Descriptor and Word Soups: Overcoming the Parameter Efficiency Accuracy Tradeoff for Out-of-Distribution Few-shot Learning

Christopher Liao
Boston University
cliao25@bu.edu

Theodoros Tsiligkaridis
MIT Lincoln Laboratory
ttsili@ll.mit.edu

Brian Kulis
Boston University
bkulis@bu.edu

Abstract

Over the past year, a large body of multimodal research has emerged around zero-shot evaluation using GPT descriptors. These studies boost the zero-shot accuracy of pretrained VL models with an ensemble of label-specific text generated by GPT. A recent study, WaffleCLIP, demonstrated that similar zero-shot accuracy can be achieved with an ensemble of random descriptors. However, both zero-shot methods are un-trainable and consequently sub-optimal when some few-shot out-of-distribution (OOD) training data is available. Inspired by these prior works, we present two more flexible methods called ***descriptor and word soups****, which do not require an LLM at test time and can leverage training data to increase OOD target accuracy. Descriptor soup greedily selects a small set of textual descriptors using generic few-shot training data, then calculates robust class embeddings using the selected descriptors. Word soup greedily assembles a chain of words in a similar manner. Compared to existing few-shot soft prompt tuning methods, word soup requires fewer parameters by construction and less GPU memory, since it does not require backpropagation. Both soups outperform current published few-shot methods, even when combined with SoTA zero-shot methods, on cross-dataset and domain generalization benchmarks. Compared with SoTA prompt and descriptor ensembling methods, such as ProDA and WaffleCLIP, word soup achieves higher OOD accuracy with fewer ensemble members. Please checkout our code:* `github.com/Chris210634/word_soups`

1. Introduction

Problem Setting There is extensive interest from the computer vision community for training classifiers that are robust to distribution shifts. Pioneering works in this area [23, 47, 67] focused on optimizing for simple shifts in the image distribution, such as sketch-to-real adaptation. As the topic evolved, the community proposed increasingly harder adaptation problems by eliminating some restrictive assumptions. For the *domain generalization (DG)* problem [55, 68], we do not assume access to unlabeled target data; for the *cross-dataset generalization (XD)* problem [70], we allow source and target label spaces to be different; and for the *parameter efficient learning (PEFT)* problem [20, 37, 57], we impose a tight budget on the number of parameters that can be tuned. Our work lies at the confluence of these three topics. Similar to CoOp [70] and MaPLe [25], we do assume access to labeled few-shot generic source data, such as ImageNet. Since we assume nothing about the relationship between source and target datasets, this setting can be more useful in practice than strict zero-shot learning. In this paper, we propose two parameter efficient few-shot methods, called word and descriptor soups, that finetune vision-language (VL) models to generalize to target datasets which may contain unseen labels and/or shifts in the image distribution. Our methods achieve state-of-the-art on some benchmarks without additional gradient-based tuning, but can also improve state-of-the-art gradient-based finetuning methods with an additional diversity loss.

Motivation Our work is motivated by the recent success of classification by description methods [24, 35, 42] in both zero-shot (ZS) classification and open-vocabulary object detection. These methods ask an LLM like GPT to generate a list of short descriptions for each class, then aggregate predictions from the descriptions to improve ZS accuracy, see Fig. 1(a). It is often claimed that the impressive gain in ZS accuracy comes from additional information given by the GPT descriptions. However, a recent study called WaffleCLIP [45] observed that random descriptors or even strings of random words can achieve similar ZS accuracy to GPT descriptors, when ensembled together (see Fig. 5). Therefore, gains in ZS accuracy achieved by descriptor methods are mostly driven by ensembling rather than the content of the descriptors themselves. Inspired by this observation, we propose descriptor and word soups, two methods which outperform WaffleCLIP by selecting descriptors or chains of words that maximize few-shot accuracy. Word soup has 3 advantages: (1) it outperforms existing descriptor-inspired ZS methods in the few-shot OOD setting since it directly maxi-

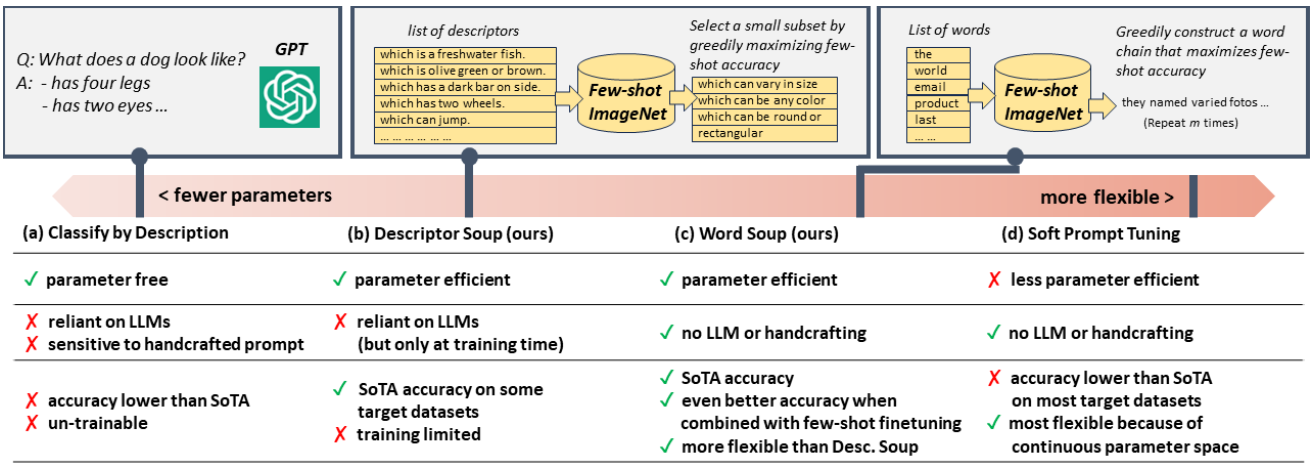

Figure 1. Illustration of word and descriptor soups. We conceptually position our two soup methods along the tradeoff between parameter efficiency and flexibility; we then list the pros and cons of our soups compared to prior work. Firstly, word soup is more parameter efficient than soft prompt tuning, because it uses discrete tokens (see Fig. 2). Secondly, word soup does not require an LLM or handcrafted prompts. Lastly, word soup attains higher target accuracy than prior descriptor methods by allowing a descriptor to be any permutation of words and explicitly maximizing its accuracy on training data (see Fig. 3). However, word soup achieves this flexibility by sacrificing the explainability of descriptors. On the other hand, descriptor soup is interpretable (see Table 1), but less flexible than word soup, since it is limited to selecting from the pool of GPT descriptors.

mizes classification accuracy (see Fig. 1(c)); (2) it is more parameter efficient than existing few-shot methods since the model is frozen and only the discrete descriptor tokens need to be stored; and (3) it does not require an LLM. The pros and cons of both descriptor and word soups are concisely stated in Figure 1 and discussed more in the method section.

Method Overview According to the above motivation, we design a progression of three methods: *descriptor soup*, *word soup*, and *word soup training with diversity loss*. These methods build upon each other but can be used independently and in combination with prior methods. We opted for this style of presentation, since there are motivating empirical insights at each stage, and each method achieves state-of-the-art depending on resource constraints (such as availability of an LLM at training time or parameter storage budget). Descriptor soup is loosely inspired by model soups [58]; "soup" refers to a set of descriptors. We calculate an aggregate prediction based on the centroid of descriptors in the soup. We start with the most accurate descriptor on the training data and greedily add descriptors to the soup if training accuracy increases, see Fig. 1(b). Similarly, for word soups, we assemble a chain of words by greedily appending a word if it increases the training accuracy of the word chain, see Fig. 1(c). Finally, we present a diversity loss that can be used to optimize the CLIP model, using the word soup as an initialization. This loss is required to maintain the initial diversity among word soup members throughout finetuning.

