ArGue: Attribute-Guided Prompt Tuning for Vision-Language Models

Supplementary Material

A. Prompting Large Language Models

The cornerstone of our contributions lies in the creation of class-specific attributes using LLMs. In this section, we offer a comprehensive insight into our attribute generation process. In our experimental setup, we systematically produce a set of $J = 15$ attributes for each class, constituting an attribute pool. Concretely, we leverage 3 distinct LLM templates, with each template yielding 5 attributes. Our approach to attribute generation involves employing in-context learning, wherein we initially present two example questions and then prompt the model to respond to a third query [11]. Furthermore, for each inquiry, we maintain a maximum token length of 200, while setting the temperature parameter to 0.8.

Template 1
Q: Describe what an animal giraffe looks like in a photo, list 6 pieces?
A: There are 6 useful visual features for a giraffe in a photo:
- covered with a spotted coat
- has a short, stocky body
- has a long neck
- owns a small neck to its body
- is yellow or brown in color
- have a black tufted tail

Q: Describe what an equipment laptop looks like in a photo, list 4 pieces?
A: There are 4 useful visual features for a laptop in a photo:
- has a built-in touchpad below the keyboard
- has a black screen
- attached with charging ports
- owns a QWERTY keyboard

Q: Describe what a {type} {class} looks like in a photo, list {num} pieces?
A: There are {num} useful visual features for a {class} in a photo:
-

Template 2
Q: Visually describe a giraffe, a type of animal, list 6 pieces?
A: There are 6 useful visual features for a giraffe in a photo:
- covered with a spotted coat
- has a short, stocky body
- has a long neck
- owns a small neck to its body
- is yellow or brown in color
- have a black tufted tail

Q: Visually describe a laptop, a type of equipment, list 4 pieces?
A: There are 4 useful visual features for a laptop in a photo:
- has a built-in touchpad below the keyboard
- has a black screen
- attached with charging ports
- owns a QWERTY keyboard

Q: Visually describe a {class}, a type of {type}, list {num} pieces?
A: There are {num} useful visual features for a {class} in a photo:
-

Template 3
Q: How to distinguish a giraffe which is an animal, list 6 pieces?
A: There are 6 useful visual features for a giraffe in a photo:
- covered with a spotted coat
- has a short, stocky body
- has a long neck
- owns a small neck to its body
- is yellow or brown in color
- have a black tufted tail

Q: How to distinguish a laptop which is an equipment, list 4 pieces?
A: There are 4 useful visual features for a laptop in a photo:
- has a built-in touchpad below the keyboard
- has a black screen
- attached with charging ports
- owns a QWERTY keyboard

Q: How to distinguish a {class} which is a {type}, list {num} pieces?
A: There are {num} useful visual features for a {class} in a photo:
-

{class} signifies the class name, and {type} represents a generic class type specific to the dataset, *e.g.*, pet for Ox-

Dataset	Class name	Attr. 1	Attr. 2	Attr. 3	Attr. 4	Attr. 5	Attr. 6	Attr. 7	Attr. 8	Attr. 9	Attr. 10	All	Top 3
EuroSAT	Industrial Buildings	86.42	88.52	85.06	89.13	87.18	89.73	90.29	87.46	87.44	86.07	89.61	90.48
	Annual Crop Land	91.53	89.15	92.47	91.05	87.80	88.49	89.34	91.22	92.37	91.92	91.14	92.72
UCF101	Blowing Candles	79.29	81.88	76.36	78.49	80.42	79.91	82.89	80.70	79.22	78.48	81.16	82.47
	Basketball Dunk	81.86	82.11	80.87	78.81	81.65	79.47	82.41	80.13	82.39	81.37	81.42	82.50
Food101	Chicken Quesadilla	93.76	92.34	93.41	91.21	92.39	93.20	91.75	90.81	93.80	91.72	92.34	93.81
	Breakfast Burrito	90.80	91.56	89.68	90.67	91.68	90.47	90.71	91.84	90.74	89.23	91.34	91.65

Table 1. **The results of ArGue concerning the selection of different attribute sets.** We conduct experiments using 10 attributes generated by 2 LLM templates for each class. The selection scenarios include: 1) choosing a single attribute iteratively, 2) selecting all attributes, and 3) choosing the top 3 attributes based on the performance of single attribute training. The model is trained on the selected attributes, and the evaluation is based on the average accuracy across both the base and novel classes. Note that here we only evaluate the accuracy of the listed class. The results underlined indicate the top 3 attributes.

fordPets [13]. This distinction serves to mitigate potential ambiguity in cases of polysemy [14], *e.g.*, bank which can refer to either a financial institution or a geographical location. The parameter {num} indicates the desired number of attributes we instruct the language model to generate. Upon generating the attribute pool, we perform attribute sampling, selecting only 3 attributes for the training process.

