learning. In *Proceedings of IEEE/CVF Computer Vision and Pattern Recognition Conference*, pages 2601–2609, 2021. 1

[48] Pengzhen Ren, Yun Xiao, Xiaojun Chang, Po-Yao Huang, Zhihui Li, Brij B Gupta, Xiaojiang Chen, and Xin Wang. A survey of deep active learning. *ACM Computing Surveys (CSUR)*, 54(9):1–40, 2021. 1, 8

[49] Dan Roth and Kevin Small. Margin-based active learning for structured output spaces. In *Proceedings of European Conference on Machine Learning*, pages 413–424, 2006. 8

[50] Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. In *Proceedings of International Conference on Learning Representations*, 2018. 1, 4, 5, 8, 13, 14, 16

[51] Burr Settles. Active learning literature survey. Technical report, Department of Computer Sciences, University of Wisconsin-Madison, 2009. 1, 8

[52] Taylor Shin, Yasaman Razeghi, Robert L Logan IV, Eric Wallace, and Sameer Singh. Autoprompt: Eliciting knowledge from language models with automatically generated prompts. *ArXiv Preprint*, 2020. 3, 8

[53] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *ArXiv Preprint*, 2012. 14

[54] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *ArXiv Preprint*, 2023. 1

[55] 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*, 2023. 1

[56] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 2017. 3

[57] Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *Proceedings of IEEE/CVF Computer Vision and Pattern Recognition Conference*, pages 3485–3492, 2010. 14

[58] Jingfeng Yang, Aditya Gupta, Shyam Upadhyay, Luheng He, Rahul Goel, and Shachi Paul. Tableformer: Robust transformer modeling for table-text encoding. *ArXiv Preprint*, 2022. 1

[59] Lewei Yao, Runhui Huang, Lu Hou, Guansong Lu, Minzhe Niu, Hang Xu, Xiaodan Liang, Zhenguo Li, Xin Jiang, and Chunjing Xu. Filip: Fine-grained interactive language-image pre-training. *ArXiv Preprint*, 2021. 1, 8

[60] Fei Yu, Jiji Tang, Weichong Yin, Yu Sun, Hao Tian, Hua Wu, and Haifeng Wang. Ernie-vil: Knowledge enhanced vision-language representations through scene graphs. In *Proceedings of AAAI Conference on Artificial Intelligence*, pages 3208–3216, 2021. 8

[61] Tao Yu, Zhihe Lu, Xin Jin, Zhibo Chen, and Xinchao Wang. Task residual for tuning vision-language models. In *Proceedings of IEEE/CVF Computer Vision and Pattern Recognition Conference*, pages 10899–10909, 2023. 8

[62] Lu Yuan, Dongdong Chen, Yi-Ling Chen, Noel Codella, Xiyang Dai, Jianfeng Gao, Houdong Hu, Xuedong Huang, Boxin Li, Chunyuan Li, et al. Florence: A new foundation model for computer vision. *ArXiv Preprint*, 2021. 8

[63] Seongjun Yun, Minbyul Jeong, Raehyun Kim, Jaewoo Kang, and Hyunwoo J Kim. Graph transformer networks. *Advances in Neural Information Processing Systems*, 32, 2019. 1

[64] Xiaohua Zhai, Xiao Wang, Basil Mustafa, Andreas Steiner, Daniel Keysers, Alexander Kolesnikov, and Lucas Beyer. Lit: Zero-shot transfer with locked-image text tuning. In *Proceedings of IEEE/CVF Computer Vision and Pattern Recognition Conference*, pages 18123–18133, 2022. 8

[65] Zexuan Zhong, Dan Friedman, and Danqi Chen. Factual probing is [mask]: Learning vs. learning to recall. *ArXiv Preprint*, 2021. 3, 8

[66] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *Proceedings of IEEE/CVF Computer Vision and Pattern Recognition Conference*, pages 16816–16825, 2022. 1, 7, 8

[67] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 130(9):2337–2348, 2022. 1, 3, 6, 7, 8, 12