Contributions We make the following contributions to the computer vision literature:

- We present word soup, which improves SoTA on few-shot cross-dataset (XD) and domain-generalization (DG) benchmarks by 1% and 0.8% resp.
- Our word soup uses fewer parameters than SoTA parameter efficient methods while achieving higher accuracy than parameter-free ZS methods in both few-shot settings.
- We propose a diversity loss to train VL models initialized with word soup. This allows our method to seamlessly combine with prior few-shot finetuning methods.
- We present qualitative results (e.g. Tab. 1) to understand what is means for a descriptor to be "good", and analyze the generalizability of these descriptors (Fig. 3). These results extend the current understanding of how descriptor and prompting methods work.

2. Related Work

Few-shot CLIP finetuning We follow the problem settings of CoOp [70], CoCoOp [69], MaPLe [25], and Clipood [51], which finetune a CLIP-like model [43] on few-shot ImageNet in a manner that generalizes to OOD target datasets. Many prompt tuning methods build on top of CoOp by using different loss functions [3, 5, 9, 41, 62], using clever optimization techniques [71], ensembling multiple prompts [6, 31], leveraging different sources of information [12, 22, 49], leveraging synergy between modalities [26, 30, 65], or using different network architectures [8, 61]. We take a fundamentally different approach from these prior methods, drawing inspiration from classification by description [35]. Specifically, prior methods tune a soft prompt while our method tunes a sequence of discrete tokens.

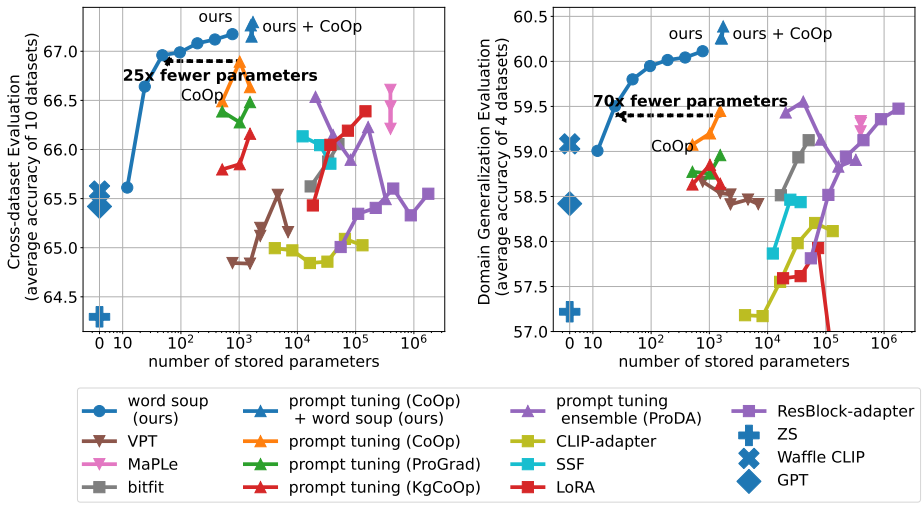

Figure 2. Comparison with PEFT and ZS methods. We vary m for word soup as in Fig. 5. We vary the number of prompt tokens for CoOp, VPT and MaPLe, the number of prompts for ProDA, the rank for LoRA and adapters, and the number of layers tuned for SSF and bitfit. CoOp stores 512 parameters per soft token, while word soup stores 1 parameter per discrete token. Average of 3 runs. Word soup achieves the maximal CoOp accuracy with only 1/25 of the parameters on the XD benchmark and 1/70 of the parameters on the DG benchmark. Detailed results see Tab. 11 in the Appendix.

Zero-shot CLIP Many recent papers use LLM descriptors to aid ZS or open-vocabulary visual tasks, including classification [35, 42] and detection [24]. WaffleCLIP [45] observed that the impressive gains in accuracy reported by these works are mostly driven by ensembling and dataset-level concepts. WaffleCLIP ensembles random descriptors and uses an LLM to discover dataset-level concepts, while we design an optimization procedure to learn good descriptors from data. Our algorithm is loosely related to model averaging methods [58, 59]. However, unlike model soups [58], we do not generate multiple training trajectories, since all descriptors share the same model weights. ZS accuracy can also be improved with hierarchical label sets [38] or handcrafted prompts [1]. Test-time prompt tuning methods [10, 33, 48, 50] train a sample-specific prompt that maximizes agreement between predictions based on a set of image augmentations. These methods suffer from long inference times due to test-time optimization.

Parameter efficient finetuning (PEFT) Our word soup can be considered a PEFT [17, 20] method, but specialised to finetuning VL models in the OOD setting. Prior PEFT methods include shallow text prompt tuning [22, 31, 70, 71], visual prompt tuning [20], bias tuning [64], adapters [11, 16, 39, 54, 66], LoRA [17], SSF [29], side-tuning [53], and others [18, 21, 32, 63]. Unlike the above works, our word soup tunes fewer parameters by leveraging discrete text tokens. Similar to LST [53], we use minimal GPU memory, since no backpropagation is required. We empirically compare with a representative subset of PEFT methods in the OOD settings in Fig. 2. Clearly, our word soup establishes a better tradeoff between parameter efficiency and OOD accuracy, compared to prior work.

3. Method

This section is organized into 4 parts. Section 3.1 reviews the classification by description [35] and WaffleCLIP [45] methods, which motivate our soup methods. Section 3.2 presents descriptor soup, a novel intermediary method which still uses GPT descriptors at training time but not at test time. Section 3.3 presents word soup, which is similarly motivated but only requires a list of English words at training time. Section 3.4 describes the diversity loss used to finetune the CLIP model using word soup as the initialization. Please use Fig. 1 as a reference. We organize the methods in this section in order of increasing flexibility, since it is more natural to motivate word soups this way. However, word soups can also be motivated in the opposite direction by shortcomings of soft prompt tuning, as noted in Fig. 1; this motivation is included in Appendix B. We also propose a token offset trick in Appendix C to augment descriptor soups.