B. Example Generated Attributes

In this section, we present examples of attributes generated by LLMs. We have randomly selected one class from ImageNet [5] and one class from Flowers102 [12] to represent both the general classification and fine-grained classification, respectively. The attributes highlighted in green are the ones selected through attribute sampling for training. It's important to note that a complete textual prompt for the text encoder should include the following format: {template} {class} {attr} rather than only listing the attributes themselves. In prompt tuning, the template is replaced by soft tokens.

A photo of a tiger cat which
is covered in stripes of orange, black, and white
has a long, thick coat of fur
has a medium-sized body
has orange or red tones
has large, pointed ears
has round, yellow eyes
has a long, thick tail
has a pointed muzzle
has a short muzzle
has a spotted fur
has a broad head
has sharp claws

A photo of a oxeye daisy which
has a broader, much-divided, and toothed leaves
petals are arranged in a flat, circular shape
blooms a single flower in the late spring
exhibits white petals around the center
grows in abundance in meadows
has a broad, flat flower head
grows in grassland habitats
has a waxy, papery texture
has an invigorating scent
prefers sunny, dry places
has bright yellow center
has a sturdy, thick stem
grows up to 30 cm tall
has short, hollow stem
has leafy green stems

C. Attribute Study

In this section, we validate the motivation behind attribute sampling, which is our belief that certain attributes in the attribute pool are more semantically relevant than others to the images and thus more crucial. We randomly select 2 classes from EuroSAT [6], UCF101 [15], and Food101 [1], generating 10 attributes for each class. This entails employing 2 LLM templates, with each template yielding 5 attributes. Table 1 demonstrates the results when different sets of attributes are selected for training.

Some attributes are much better than others. It is evident from our observations that the choice of attributes significantly impacts the model's accuracy. For instance, in the case of the Industrial Buildings class in EuroSAT, Attr. 7 outperforms Attr. 3 by a substantial margin of 5.23%. This observation highlights the unequal importance of various attributes in the training process, indicating that specific attributes may provide more advantages in enhancing the model's performance.

Combining useful attributes enhances the performance.

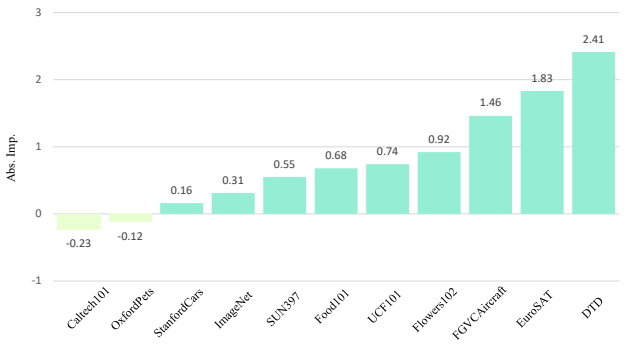

Figure 1. **The absolute improvement of manual labeling compared with LLMs on novel class prediction.** We randomly select 10 classes from each benchmark dataset for simplicity, *i.e.*, 5 base classes and 5 novel classes. The accuracy is calculated solely based on the selected classes. It is worth noting that we have omitted the incorporation of negative prompting in the comparison, and attribute sampling has not been applied in the context of manual labeling.

When we endeavor to train the model by combining the top three attributes based on the single attribute training, although this straightforward combination doesn't efficiently eliminate redundant attributes as attribute sampling does, we observe that the model's accuracy exceeds that of using all attributes and consistently outperforms the best results achieved with single attribute training. This finding lays a practical groundwork for attribute sampling.

