Label Propagation for Zero-shot Classification with Vision-Language Models

Vladan Stojnić[1] Yannis Kalantidis[2] Giorgos Tolias[1]

[1] VRG, FEE, Czech Technical University in Prague [2] NAVER LABS Europe

Abstract

Vision-Language Models (VLMs) have demonstrated impressive performance on zero-shot classification, i.e. *classification when provided merely with a list of class names. In this paper, we tackle the case of zero-shot classification in the presence of unlabeled data. We leverage the graph structure of the unlabeled data and introduce* ***ZLaP****, a method based on label propagation (LP) that utilizes geodesic distances for classification. We tailor LP to graphs containing both text and image features and further propose an efficient method for performing inductive inference based on a dual solution and a sparsification step. We perform extensive experiments to evaluate the effectiveness of our method on 14 common datasets and show that ZLaP outperforms the latest related works. Code:* https://github.com/vladan-stojnic/ZLaP

1. Introduction

Vision-Language Models (VLMs) have demonstrated impressive performance on a variety of computer vision tasks. They are usually trained on large datasets of image-text pairs and contain visual and textual encoders that map to a common feature space. Visual encoders from such models have been shown to produce strong visual representations for perception tasks [31]. Given labeled data from a downstream dataset, one can fine-tune the model or learn classifiers and achieve really high classification accuracy.

Besides using the visual encoder in isolation, the joint text and visual encoder feature space of VLMs enables us to define text-based "classifiers", *e.g.* using the class names as textual prompts. This means that we only need a list of class names to perform *zero-shot* classification for a target dataset, *i.e.* without access to any labeled images. Although utilizing priors or devising better textual prompts can improve zero-shot performance [8, 24, 29, 46], here, we are interested in the case where we further have access to *unlabeled* data. Our goal is to find the best way of utilizing such data for zero-shot classification.

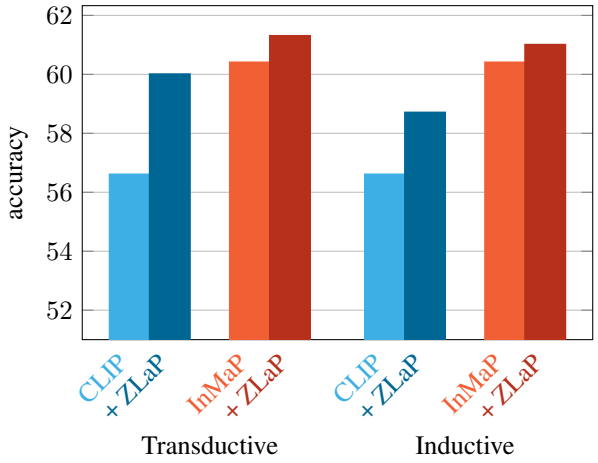


Figure 1. **Zero-shot classification performance over 14 datasets** using the proposed ZLaP classifier over CLIP [31], as well as over the (concurrent) InMaP [30] approach. Our method offers performance gains for both transductive (left) and inductive (right) inference. Average accuracy over 14 common datasets is reported.

In this paper, we leverage the inherent structure of the unlabeled data represented by a proximity graph and apply *label propagation* (LP) between the text-based classifiers and unlabeled images to derive geodesic distances we then use for classification. We tailor LP to VLMs and graphs containing both text and image features, and show that without proper handling of the bimodality, vanilla application of LP fails dramatically. We introduce ZLaP, a novel classification method based on label propagation that can perform both transductive and inductive inference. We perform the former with the standard (primal) solution of LP for classification and devise a more efficient dual solution for the latter. Our method is not only highly effective but also efficient, making LP a more attractive inductive classifier in terms of complexity.

We implement our methods using publicly available VLMs as feature encoders, primarily the ResNet and ViT CLIP [31] models, and perform extensive experiments to evaluate the effectiveness of our method on 14 common

datasets. We show that we are able to achieve top performance on two zero-shot inference setups, *i.e*. inductive and transductive inference. Figure 1 summarizes our gains over 14 datasets on both setups when applying our LP-powered classifiers on top of CLIP, as well as after incorporating the class proxies from the recent InMaP [30] zero-shot approach.