3.1. LLM Descriptors and WaffleCLIP

Several works use LLM descriptors to supplement class names in VL models [24, 35, 42]. These methods ask an LLM to describe the object being classified and incorporate this information into the textual input by forming sentences such as "a photo of a tench, which is a freshwater fish" or "a photo of a goldfish, which has small black eyes". The LLM generates on average 5.8 such descriptors per label, and the centroids of the resulting text embeddings are used for zero-shot classification of images. The improvement in zero-shot accuracy can be attributed to (1) additional information coming from the LLM and (2) ensembling. In WaffleCLIP, Roth et al. [45] claim that most of the gain in accuracy reported by Menon and Vondrick [35] can be attributed to ensembling. They showed that appending a similar number of *randomly selected* descriptors to the class names can achieve similar zero-shot accuracies as the GPT descriptors. We confirm this result in Fig. 5. Observe in this figure that both random descriptors (labeled as "random soup") and chains of random nonsensical words (labeled as "waffle CLIP") perform better than classification by description ("GPT centroids")

Color-coded by **source**: ImageNet, Pets, DTD, Random

Target: ImageNet	Alignment	Accuracy
no descriptor	0.301	67.1
which typically brightly colored.	0.305 (+0.004)	68.2 (+1.1)
which has usually white or off-white.	0.310 (+0.009)	68.4 (+1.3)
which is a long, low-slung body.	0.312 (+0.011)	68.3 (+1.2)
which is a curved or rectangular shape.	0.309 (+0.008)	68.6 (+1.5)
which can vary in size from small to large.	0.315 (+0.014)	68.5 (+1.4)
which has reddish brown fur.	0.300 (-0.001)	66.2 (-0.9)
which is a hard skeleton.	0.295 (-0.006)	66.6 (-0.5)
which is a medium-sized, short-haired cat.	0.291 (-0.010)	66.0 (-1.1)
which has sharp claws.	0.299 (-0.002)	66.6 (-0.5)
which is a repeating pattern.	0.295 (-0.006)	66.1 (-1.0)
which is a sign with the shop's name.	0.295 (-0.006)	66.7 (-0.4)

Target: Pets	Alignment	Accuracy
no descriptor	0.322	88.4
a type of pet. (handcrafted; for reference)	0.331 (+0.009)	89.0 (+0.6)
which is a large, powerful cat.	0.321 (-0.001)	89.8 (+1.4)
which has sharp claws.	0.324 (+0.002)	89.9 (+1.5)
which has soulful eyes.	0.317 (-0.005)	89.9 (+1.5)
which is a long arm with a claw ...	0.324 (+0.002)	87.8 (-0.6)
which is a medium-sized, short-haired cat.	0.327 (+0.005)	91.4 (+3.0)
which is a boat with sails.	0.293 (-0.029)	81.5 (-6.9)
which often used by knights and soldiers.	0.315 (-0.007)	80.8 (-7.6)
which can vary in size from small to large.	0.333 (+0.011)	88.6 (+0.2)
which typically has a yellow or brownish color.	0.335 (+0.013)	89.3 (+0.9)

Target: Textures (DTD)	Alignment	Accuracy
no descriptor	0.273	44.3
a type of texture. (handcrafted; for reference)	0.287 (+0.014)	44.1 (-0.2)
which may be decorated with a pattern or logo.	0.286 (+0.013)	47.2 (+2.9)
which is a sign with the shop's name.	0.261 (-0.012)	45.3 (+1.0)
which is a backdrop.	0.280 (+0.007)	46.6 (+2.3)
which is a repeating pattern.	0.283 (+0.010)	46.3 (+2.0)
which typically has a pattern or design.	0.295 (+0.022)	45.5 (+1.2)
which is a guard tower.	0.243 (-0.030)	43.4 (-0.9)
which has loud crow.	0.253 (-0.020)	42.4 (-1.9)
which can be brightly colored or patterned.	0.283 (+0.010)	44.5 (+0.2)
which is a curved or rectangular shape.	0.281 (+0.008)	44.4 (+0.1)

Table 1. Qualitative comparison of descriptors. We select descriptors based on a source dataset using Alg. 1 and test on a target dataset. The tables are organized by the target dataset; the color of the highlight indicates the source dataset. We include randomly selected descriptors in gray for comparison. Alignment refers to the average cosine similarity between image embeddings and the corresponding text embeddings. Observe that selected descriptors tend to describe the source dataset as a whole and improve both accuracy and alignment. Also observe that a descriptor soup trained on ImageNet (blue) generalizes to other datasets, but not vice versa.

for the same number of descriptors per label (m). This is a surprising result. We reason that selecting descriptors which maximize few-shot training accuracy would achieve higher accuracy than random descriptors; this motivates descriptor soup.

3.2. Descriptor Soup

We reference Alg. 1 in the Appendix throughout this section. Let $\mathcal{D} = \{d_1, ..., d_n\}$ denote a set of n descriptors such as "which is a freshwater fish". These descriptors are obtained by combining all descriptors generated by GPT for 1,000 ImageNet classes [35], and keeping only unique entries. Descriptors are no longer connected to their original classes. We wish to select a set of m descriptors that maximizes accuracy on few-shot training data. Let's define the loss function $\ell(\mathcal{S}_{\text{train}}, \mathcal{T}_{\text{train}}(d))$ to be the 0-1 loss of the model using descriptor d over the entire training dataset $\mathcal{S}_{\text{train}}$. $\mathcal{T}_{\text{train}}(d)$ denotes the label text embeddings calculated by the text encoder by appending descriptor d to all class names. Since all parameters of the vision model remain constant, we ignore vision model parameters in the notation. We aim to find a set of m descriptors whose centroids in the text embedding space minimize the 0-1 loss:

$$\mathcal{D}^*_m = \{d^*_1, ..., d^*_m\} = \underset{d_{1:m} \in \mathcal{D}}{\arg\min}\, \ell \left(\mathcal{S}_{\text{train}}, \frac{1}{m} \sum_{i=1}^{m} \mathcal{T}_{\text{train}}(d_i) \right) \tag{1}$$

Note that $\frac{1}{m} \sum_{i=1}^{m} \mathcal{T}_{\text{train}}(d_i)$ denotes the L2-normalized centroid of text embeddings for each class. We always normalize the centroid so it can be used to calculate the cosine similarity with image embeddings; this is omitted from the math to avoid clutter.

Eq. 1 is an intractable combinatorial problem, but we can approximately solve it via a greedy approach or by solving the continuous version of the problem using gradient descent. We use a greedy approach, inspired by Wortsman et al. [58]. The algorithm can be summarized as (reference Alg. 1):

1. Calculate $\ell(\mathcal{S}_{\text{train}}, \mathcal{T}_{\text{train}}(d))$ for all $d \in \mathcal{D}$. Sort the descriptors by increasing loss / decreasing accuracy. With slight abuse of notation, denote the sorted list as $\mathcal{D} = [d_0, ..., d_n]$.
2. Initialize the "descriptor soup" $\mathcal{D}^* = \{d_0\}$ with the best descriptor.
3. For i in $1 : n$: Add d_i to $\mathcal{D}^*$ if it decreases the loss of $\mathcal{D}^*$.
4. Return the first m descriptors in $\mathcal{D}^*$.

Please find ZS results for descriptor soup in Tab. 3.

Target: ImageNet	Alignment	Uniformity	Accuracy
no descriptor	0.301	0.173	67.1
dat they ... difficulties.	0.306 (+0.005)	0.174 (+0.001)	68.9 (+1.8)
similar vary ... mention etc.	0.314 (+0.013)	0.183 (+0.010)	69.1 (+2.0)
separately aspects ... adopted.	0.315 (+0.014)	0.181 (+0.008)	69.2 (+2.1)
tue alot ... itself.	0.303 (+0.002)	0.178 (+0.005)	69.0 (+1.9)
bufing beginner ... status.	0.311 (+0.010)	0.181 (+0.008)	68.8 (+1.7)
soviet vbulletin ... inexpensive.	0.320 (+0.019)	0.195 (+0.022)	62.0 (-5.1)
ideal ips ... filename.	0.314 (+0.013)	0.196 (+0.023)	59.7 (-7.4)

Table 2. Example of a 5 member word soup trained on ImageNet (in blue) along with random chains of words (in gray) for comparison. Comparing with Tab. 1, we observe that the word soup descriptors achieve higher accuracy than descriptor soups, since word soup is more flexible from an optimization perspective. Here, we include uniformity scores, since chains of random words improve alignment at the expense of increasing uniformity. Uniformity is the average cosine similarity between image and text embeddings with different labels.