D. Manual Labeling vs LLMs

In light of the previous attribute study, we demonstrate that distinct attributes can exert a significant influence on the model accuracy. This section delves deeper into exploring the performance boundaries of ArGue, while also raising questions about the potential for further improvement in the attributes generated by LLMs. As part of this investigation, we manually annotate attributes for 10 classes randomly selected from benchmark datasets and conduct a comparative analysis with attributes generated by LLMs. Following the setting in previous experiments, we annotate 3 attributes for each class. We declare that manual labeling is not considered the main contribution of this article, due to its impracticality in scenarios characterized by complex dataset distributions or a high number of classes. Our primary objective here is to illustrate that ArGue can unleash greater potential when equipped with more precise and semantically relevant attributes.

Manual labeling demonstrates a more pronounced advantage on specialized datasets. Fig. 1 presents a comparison of model accuracy when manual labeling is employed versus the use of LLMs. Notably, for commonly encountered categories, *e.g.*, OxfordPets, ImageNet, manual labeling does not exhibit substantial deviations from LLM-generated attributes. However, the distinct advantage of manual labeling becomes evident when dealing with less prevalent datasets such as satellite imagery (*e.g.*, EuroSAT) and textures (*e.g.*, DTD [4]), resulting in an average performance increase of around 2%. This discrepancy is comprehensible as LLMs lack pre-training data specific to such datasets, rendering them less proficient in providing precise attribute descriptions.

Figure 2. **ColoredMNIST.**

CoOp	CoOp + Van. Neg.	CoOp + Man. Neg.
78.32	86.01	**92.19**

Table 2. **The test accuracy on the subpopulation shift.** We compare among CoOp, CoOp with vanilla negative prompting (Van. Neg.), and CoOp with manual negative prompting (Man. Neg.).

There is still room for improvement in generating attributes using LLMs. In summary, manual labeling outperforms LLMs on 9 out of 11 datasets. This implies that, despite the application of attribute sampling, attributes generated by LLMs are generally less accurate than those obtained through manual labeling. This can be attributed to 1) LLMs lack direct access to images, making it challenging to generate dataset-specific attributes, and 2) LLMs may have inherent biases in their understanding of classes. We believe that exploring more effective ways to generate large-scale, high-quality attributes through LLMs is a promising direction for future research.

E. Negative Prompt Engineering

In this section, we delve into the intriguing concept of designing an effective negative prompt. In prior sections, we introduce a practical assumption wherein we set the negative attribute merely to background instead of specifying a particular dataset. This approach offers the advantage of obviating the requirement for extra manual labeling. Our empirical investigations have indicated the efficacy of this strategy across a majority of datasets.

Nonetheless, it is apparent that this approach necessitates further examination, especially when dealing with specific datasets. For example, datasets such as DTD or EuroSAT exhibit spurious correlations that do not originate from the

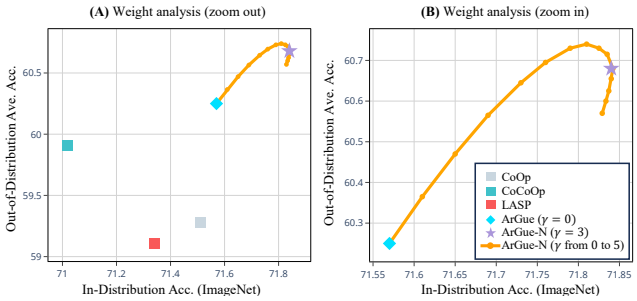

Figure 3. **The accuracy of ID, *i.e.*, ImageNet, and OOD, *i.e.*, four variant datasets, while varying γ compared with baselines.** The OOD accuracy is averaged over four datasets.

	Dataset	CoOp	CoCoOp	LASP	ArGue	ArGue-N
ID	ImageNet	71.51	71.02	71.34	71.57	**71.84**
OOD	Caltech101	93.70	94.43	93.87	**94.78**	94.30
	OxfordPets	89.14	90.14	91.74	93.43	**93.75**
	StanfordCars	64.51	65.32	68.16	69.85	**70.48**
	Flowers102	68.71	71.88	71.18	**72.11**	72.07
	Food101	85.30	86.06	89.71	89.93	**90.41**
	FGVCAircraft	18.47	22.94	28.15	31.70	**32.90**
	SUN397	64.15	67.36	65.44	69.23	**72.46**
	DTD	41.92	45.73	59.03	57.34	**60.06**
	EuroSAT	46.39	45.37	72.79	81.57	**82.46**
	UCF101	66.55	68.21	70.98	72.09	**72.76**

Table 3. **The results on cross-dataset transfer.** The model is trained on ImageNet, and evaluated on 10 entirely different datasets.

image background. In such scenarios, the general negative prompt may not effectively mitigate incorrect rationales. Hence, within this section, we delve into the prospect of devising an interpretable and more contextually suitable negative prompt tailored to a dataset. Furthermore, our objective is to illustrate that the scope and efficacy of negative prompting extend beyond a singular, predefined prompt.