It is worth highlighting that ZLaP is a non-parametric method, *i.e*. it does not involve a learning step. In fact, our approach does not even require access to the VLM model weights and can therefore be used to improve the zero-shot performance of a black-box model even, *e.g*. provided only via an API. In summary, our contributions are as follows.

- We tailor label propagation to VLMs and zero-shot classification over bi-modal graphs, proposing per modality neighbor search and balancing of contributions.
- We propose an efficient way for performing inductive inference with label propagation via a dual solution and through sparsification. This not only improves the test-time efficiency of our method but also performance.
- We complement our method with the class proxies presented in concurrent work [30] and achieve state of the art results for zero-shot on 14 common datasets.

2. Related work

In this section, we discuss related works that improve the already impressive zero-shot classification performance of vision-language models [4, 16, 31] even further. This is achieved by devising better distance metrics, utilizing external knowledge to learn more expressive textual prompts, or by leveraging synthetic and unlabeled data.

Improved distance metrics. Zero-shot classification can be improved by devising a better distance metric between image and text representations [8, 46]. CALIP [8] uses a parameter-free attention mechanism and a local patch representation, *i.e*. instead of global representations, to improve the estimation of class-to-image similarity. CLIP-DN [46] improves the test-time similarity estimation by alignment with the similarity used during contrastive pre-training of VLMs. To achieve this, the method assumes access to unlabeled data from the target distribution. TPT [35] optimizes textual prompts via a consistency objective across test image augmentations. Our method can be considered a part of this line of work, as label propagation is a similarity measure in a geodesic space instead of the Euclidean space.

Improved textual prompts using language models. CLIP [31] uses hand-crafted prompts that are specialized for each domain. Instead of using hand-crafted prompts, generating them with large language models (LLMs) is shown to be promising [24, 29]. VisDesc [24] and CuPL [29] query LLMs to generate diverse descriptions of all classes, while WaffleCLIP [32] operates on top of VisDesc to systematically analyze which parts of the generated prompts are the most important. Instead of generating class descriptions, CHiLS [26] targets diversifying the set of classes, by generating sub-classes per class, either through an existing class hierarchy or by querying an LLM. It then performs zero-shot classification using sub-classes and linking them to the parent class. We show that methods improving the textual prompts are complementary to our approach.

Synthetic data. Recent methods [9, 38, 43] demonstrate that the use of synthetic data is beneficial for zero-shot classification. CLIP+SYN [9] uses a stable-diffusion-based model to generate synthetic images using class names and uses them to train a linear classifier, initialized by the VLM class representations. SuS-X [38] considers a similar approach, but relies on a non-parametric classifier. CaFO [43] follows the same path, but additionally includes text prompts generated by LLMs.

External datasets. Besides the use of synthetic data, SuS-X proposes a variant that operates on an extensive unlabeled image dataset (LAION-5B [34]). This dataset encompasses a distribution that is a super set of the target one. The method generates pseudo-labels using the zero-shot approach which are then incorporated within the non-parametric classifier. NeuralPriming [40] additionally assumes that images have captions, which are used to improve the pseudo-labeling.

Unlabeled images from the target distribution. Another line of research [11, 12, 27, 30] propose operating on unlabeled datasets from the target distribution. The main ingredient of all these methods is the prediction of pseudo-labels for unlabeled examples, that are later used for further processing. UPL [12] optimizes learnable text prompts based on the pseudo-labels. SVL-Adapter [27] first trains a self-supervised model on unlabeled data, and then an adapter module to align its outputs to the pseudo-labels. ReCLIP [11] performs transductive label propagation to obtain the pseudo-labels and uses them to fine-tune the VLM visual and textual encoders. In contrast to that, we do not fine-tune the model, which we may not have access to, and efficiently use label propagation for inductive inference too. InMaP [30] is a concurrent work that uses pseudo-labels to update the class representations such that they are now closer to image representations. We show in the experiments that this approach is complementary to ours. In contrast to all those methods, we do not explicitly require pseudo-label prediction, but rather capture interactions between all unlabeled examples through a proximity graph and label propagation.

3. Method

We first define the task of zero-shot classification with access to unlabeled examples, present label propagation with our contributions and then present the proposed approach for zero-shot classification using unlabeled examples.

3.1. Problem formulation

Vision-language models consist of an image encoder $f : \mathcal{I} \to \mathbb{R}^d$ and a text encoder $g : \mathcal{T} \to \mathbb{R}^d$, where $\mathcal{I}$ and $\mathcal{T}$ represent the space of images and text, respectively. We consider the outputs of these encoders to be ℓ_2-normalized.