Building Intuition A natural question to ask is: descriptor soup members no longer describe individual classes, so why does Alg. 1 work? The answer has two parts (1) Alg. 1 finds

		Source	Cross-dataset (XD) Evaluation Targets											Domain Generalization Targets				
	m	INet	Caltech	Pets	Cars	Flowers	Food	Aircraft	SUN	DTD	EuroSAT	UCF	**Mean**	INet-V2	Sketch	INet-A	INet-R	**Mean**
CLIP ZS [70]	1	67.1	93.3	89.0	65.4	71.0	85.7	25.0	63.2	43.6	46.7	67.4	65.02	61.0	46.6	47.2	74.1	57.22
Ensemble [43]	80	68.4	93.5	88.8	66.0	71.1	86.0	24.8	66.0	43.9	45.0	68.0	65.31	61.9	48.5	49.2	**77.9**	59.36
GPT centroids [35]	5.8	68.2	94.1	88.4	65.8	71.5	85.7	24.7	**67.5**	44.7	46.6	67.4	65.63	61.5	48.2	48.9	75.1	58.40
GPT score mean [35]	5.8	68.6	93.7	89.0	65.1	72.1	85.7	23.9	67.4	44.0	46.4	66.8	65.42	61.8	48.1	48.6	75.2	58.42
Random descriptors	16	67.9	94.1	87.6	65.6	71.5	85.6	24.9	66.1	44.7	49.1	67.2	65.65	61.6	48.7	50.0	76.7	59.22
+ offset trick (ours)	96	68.5	93.5	89.2	65.8	72.0	85.7	25.2	66.1	44.4	53.0	68.2	66.29	61.9	48.9	**50.6**	77.5	59.76
Waffle CLIP [45]	16	68.1	93.5	88.4	65.4	72.0	85.9	25.9	66.2	44.1	46.3	68.0	65.58	61.8	48.6	49.8	76.2	59.08
+ offset trick (ours)	96	68.6	93.1	89.5	65.9	72.1	86.1	**26.3**	66.2	44.2	52.5	68.8	66.49	62.1	48.9	50.2	77.1	59.59
Descriptor soup (ours)	16.7	68.9	**94.7**	89.4	**66.2**	72.2	86.2	25.5	67.3	45.1	46.6	68.7	66.18	62.1	48.7	49.7	76.4	59.25
+ offset trick (ours)	100	69.1	93.8	**89.8**	66.0	**72.9**	**86.2**	25.4	66.8	45.0	51.6	**69.1**	66.67	62.6	**49.0**	50.5	77.2	59.82
Word soup (ours)	8	69.2	94.4	89.5	65.4	72.3	85.8	25.8	67.4	44.7	53.5	68.4	66.72	62.9	48.7	50.2	77.0	59.69
Word soup score mean (ours)	8	**69.4**	94.3	89.6	65.4	72.4	85.9	25.9	67.3	**45.2**	**55.8**	68.5	**67.03**	**63.0**	49.0	50.4	77.2	**59.90**
gain over GPT		+0.8	+0.6	+0.6	+0.3	+0.3	+0.2	+2.0	-0.1	+1.2	+9.4	+1.7	+1.6	+1.2	+0.9	+1.8	+2.0	+1.5
gain over Waffle		+1.3	+0.8	+1.2	+0.0	+0.4	+0.0	-0.0	+1.1	+1.1	+9.5	+0.5	+1.5	+1.2	+0.4	+0.6	+1.0	+0.8

Table 3. Comparison with ZS methods. All baseline methods in this table use prompts/descriptors on top of the pretrained model in a ZS manner. Note that the soup methods are not truly zero-shot because they require some training data. However, we do compare against all baselines in the few-shot setting in Table 6. We use the ViT/B-16 CLIP model trained by Open-AI. All non-deterministic numbers are an average of 3 random seeds. m indicates the number of descriptors used. "Ensemble" refers to the set of 80 handcrafted prompts created by Open-AI; GPT score mean corresponds to the classification by description method. We use centroid evaluation unless "score mean" is explicitly stated. We achieve substantial gains over GPT descriptors and waffle CLIP as indicated in the bottom two rows.

descriptors which describe the dataset as a whole, rather than individual labels; these descriptors are orthogonal to the classification problem and *increase classification accuracy by increasing alignment between corresponding image and text embeddings.* (2) Descriptor soups generalize when the target classification problem has a narrower scope than the source classification problem. Prior work (e.g. [25, 45, 70]) suggests that handcrafted dataset-specific descriptors such as "a type of aircraft" or "a type of pet" improve ZS accuracy. Dataset-level descriptors like these are easier to design than label-level descriptors, so using dataset-level descriptors is currently standard practice. We hypothesize that these descriptors improve accuracy by increasing alignment between corresponding image and text embeddings; we demonstrate this in Tab. 1. e.g. "a type of pet" improves pet classification accuracy by 0.6% and alignment by 0.01.

We further hypothesize that descriptor soup members learn to mimic the behavior of handcrafted dataset-level descriptors. We display examples of descriptor soups trained on three different datasets in Table 1 in support of this intuition. Descriptors trained on pets (in pink) mention "claws", "eyes", and "hair", which are concepts common to most pets. In a similar vein, descriptors trained on textures/DTD (in yellow) mention "pattern", "logo", and "design". Meanwhile, ImageNet is a broader dataset, so descriptors trained on ImageNet (in blue) are generally non-specific (e.g. "which could be brown or grey"). This is intuitive, since ImageNet is a dataset with diverse classes. A descriptor such as "which is a type of dog" would be detrimental to the zero-shot accuracy, since it would bias the classifier toward labels that are types of dogs. Table 1 shows that individual descriptor soup members increase both the alignment and classification accuracy, when the source and target datasets are the same. The next paragraph addresses the issue of generalizability when source and target datasets are different.

Generalizability Descriptor soups trained on ImageNet generalize to target datasets with narrower scopes, but not vice versa. This is because ImageNet concepts are a superset of narrower target datasets; e.g. ImageNet classes contain types of cars and pets. Table 1 shows that descriptors trained on ImageNet (blue) improve both the alignment and accuracy on Pets and Textures; but descriptors trained on the latter two datasets (pink and yellow) decrease the same metrics on ImageNet. To further support the generalizability of descriptor soups, we show a positive correlation between ImageNet accuracy and *average* target dataset accuracy in Fig. 3 (right). Finally, we train a descriptor soup on *test data* to maximize average accuracy of 10 datasets; we call this the "descriptor soup upper bound" in the middle of Tab. 6. The upper bound only achieves marginal improvement over the descriptor soup trained on ImageNet (three rows above the upper bound in Tab. 6). This suggests that greedily maximizing the descriptor soup accuracy on ImageNet training data is a good approximation of maximizing the target accuracy; i.e. the generalization gap is small.