We create the ColoredMNIST dataset, which, alongside the handwritten digit labels ranging from 0 to 9, incorporates a distinctive background color assigned to each label in the training set. Empirically, conventional prompt tuning exhibits a propensity to acquire spurious correlations between colors and labels, thereby deviating from the primary objective of recognizing digit shapes. In the test set, we introduce subpopulation shift by randomly associating 10 different colors with the 10 labels. Fig. 2 provides a visual representation of the images corresponding to each label, accompanied by their respective background colors. We establish two baseline methods: CoOp [17], *i.e.*, vanilla prompt tuning, and CoOp with vanilla negative prompting, which exclusively utilizes the general negative prompt, *i.e.*, the background of a {digit}. Additionally, we develop 10 negative prompts tailored for each class manually. These customized negative prompts are structured to encompass the specific colors associated with the labels, *e.g.*, the green background of a zero or the purple background of a three. In essence, beyond employing a general attribute, we introduce more precise specifications for addressing the spurious correlations within each class. It's worth noting that, for a fair comparison with vanilla prompt tuning, in this experiment, we exclusively utilize negative prompting without employing any additional class-specific attributes for attribute-guided prompt tuning. In other words, our experiment is solely based on CoOp implementation.

Table 2 presents a comparison between CoOp, CoOp with vanilla negative prompting, and CoOp with manual negative prompting. It is evident that merely using the background as the negative attribute results in an approximately 8% increase compared to CoOp. Furthermore, employing class-wise attributes, *i.e.*, specifying the background color for each class, leads to an additional 6% improvement. While this synthetic dataset leans toward the ideal side due to its highly apparent and easy-to-specify spurious correlations, it also indicates that negative prompting holds greater potential when enhanced prior knowledge and more speci-

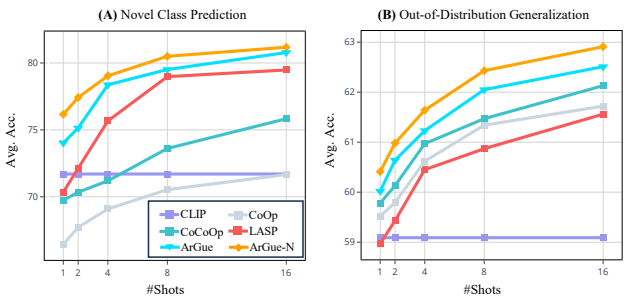

Figure 4. **The results for two tasks with varying numbers of shots.** It is important to note that, given CLIP's status as a pre-trained model, its performance remains constant regardless of the number of shots. The average accuracy denotes the mean performance results aggregated from all datasets within the current task.

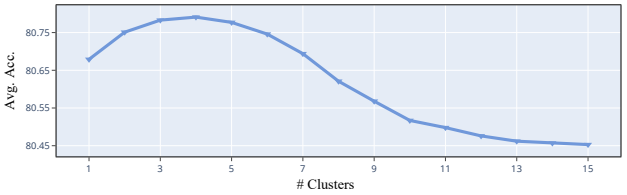

Figure 5. **The results of ArGue on novel class prediction with varying cluster numbers, *i.e.*, N from 1 to 15.** The accuracy is averaged over all the benchmark datasets across base and novel classes.

Baseline	$\beta = 0$	$\beta = 5$	$\beta = 10$	$\beta = 20$	$\beta = 50$	$\beta = 100$
LASP	59.85	61.43	**61.85**	61.56	59.32	59.04
ArGue-N	61.24	**62.94**	62.86	62.91	62.25	61.53

Table 4. The average results on the OOD task while varying β.

fications are available. We believe that designing effective negative prompts is a promising area for future research.