Let $\mathcal{C}$ denote a set of known classes with associated class names $\{l_1, ..., l_C\}$ and $\mathcal{P} = \{p_1, ..., p_P\}$ a set of prompt templates. Each prompt is combined with a class name to produce a textual description of the class, *i.e.* $p_i(l_c)$ for the i-th template used for class c. Class representations $\mathbf{w}_c = {}^1\!/_P \sum_{i=1}^{P} g(p_i(l_c))$ are obtained using the VLM. Then, for a test image u, we extract representation $\mathbf{u} = f(u)$, and perform zero-shot classification by $\operatorname{argmax}_c \mathbf{u}^T \mathbf{w}_c$.

We further assume access to a set $\mathcal{U}$ of M unlabeled images. Let $\{\mathbf{u}_1, \dots, \mathbf{u}_M\}$ denote the representations of the unlabeled images.

In this work, we assume no direct access to the VML model weights, *i.e.* VLM training is neither possible nor desired. This allows to consider the underlying VLM as a black box, possibly only available through an API that generates features. We consider two inference setups:

Inductive inference: We consider an *inductive* setup, where we need to construct a classifier that can operate on new examples. This classifier should take advantage of the unlabeled examples in $\mathcal{U}$.

Transductive inference: We consider $\mathcal{U}$ to be the test set, *i.e.* all test examples are jointly provided in advance. Prediction for test example $\mathbf{u}_i$ may therefore depend on the representations and predictions of all other test examples. In this *transductive* setup the models are not required to provide predictions for any example that is not in $\mathcal{U}$.

3.2. Label propagation (LP)

Let $\{\mathbf{x}_1, ..., \mathbf{x}_N\}$, with $\mathbf{x}_i \in \mathbb{R}^d$, be a set of features for N examples. Each feature represents a graph node. We construct an adjacency matrix $S \in \mathbb{R}^{N \times N}$ with zero diagonal, and s_{ij} equal to $\mathbf{x}_i^\top \mathbf{x}_j$ if $\mathbf{x}_j$ is in the k-nearest neighbors of $\mathbf{x}_i$ (denoted by $\mathbf{x}_j \in \text{kNN}(\mathbf{x}_i)$), and 0 otherwise. We obtain a symmetric adjacency matrix by $\bar{S} = S + S^\top$, and its symmetrically normalized version by $\hat{S} = D^{-\frac{1}{2}} \bar{S} D^{-\frac{1}{2}}$, where $D = \operatorname{diag}(\bar{S}\mathbf{1}_N)$ is the degree matrix, and $\mathbf{1}_N$ is the all-ones N-dimensional vector. We assume the first C examples, and the corresponding nodes, to be labeled among C classes; each class is assigned to a single node[1].

Transductive inference. Label propagation [44] is originally proposed for the transductive inference setup; we need to predict labels for the unlabeled nodes of the graph. Given the normalized adjacency matrix $\hat{S}$, label propagation is an iterative process given by

$$\hat{\mathbf{y}}_c^{(t+1)} = \alpha \hat{S} \hat{\mathbf{y}}_c^{(t)} + (1 - \alpha)\mathbf{y}_c \quad \forall c \in \{1, ..., C\} \tag{1}$$

until convergence. Where $\alpha \in (0, 1)$ is a propagation hyper-parameter, $\mathbf{y}_c = \mathbf{e}_c \in \{0, 1\}^N$ is a one-hot vector with the non-zero element at index c, and t is the current iteration. Prediction of the label for an unlabeled node $j \in \{C + 1, ..., N\}$ is then given by

$$\hat{y}_j = \operatorname{argmax}_c \hat{\mathbf{y}}_c(j), \tag{2}$$

where $\hat{\mathbf{y}}_c(j) = \mathbf{e}_j^T \hat{\mathbf{y}}_c$ is the j-th element of the vector $\hat{\mathbf{y}}_c$. One can show [44] that this iterative solution is equivalent to solving C linear systems

$$L\hat{\mathbf{y}}_c = \mathbf{y}_c \quad \forall c \in \{1, ..., C\}, \tag{3}$$

where $L = I - \alpha \hat{S}$ is the graph Laplacian. These linear systems have a closed-form solution

$$\hat{\mathbf{y}}_c = L^{-1}\mathbf{y}_c = L_{\text{inv}}\mathbf{y}_c. \tag{4}$$

However, this closed-form solution is not practical for large datasets as the inverse graph Laplacian L_{inv} is a non-sparse $\mathbb{R}^{N \times N}$ matrix. For this reason it is usual [3, 7, 13, 15] to solve (3) using the conjugate-gradient (CG) method, which is known to be faster than running the iterative solution [13]. Using CG is possible because L is positive-definite.