3.3. Word Soup

Descriptor soup achieves impressive state-of-the-art performance, but it is still *reliant on an LLM* at training time to generate a list of candidate descriptors and is *limited* to this fixed descriptor list. In order to remove the reliance on LLMs

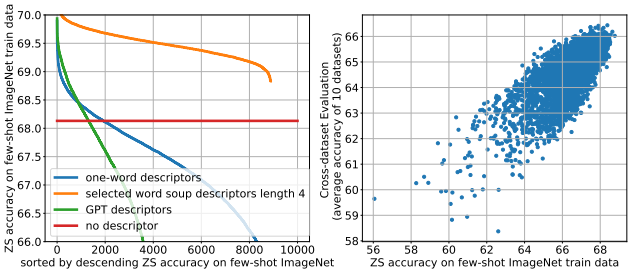

Figure 3. (Left) Plot of ImageNet accuracy when the same descriptor is appended to every class label. Observe that there are more than 1,000 GPT descriptors and single-word descriptors that are better than standard ZS (in red). When we further consider word chains of length 4, the number of accurate descriptors increases dramatically (orange). (Right) Scatter plot of average target accuracy vs. ImageNet accuracy of GPT descriptors. We observe a positive correlation, so descriptors trained on ImageNet are likely to generalize to other datasets.

and make the optimization process more flexible, we propose to generate descriptors in a greedy fashion using individual words selected from a dictionary. We use the list of 10,000 most commonly-used words on the web[1] as the candidate pool of words.

Given a list of n words $\mathcal{W} = \{w_1, ..., w_n\}$ (we abuse some notations slightly, since the word soup is a separate method). Descriptors are allowed to be any sequence of words, as long as the length does not exceed p. Concretely,

$$\mathcal{D}_m^* = \{d_1^*, ..., d_m^*\} = \arg\min_{d_{1:m} \in D'} \ell \left(\mathcal{S}_{\text{train}}, \frac{1}{m} \sum_{i=1}^{m} \mathcal{T}_{\text{train}}(d_i) \right)$$
$$D' := \{\text{all } q \text{ permutations of } \mathcal{W}, \forall q \leq p\} \tag{2}$$

The word soup problem described by Eq. 2 is again intractable, so we propose an approximate greedy solution using the following steps (see Alg. 2 in the Appendix):

1. **Initialization:** Sort $\mathcal{W}$ by decreasing ZS accuracy to filter out unsuitable words (see Fig. 3 left). For this step, we only consider single word descriptors (e.g. "a photo of a cat, the."). Select the top-k_0 and top-k_1 words, denoted as $\mathcal{W}_{\text{top}k_0}$ and $\mathcal{W}_{\text{top}k_1}$, resp. $k_0 < k_1$.
2. Randomly select a word w from $\mathcal{W}_{\text{top}k_0}$ and initialize the descriptor $d = w$.
3. Shuffle $\mathcal{W}_{\text{top}k_1}$. Then, for $w' \in \mathcal{W}_{\text{top}k_1}$, append w' to d, only if it increases the accuracy of d.
4. return d.

We obtain a total of m independent (in a loose sense) descriptors by repeating steps 2-4. In these steps, we randomly select from $\mathcal{W}_{\text{top}k_0}$ and shuffle $\mathcal{W}_{\text{top}k_1}$ to encourage diversity among the m selected descriptors. Instead of truncating all descriptors to a pre-determined length p, we introduce a patience parameter in Alg. 2, which implicitly controls the average descriptor length. We now motivate word soup.

[1] github.com/first20hours/google-10000-english

	m	Source INet	**XD Mean** (10 datasets)	**DG Mean** (4 datasets)
CLIP ZS	1	67.1	65.02	57.22
Vanilla CoOp	1	70.0	66.52	59.25
+ word soup	8	69.6	66.59	59.26
CoOp ensemble	8	69.8	66.68	59.18
CoOp regularized towards initialization	1	70.2	66.97	59.94
+ word soup	8	69.9	66.69	60.05
CoOp with label smoothing	1	70.1	66.37	60.09
+ word soup	8	69.9	66.13	60.16
CoOp + word soup ($\lambda = 0$)	8	69.8	66.21	59.15
+ our diversity loss ($\lambda = 0.25$)	8	**70.2**	**67.23**	**60.20**

Table 4. Ablation results to support the diversity loss. "Vanilla CoOp + word soup" refers to appending the word soup descriptors directly to soft CoOp prompts. "CoOp ensemble" refers to ensembling m randomly-initialized soft descriptors trained with CoOp. Observe that the model trained with our diversity loss ($\lambda = 0.25$) achieves a 1% increase in accuracy on average. This increase in accuracy cannot be achieved with label smoothing or regularization towards the initialization as in MIRO [4] and ProGrad [71]. Detailed results see Tab. 9 in the Appendix.

Motivation from descriptor soup The descriptor soup method has some intuitive properties covered in the previous sub-section, but is limited by the small number of good descriptors. Fig. 3 left shows that only about 1,200 descriptors (green line) in $\mathcal{D}$ are better than no descriptor (vanilla ZS; red line). The descriptor soup is limited to various combinations of these 1,200 "good" descriptors. On the contrary, when we expand the hypothesis space to be D', any permutation of a set of words, there are many more good descriptors to choose from, as indicated by the orange line in Fig. 3 left. In other words, word soup improves classification accuracy by increasing the size of the hypothesis class. Tab. 2 supports this assertion by showing that individual word soup descriptors achieve higher accuracies on ImageNet than descriptor soup members.

3.4. Diversity loss

Word soup already achieves competitive performance on most benchmarks. A reasonable next step would be to finetune using the word soup descriptors as an initialization. A variety of methods exist for few-shot finetuning of CLIP, e.g. CoOp, Clipood, and MaPLe. However, in many cases we actually see a slight decline in target accuracy after finetuning in Tab. 4 ($\lambda = 0$). This is because finetuning all descriptors on the same few-shot data forces text-prototypes to converge to the same locations in the embedding space, eliminating the initial diversity. Given *fixed* word soup descriptors $\mathcal{D}^* = \{d_1^*, ..., d_m^*\}$, our training loss is:

$$\ell_{\text{train}} = \mathbb{E}_{d_i^* \sim \mathcal{D}^*} \left[\text{CE}(\hat{y}_{d_i^*}, (1-\lambda) y_{\text{truth}} + \lambda \hat{y}_{d_i^*, 0}) \right] \tag{3}$$

	m	Cross-dataset Evaluation Target Mean B/32†	B/16†	L/14‡	CoCa L/14‡	g/14‡
ZS	1	61.32	65.02	73.11	74.82	77.58
GPT score mean	5.8	61.22	65.42	73.08	75.48	77.14
Waffle CLIP	16	62.13	65.58	73.25	75.37	77.72
Desc. soup + offsets	100	**62.79**	66.67	73.19	75.95	78.04
Word soup (ours)	8	62.24	**67.03**	**73.56**	**76.08**	**78.09**
	m	Domain Generalization Evaluation Target Mean B/32†	B/16†	L/14‡	CoCa L/14‡	g/14‡
ZS	1	47.68	57.22	64.88	67.94	71.37
GPT score mean	5.8	47.95	58.42	64.96	67.67	71.26
Waffle CLIP	16	49.07	59.08	64.47	67.85	70.99
Desc. soup + offsets	100	**50.05**	59.82	**65.81**	68.32	**72.21**
Word soup (ours)	8	50.00	**59.90**	65.73	**68.73**	72.05

Table 5. Comparison with ZS baselines at different model scales. † indicates a model trained by Open-AI [43]; ‡ indicates a model trained by Open-CLIP [19]. Detailed results see Tab. 12 in the Appendix.