F. Cross-Dataset Transfer

In this section, we assess ArGue and contemporary state-of-the-art methods on a more demanding task, namely, cross-dataset transfer. This task involves training the model on an in-distribution dataset and evaluating its performance on entirely different datasets, making it more challenging but indicative of broader potential. The results for this task are presented in Table 3.

G. Prompting Weight Analysis

In our empirical findings, we have observed that the weight of negative prompting in the loss function, *i.e.*, γ, exerts a substantial influence on the training performance. This section is dedicated to a comprehensive analysis of the relationship between γ and the experimental outcomes.

Fig. 3 illustrates the performance of ArGue-N in the OOD generalization task as the value of gamma varies from 0 to 5. Commencing at $\gamma = 0$, representative of ArGue, the model's accuracy in both ID and OOD datasets exhibits gradual improvement as γ increases. This progression signifies the model's effective transition from concentrating on spurious correlations to intrinsic semantics. When γ reaches 3, the model achieves its highest ID accuracy. However, further increments in γ lead to a decline in OOD accuracy. This phenomenon is comprehensible because, at this stage, the loss associated with negative prompting becomes disproportionately significant, causing the model to overlook the minimization of the original classification loss, ultimately resulting in underfitting. Empirically, we conclude that the optimal range for γ lies between 2.5 and 3.5.

H. Cluster Number Analysis

Attribute sampling indicates that it is not necessary to utilize the entire set of attributes within the attribute pool. Rather, employing a small subset is adequate to achieve or even surpass the performance of using all attributes. Nevertheless, determining the optimal proportion of this subset involves a trade-off. Choosing too few attributes may result in an insufficient semantic component of the class, while an excessive number of attributes can lead to redundancy, causing computational burdens or introducing ineffective attributes. In this section, we delve into the discussion of identifying the optimal proportion for this small subset. Specifically, based on the outcomes of previous experiments, we generate 15 attributes for each class, constituting an attribute pool. We linearly vary the cluster number, *i.e.*, N, from 1 to 15 and evaluate its performance in the context of the novel class prediction task. It is noteworthy that, taking into account the distinctive characteristics of classes, a potentially more effective strategy involves determining an optimal cluster number for each class, *i.e.*, N_c. While this expands the search space, potentially yielding enhanced results, it also introduces additional computational complexity. We leave the exploration of this approach to future work.

Set	ProDA	PLOT	PBPrompt	MaPLe	ALIGN	ArGue	ArGue-N
Base	81.56	82.46	80.88	82.28	83.38	83.69	**83.77**
New	72.30	72.53	74.74	75.14	75.51	78.07	**78.74**
H	76.65	77.18	77.69	78.55	79.25	80.83	**81.22**

Table 5. The average results of base & new acc. over 11 datasets on more state-of-the-art methods.

Fig. 5 illustrates the results of ArGue in the context of novel class prediction. For simplicity, negative prompting is omitted in the context. From the figure, it is evident that the accuracy notably increases as the cluster number ranges from 1 to 3. This phenomenon is ascribed to the meticulous selection of attributes within this range, emphasizing their semantic relevance and representativeness. At the inflection point of 4, with the continued increase in the cluster number, a gradual decline in accuracy is observed due to the influence of certain ineffective attributes. As the cluster number reaches 15, attribute sampling is entirely inoperative, causing ArGue to degrade to vanilla attribute-guided prompt tuning with regularization. Given the above observations, we posit that a cluster number of 3 or 4 is the most suitable choice. Since we aim to minimize the number of attributes to reduce computational burden, $N = 3$ is preferred.

I. Further Comparison

Varying shots. In this section, we present a comparison of our model's performance with different shot numbers in contrast to various baselines. Fig. 4 showcases the performance of our method and the baselines at 1, 2, 4, 8, and 16 shots. As depicted, there is a notable trend of improved accuracy across most methods as the number of shots increases. Notably, ArGue-N consistently outperforms the other methods, and this advantage is most prominent when the number of shots is limited.

Varying β. Considering that prompt regularization has been studied in [2], our choice of β is following their setup for fairness. To study the impact of β, we select [2] as the baseline and compare the results while varying β in Table 4.

Notably, we observe that optimal performance is achieved when β ranges between 5 and 20. This empirical finding aligns with the experimental setup in [2], where they use $\beta = 20$.

Other baselines.. We also compare our method with other state-of-the-art methods encompassing ProDA [9], PLOT [3], PBPrompt [8], MaPLe [7] and ALIGN [16]. The results are displayed in Table 5.

J. Limitation Analysis

In this section, we outline the limitations of our work, providing several potential avenues for future research in the field.

Relative Discriminative Attributes. Attribute sampling enables us to select attributes from a class's attribute pool that are both representative and highly semantically relevant to the associated images. Nonetheless, in a classification context, it is crucial to consider the interrelationships between attributes across different classes. Take, for instance, the FGVCAircraft [10] classification task, where we observe that LLMs often produce similar attributes for distinct classes. This phenomenon arises because each class serves as a subcategory within the broader "aircraft" category, sharing numerous common features. When these common attributes are shared across all classes in the dataset, it becomes arduous to employ them effectively for class differentiation. Attributes that can uniquely discriminate one class from others are denoted as relative discriminative attributes signifying that other classes lack these particular attributes. We posit that relative discriminative attributes offer a more robust characterization of individual classes, and exploring methods for their selection represents a potential avenue for future research.

Attribute Quality of LLMs. Manually annotating attributes for each class is a resource-intensive and time-consuming task. Nevertheless, our prior comparative analysis between human-generated annotations and the attributes produced by LLMs has underscored the fact that LLMs still have room for improvement in generating accurate and exhaustive attributes. We are optimistic that as LLMs continue to advance at a rapid pace, our approach will inherently gain from these developments, potentially yielding more substantial advancements.

References

[1] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101 - mining discriminative components with random forests. In *ECCV (6)*, pages 446–461. Springer, 2014. 2

[2] Adrian Bulat and Georgios Tzimiropoulos. LASP: text-to-text optimization for language-aware soft prompting of vision & language models. In *CVPR*, pages 23232–23241. IEEE, 2023. 5, 6

[3] Guangyi Chen, Weiran Yao, Xiangchen Song, Xinyue Li, Yongming Rao, and Kun Zhang. PLOT: prompt learning with optimal transport for vision-language models. In *ICLR*. OpenReview.net, 2023. 6

[4] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *CVPR*, pages 3606–3613. IEEE Computer Society, 2014. 3

[5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255. IEEE Computer Society, 2009. 2

[6] 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 J. Sel. Top. Appl. Earth Obs. Remote. Sens.*, 12(7):2217–2226, 2019. 2

[7] Muhammad Uzair Khattak, Hanoona Abdul Rasheed, Muhammad Maaz, Salman H. Khan, and Fahad Shahbaz Khan. Maple: Multi-modal prompt learning. In *CVPR*, pages 19113–19122. IEEE, 2023. 6

[8] Xinyang Liu, Dongsheng Wang, Miaoge Li, Zhibin Duan, Yishi Xu, Bo Chen, and Mingyuan Zhou. Patch-token aligned bayesian prompt learning for vision-language models. *CoRR*, abs/2303.09100, 2023. 6

[9] Yuning Lu, Jianzhuang Liu, Yonggang Zhang, Yajing Liu, and Xinmei Tian. Prompt distribution learning. In *CVPR*, pages 5196–5205. IEEE, 2022. 6

[10] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew B. Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *CoRR*, abs/1306.5151, 2013. 6

[11] Sachit Menon and Carl Vondrick. Visual classification via description from large language models. In *ICLR*. OpenReview.net, 2023. 1

[12] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *ICVGIP*, pages 722–729. IEEE Computer Society, 2008. 2

[13] Omkar M. Parkhi, Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. Cats and dogs. In *CVPR*, pages 3498–3505. IEEE Computer Society, 2012. 2

[14] Karsten Roth, Jae Myung Kim, A. Sophia Koepke, Oriol Vinyals, Cordelia Schmid, and Zeynep Akata. Waffling around for performance: Visual classification with random words and broad concepts. In *ICCV*, pages 15746–15757, 2023. 2

[15] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. A dataset of 101 human action classes from videos in the wild. *Center for Research in Computer Vision*, 2(11), 2012. 2

[16] Dongsheng Wang, Miaoge Li, Xinyang Liu, Mingsheng Xu, Bo Chen, and Hanwang Zhang. Tuning multi-mode token-level prompt alignment across modalities. In *NeurIPS*, 2023. 6

[17] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *Int. J. Comput. Vis.*, 130(9):2337–2348, 2022. 4