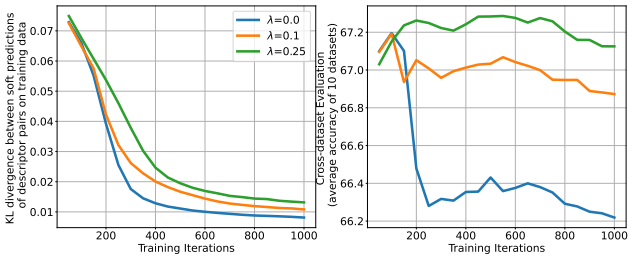

Figure 4. Varying λ in the diversity loss. $\lambda = 0$ corresponds to the standard CE loss. The left plot displays the average KL divergence between predicted class probabilities of word soup descriptors over the course of training. The right plot displays the cross-dataset accuracy for the same training runs. We observe that a larger λ leads to higher diversity among descriptors; this results in a higher test accuracy.

where CE denotes the cross entropy loss, $\hat{y}_{d_i^*} \in \Delta_c$ (c is the number of classes) denotes the soft prediction of the model with descriptor d_i^*; y_{truth} denotes the one-hot encoding of the true label; and $\hat{y}_{d_i^*,0} \in \Delta_c$ denotes the soft prediction of the initial model with descriptor d_i^*. $\lambda \in [0, 1]$ is a hyperparameter controlling the amount of regularization. $\hat{y}_{d_i^*}$ is the quantity being optimized. $\hat{y}_{d_i^*,0}$ is the output of a softmax with temperature τ_0 (the teacher temperature). As in classical knowledge distillation, it is often useful to set the teacher temperature to be different than the training temperature. Training the expectation directly in Eq. 3 requires storing mc forward and backward passes of the text encoder in memory, which is not scalable. In practice, we use one descriptor per mini-batch and rotate among the m descriptors in a round-robin fashion, but we train for the *same number of iterations* as finetuning with one descriptor.

Our training loss biases the model prediction toward the initial prediction of the model using each description, thereby maintaining the diversity of predictions present at initialization. Fig. 4 verifies this interpretation by showing that training with $\lambda = 0.25$ results in a higher average KL divergence between descriptor predictions $\hat{y}_{d_i^*}$ and a higher average target accuracy than training with lower λs. Additionally, Tab. 4 displays results for a naive CoOp ensemble and CoOp trained with regularization towards the initialization. These results show that our diversity loss results cannot be obtained by simply ensembling or regularizing predictions towards the initialization as in [4, 71]. The training does not take longer than standard cross entropy training, since only one model is trained for all descriptors. Descriptor tokens are fixed.

	m	Source INet	**XD Mean** (10 datasets)	**DG Mean** (4 datasets)
CLIP ZS [43]	1	67.1	65.02	57.22
CoOp [70]†		71.5	63.88	59.3
Co-CoOp [69]†		71.0	65.74	59.9
MaPLe [25]†		70.7	66.30	60.3
CLIPood [51]†		71.6		60.5
Cross Entropy (CE)	1	**72.3**	66.80	60.39
+ GPT score mean [35]	5.8	71.7	66.86	59.92
+ Random descriptors	32	71.6	66.89	60.69
+ Waffle CLIP [45]	32	71.6	66.58	60.65
+ Descriptor soup (ours)	16.7	72.1	67.10	60.70
+ offset trick (ours)	100	72.1	**67.51**	61.01
+ Word soup centroids (ours)	8	71.8	67.16	61.22
+ Word soup score mean (ours)	8	71.7	67.43	**61.32**
+ Descriptor soup upper bound	11	71.7	67.62	61.01
ProGrad [71]	1	69.8	66.48	58.96
KgCoOp [22]	1	69.2	66.16	58.64
ProDA [31]	32	70.0	66.23	58.83
Vanilla CoOp [70]	1	70.0	66.52	59.25
+ Word soup score mean (ours)	8	**70.2**	**67.30**	**60.25**
Vanilla MaPLe [25]	1	70.7	66.44	59.32
+ Word soup score mean (ours)	8	**70.8**	**66.65**	**60.20**
Vanilla CLIPood [51]	1	**72.9**	66.50	60.47
+ Word soup score mean (ours)	8	72.0	**67.42**	**61.23**

Table 6. Comparison with few-shot methods and few-shot methods stacked with ZS methods. † indicates author-reported numbers on the same datasets with the same train-test splits. Other numbers are our reproductions. All methods except the upper bound were trained on 3 random 16-shot splits of ImageNet. m indicates number of descriptors used. Either our descriptor soup with the offset trick or our word soup achieves the best accuracy on average. We use the ViT/B-16 CLIP model. Detailed results see Tab. 10 in the Appendix.

4. Results

We present the main few-shot results in Tab. 6. The goal this section is to demonstrate the following in the OOD setting:

1. **Complementary to existing few-shot methods:** Stacking either descriptor soup or word soup on top of traditional finetuning baselines (Cross Entropy, MaPLe, Clipood, or CoOp) improves target accuracy, exceeding current published state-of-art. (Tab. 6)
2. **Parameter Efficiency:** Our method is more parameter efficient than CoOp due to the discrete nature of word soup tokens. We additionally compare to other PEFT

methods: VPT [20], bitfi t[64], CLIP-adapter [11], SSF [29], LoRA [17], and adapter [16]. (Fig. 2)

3. **Descriptor Efficiency:** We outperform prior state-of-the-art ZS methods with only 1 or 2 descriptors. Therefore, unlike some prior methods, our method is *not* primarily driven by ensembling. (Fig. 5)

Datasets We train on random 16-shot splits of ImageNet-1K [46] and test on 14 unseen target datasets: Caltech-101 [28], Oxford-Pets [40], Stanford-Cars [27], Flowers-102 [36], Food-101 [2], FGVC-Aircraft [34], SUN-397 [60], Describable-Textures (DTD) [7], EuroSAT [13], UCF-101 (an action recognition dataset) [52], ImageNet-V2 [44], ImageNet-Sketch [56], ImageNet-A (natural adversarial examples) [15], and ImageNet-R [14]. The last four datasets are domain-shifted versions of ImageNet containing images from the ImageNet-1K label space.

Experimental Setting All baselines and methods are trained on 16-shot ImageNet-1K data and tested on the indicated target datasets. *Hyperparameters:* We tune parameters on a withheld validation set. Word soup (Alg. 2) has three parameters: k_0, k_1 and patience. The diversity loss has two parameters: λ and τ_0. These 5 parameters are constant across all experiments. We tune the learning rate separately for each baseline, but keep all other training parameters consistent across methods. We report temperature, batch size, optimizer, EMA setting, token length, initialization and other training details in Appendix A. We discuss the difference between centroid and score mean evaluation in Appendix D.

Discussion In Tab. 6, we first observe that stacking our word soup method on top of CE, CoOp, MaPLe, or CLIPood achieves approximately 0.8-1.0% increase in average target accuracy for both XD and DG benchmarks. Due to the space limitation, we only compare word soup with other ZS methods when combined with CE, since CE achieves the highest XD accuracy out of the 4 finetuning methods. m indicates the number of descriptors for each label, on average. The greedy descriptor soup can be augmented using our token offset trick, which uses 6 augmented copies of each descriptor. The token offset trick improves accuracy by 0.4% and 0.3% on XD and DG, resp. but at a significant computational cost. The greedy word soup matches the performance of the augmented descriptor soup without the additional computational cost. Overall, the best OOD accuracy is achieved by either the descriptor soup with token offsets or word soup.

Ablation Study An ablation study on our soup methods with varying m is presented in Fig. 5. On both benchmarks, our word soup performs best for all m. We note that the word soup with $m = 2$ already outperforms all ZS baselines for all values of m up to 64. This result indicates that, unlike state-of-the-art ZS methods, ensembling is not the main ingredient

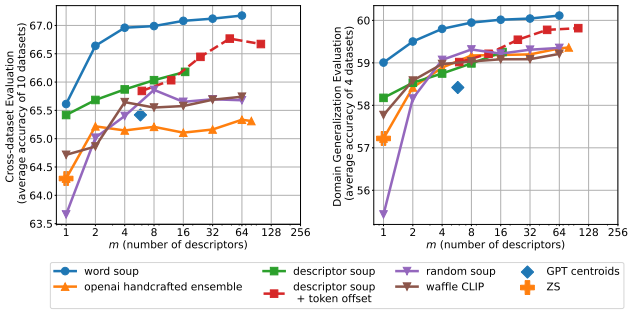

Figure 5. Comparison of our soups with ZS baselines for varying m on XD and DG evaluations. This experiment uses the same settings as Tab. 3. Our word soup achieves the best accuracies for all m. This shows that word soup is more descriptor efficient than baseline ZS methods.

of our method. Additional ablation studies are presented in Appendix E.

Parameter Efficiency and Computational Efficiency A discussion regarding efficiency of our methods is deferred to Appendix E.

5. Conclusion

In this paper, we proposed descriptor and word soups to tackle the cross-dataset and domain generalization problems. Descriptor soup greedily selects a set of descriptors by maximizing training accuracy on a source dataset. Word soup builds a chain of words using a similar greedy procedure. These greedy soup methods achieve higher target classification accuracy than prior descriptor-based methods by explicitly maximizing training accuracy. We further proposed a loss function to preserve word soup diversity throughout finetuning. When using word soup for initialization and finetuning with the diversity loss, we can significantly improve the accuracy of existing few-shot OOD finetuning methods. Compared to all baselines, word soup achieves the best trade-off between parameter efficiency and target accuracy.

Acknowledgements

DISTRIBUTION STATEMENT A. Approved for public release. Distribution is unlimited.

This material is based upon work supported by the Under Secretary of Defense for Research and Engineering under Air Force Contract No. FA8702-15-D-0001. Any opinions, findings, conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Under Secretary of Defense for Research and Engineering.

References

[1] James Urquhart Allingham, Jie Ren, Michael W Dusenberry, Xiuye Gu, Yin Cui, Dustin Tran, Jeremiah Zhe Liu, and Balaji Lakshminarayanan. A simple zero-shot prompt weighting technique to improve prompt ensembling in text-image models. In *International Conference on Machine Learning*, pages 547–568. PMLR, 2023. 3

[2] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101 – mining discriminative components with random forests. In *European Conference on Computer Vision*, 2014. 8

[3] Adrian Bulat and Georgios Tzimiropoulos. Lasp: Text-to-text optimization for language-aware soft prompting of vision & language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23232–23241, 2023. 2

[4] Junbum Cha, Kyungjae Lee, Sungrae Park, and Sanghyuk Chun. Domain generalization by mutual-information regularization with pre-trained models. In *European Conference on Computer Vision*, pages 440–457. Springer, 2022. 6, 7, 5

[5] Guangyi Chen, Weiran Yao, Xiangchen Song, Xinyue Li, Yongming Rao, and Kun Zhang. Prompt learning with optimal transport for vision-language models. *arXiv preprint arXiv:2210.01253*, 2022. 2

[6] Junhyeong Cho, Gilhyun Nam, Sungyeon Kim, Hunmin Yang, and Suha Kwak. Promptstyler: Prompt-driven style generation for source-free domain generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15702–15712, 2023. 2

[7] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. *CoRR*, abs/1311.3618, 2013. 8

[8] Rajshekhar Das, Yonatan Dukler, Avinash Ravichandran, and Ashwin Swaminathan. Learning expressive prompting with residuals for vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3366–3377, 2023. 2

[9] Mohammad Mahdi Derakhshani, Enrique Sanchez, Adrian Bulat, Victor G Turrisi da Costa, Cees GM Snoek, Georgios Tzimiropoulos, and Brais Martinez. Bayesian prompt learning for image-language model generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15237–15246, 2023. 2

[10] Chun-Mei Feng, Kai Yu, Yong Liu, Salman Khan, and Wangmeng Zuo. Diverse data augmentation with diffusions for effective test-time prompt tuning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2704–2714, 2023. 3

[11] Peng Gao, Shijie Geng, Renrui Zhang, Teli Ma, Rongyao Fang, Yongfeng Zhang, Hongsheng Li, and Yu Qiao. Clip-adapter: Better vision-language models with feature adapters. *International Journal of Computer Vision*, pages 1–15, 2023. 3, 8, 2

[12] Xuehai He, Diji Yang, Weixi Feng, Tsu-Jui Fu, Arjun Akula, Varun Jampani, Pradyumna Narayana, Sugato Basu, William Yang Wang, and Xin Eric Wang. Cpl: Counterfactual prompt learning for vision and language models. *arXiv preprint arXiv:2210.10362*, 2022. 2

[13] 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. 8

[14] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8340–8349, 2021. 8

[15] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15262–15271, 2021. 8

[16] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, pages 2790–2799. PMLR, 2019. 3, 8, 2

[17] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021. 3, 8, 2

[18] Zi-Yuan Hu, Yanyang Li, Michael R Lyu, and Liwei Wang. Vl-pet: Vision-and-language parameter-efficient tuning via granularity control. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3010–3020, 2023. 3

[19] Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. Openclip, 2021. If you use this software, please cite it as below. 7

[20] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *European Conference on Computer Vision*, pages 709–727. Springer, 2022. 1, 3, 8, 2

[21] Shibo Jie and Zhi-Hong Deng. Convolutional bypasses are better vision transformer adapters. *arXiv preprint arXiv:2207.07039*, 2022. 3

[22] Baoshuo Kan, Teng Wang, Wenpeng Lu, Xiantong Zhen, Weili Guan, and Feng Zheng. Knowledge-aware prompt tuning for generalizable vision-language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15670–15680, 2023. 2, 3, 7, 5

[23] Guoliang Kang, Lu Jiang, Yi Yang, and Alexander G Hauptmann. Contrastive adaptation network for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4893–4902, 2019. 1

[24] Prannay Kaul, Weidi Xie, and Andrew Zisserman. Multi-modal classifiers for open-vocabulary object detection. *arXiv preprint arXiv:2306.05493*, 2023. 1, 3

[25] Muhammad Uzair Khattak, Hanoona Rasheed, Muhammad Maaz, Salman Khan, and Fahad Shahbaz Khan. Maple: Multi-modal prompt learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19113–19122, 2023. 1, 2, 5, 7

[26] 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 the IEEE/CVF International Conference on Computer Vision*, pages 15190–15200, 2023. 2

[27] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 554–561, 2013. 8

[28] Fei-Fei Li, Marco Andreeto, Marc'Aurelio Ranzato, and Pietro Perona. Caltech 101, 2022. 8

[29] Dongze Lian, Daquan Zhou, Jiashi Feng, and Xinchao Wang. Scaling & shifting your features: A new baseline for efficient model tuning. *Advances in Neural Information Processing Systems*, 35:109–123, 2022. 3, 8, 2

[30] Zhiqiu Lin, Samuel Yu, Zhiyi Kuang, Deepak Pathak, and Deva Ramanan. Multimodality helps unimodality: Cross-modal few-shot learning with multimodal models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19325–19337, 2023. 2

[31] Yuning Lu, Jianzhuang Liu, Yonggang Zhang, Yajing Liu, and Xinmei Tian. Prompt distribution learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5206–5215, 2022. 2, 3, 7, 5

[32] Gen Luo, Minglang Huang, Yiyi Zhou, Xiaoshuai Sun, Guannan Jiang, Zhiyu Wang, and Rongrong Ji. Towards efficient visual adaption via structural re-parameterization. *arXiv preprint arXiv:2302.08106*, 2023. 3

[33] Xiaosong Ma, Jie Zhang, Song Guo, and Wenchao Xu. Swap-prompt: Test-time prompt adaptation for vision-language models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. 3

[34] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew B. Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *CoRR*, abs/1306.5151, 2013. 8

[35] Sachit Menon and Carl Vondrick. Visual classification via description from large language models. *arXiv preprint arXiv:2210.07183*, 2022. 1, 2, 3, 4, 5, 7

[36] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian Conference on Computer Vision, Graphics and Image Processing*, pages 722–729, 2008. 8

[37] Laura Niss, Kevin Vogt-Lowell, and Theodoros Tsiligkaridis. Quantified task misalignment to inform PEFT: An exploration of domain generalization and catastrophic forgetting in CLIP. In *ICLR 2024 Workshop on Mathematical and Empirical Understanding of Foundation Models*, 2024. 1

[38] Zachary Novack, Julian McAuley, Zachary Chase Lipton, and Saurabh Garg. Chils: Zero-shot image classification with hierarchical label sets. In *International Conference on Machine Learning*, pages 26342–26362. PMLR, 2023. 3

[39] Omiros Pantazis, Gabriel Brostow, Kate Jones, and Oisin Mac Aodha. Svl-adapter: Self-supervised adapter for vision-language pretrained models. *arXiv preprint arXiv:2210.03794*, 2022. 3

[40] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3498–3505. IEEE, 2012. 8

[41] Fang Peng, Xiaoshan Yang, Linhui Xiao, Yaowei Wang, and Changsheng Xu. Sgva-clip: Semantic-guided visual adapting of vision-language models for few-shot image classification. *IEEE Transactions on Multimedia*, 2023. 2

[42] Sarah Pratt, Ian Covert, Rosanne Liu, and Ali Farhadi. What does a platypus look like? generating customized prompts for zero-shot image classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15691–15701, 2023. 1, 3

[43] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021. 2, 5, 7

[44] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? *CoRR*, abs/1902.10811, 2019. 8

[45] Karsten Roth, Jae Myung Kim, A Koepke, Oriol Vinyals, Cordelia Schmid, and Zeynep Akata. Waffling around for performance: Visual classification with random words and broad concepts. *arXiv preprint arXiv:2306.07282*, 2023. 1, 3, 5, 7

[46] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115 (3):211–252, 2015. 8

[47] Kuniaki Saito, Donghyun Kim, Stan Sclaroff, Trevor Darrell, and Kate Saenko. Semi-supervised domain adaptation via minimax entropy. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8050–8058, 2019. 1

[48] Jameel Hassan Abdul Samadh, Hanan Gani, Noor Hazim Hussein, Muhammad Uzair Khattak, Muzammal Naseer, Fahad Khan, and Salman Khan. Align your prompts: Test-time prompting with distribution alignment for zero-shot generalization. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. 3

[49] Cheng Shi and Sibei Yang. Logoprompt: Synthetic text images can be good visual prompts for vision-language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2932–2941, 2023. 2

[50] Manli Shu, Weili Nie, De-An Huang, Zhiding Yu, Tom Goldstein, Anima Anandkumar, and Chaowei Xiao. Test-time prompt tuning for zero-shot generalization in vision-language models. *Advances in Neural Information Processing Systems*, 35:14274–14289, 2022. 3

[51] Yang Shu, Xingzhuo Guo, Jialong Wu, Ximei Wang, Jianmin

Wang, and Mingsheng Long. Clipood: Generalizing clip to out-of-distributions, 2023. 2, 7, 5

[52] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. UCF101: A dataset of 101 human actions classes from videos in the wild. *CoRR*, abs/1212.0402, 2012. 8

[53] Yi-Lin Sung, Jaemin Cho, and Mohit Bansal. Lst: Ladder side-tuning for parameter and memory efficient transfer learning. *Advances in Neural Information Processing Systems*, 35: 12991–13005, 2022. 3

[54] Yi-Lin Sung, Jaemin Cho, and Mohit Bansal. Vl-adapter: Parameter-efficient transfer learning for vision-and-language tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5227–5237, 2022. 3

[55] Kevin Vogt-Lowell, Noah Lee, Theodoros Tsiligkaridis, and Marc Vaillant. Robust fine-tuning of vision-language models for domain generalization. In *IEEE High Performance Extreme Computing Conference (HPEC)*, 2023. 1

[56] Haohan Wang, Songwei Ge, Eric P. Xing, and Zachary C. Lipton. Learning robust global representations by penalizing local predictive power. *CoRR*, abs/1905.13549, 2019. 8

[57] Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. Learning to prompt for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 139–149, 2022. 1

[58] Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, et al. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International Conference on Machine Learning*, pages 23965–23998. PMLR, 2022. 2, 3, 4

[59] Mitchell Wortsman, Gabriel Ilharco, Jong Wook Kim, Mike Li, Simon Kornblith, Rebecca Roelofs, Raphael Gontijo Lopes, Hannaneh Hajishirzi, Ali Farhadi, Hongseok Namkoong, et al. Robust fine-tuning of zero-shot models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7959–7971, 2022. 3, 2

[60] Jianxiong Xiao, James Hays, Krista A. Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3485–3492, 2010. 8

[61] Yinghui Xing, Qirui Wu, De Cheng, Shizhou Zhang, Guoqiang Liang, and Yanning Zhang. Class-aware visual prompt tuning for vision-language pre-trained model. *arXiv preprint arXiv:2208.08340*, 2022. 2

[62] Hantao Yao, Rui Zhang, and Changsheng Xu. Visual-language prompt tuning with knowledge-guided context optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6757–6767, 2023. 2

[63] Tao Yu, Zhihe Lu, Xin Jin, Zhibo Chen, and Xinchao Wang. Task residual for tuning vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10899–10909, 2023. 3

[64] Elad Ben Zaken, Shauli Ravfogel, and Yoav Goldberg. Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. *arXiv preprint arXiv:2106.10199*, 2021. 3, 8, 2

[65] Yuhang Zang, Wei Li, Kaiyang Zhou, Chen Huang, and Chen Change Loy. Unified vision and language prompt learning. *arXiv preprint arXiv:2210.07225*, 2022. 2

[66] Renrui Zhang, Rongyao Fang, Wei Zhang, Peng Gao, Kunchang Li, Jifeng Dai, Yu Qiao, and Hongsheng Li. Tip-adapter: Training-free clip-adapter for better vision-language modeling. *arXiv preprint arXiv:2111.03930*, 2021. 3

[67] Yuchen Zhang, Tianle Liu, Mingsheng Long, and Michael Jordan. Bridging theory and algorithm for domain adaptation. In *International conference on machine learning*, pages 7404–7413. PMLR, 2019. 1

[68] Kaiyang Zhou, Ziwei Liu, Yu Qiao, Tao Xiang, and Chen Change Loy. Domain generalization: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 1

[69] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16816–16825, 2022. 2, 7, 5

[70] 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, 2, 3, 5, 7

[71] Beier Zhu, Yulei Niu, Yucheng Han, Yue Wu, and Hanwang Zhang. Prompt-aligned gradient for prompt tuning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15659–15669, 2023. 2, 3, 6, 7, 5