Observe that (4) simply picks one of the columns of L_{inv}. Matrix element $L_{\text{inv}}(j, c)$ is the confidence of example j belonging to class c. Its values are similarities, after label propagation, between each node pair. It is a type of geodesic similarity that captures the geometry of the feature space as this is indicated by the graph structure. Focusing on a classification task, we are only interested in similarities between an unlabeled example and a class node.

Dual solution. Herein, we show that solving C linear systems of the form in (4) to obtain predictions for all unlabeled nodes using (2) is equivalent to solving $N - C$ linear systems of form

$$\hat{\mathbf{z}}_j = L^{-1}\mathbf{e}_j \quad \forall j \in \{C + 1, ..., N\}, \tag{5}$$

and obtaining the unlabeled node prediction using

$$\hat{y}_j = \operatorname{argmax}_c \hat{\mathbf{z}}_j(c). \tag{6}$$

[1]The theoretical part described in this section holds for the case of more labeled examples per class too. We consider this specific case for simplicity of the presentation and because it corresponds to the task of zero-shot classification with unlabeled examples.

This comes from the fact that

$$\hat{\mathbf{z}}_j(c) = \mathbf{e}_c^T L^{-1} \mathbf{e}_j = \mathbf{e}_j^T L^{-1} \mathbf{e}_c = \hat{\mathbf{y}}_c(j). \tag{7}$$

Although we present the dual solution using the closed-form (4), the same holds with the CG solution of (3). Using the dual solution (5) is not practical for transductive learning as usually the unlabeled nodes are many more than the labeled ones. However, we show that this dual solution is efficiently used for inductive inference.

As discussed, we can view L_{inv} as a pairwise similarity matrix. The confidence of example j belonging to class c, due to symmetry of L_{inv}, is equivalently obtained either by $L_{\text{inv}}(j, c)$ or $L_{\text{inv}}(c, j)$. This constitutes, an additional interpretation of the duality in the solution.

Inductive inference. Test examples now come individually and are not known during graph construction. A possible way to perform inductive inference is by adding the new node to the graph, which is expensive for a test-time operation as $\hat{S}$ would have to be updated for each new test example. Instead, inspired by [13] that uses LP for retrieval, we construct indicator vector $\mathbf{y}_\mathbf{x} \in \mathbb{R}^N$ for test example $\mathbf{x}$ such that

$$\mathbf{y}_\mathbf{x}(j) = \begin{cases} \mathbf{x}^T \mathbf{x}_j, & \text{if } \mathbf{x}_j \in \text{kNN}\,(\mathbf{x}) \\ 0, & \text{otherwise.} \end{cases} \tag{8}$$

Then, we solve linear system

$$\hat{\mathbf{z}}_\mathbf{x} = L^{-1} \mathbf{y}_\mathbf{x}, \tag{9}$$

as in the dual formulation (5) and get a prediction with $\hat{y}_\mathbf{x} = \text{argmax}_c\, \hat{\mathbf{z}}_\mathbf{x}(c)$. With the usual formulation of label propagation in (4), C linear systems need to be solved to get prediction for a single test example. The dual formulation allows us to do it by solving only a single linear system.

Fast inductive inference with sparsification. We further introduce an additional off-line step, where we solve (4), get $\hat{\mathbf{y}}_c$ for all $c \in \{1, ..., C\}$, and store them in a matrix $\hat{Y} = [\hat{\mathbf{y}}_1; ...; \hat{\mathbf{y}}_c] \in \mathbb{R}^{N \times C}$. Then, the solution for a test example is equivalent to a weighted sum of rows of $\hat{Y}$ [1], which is a byproduct of using the indicator vector (8) for representing a test example. Its prediction is given by $\hat{\mathbf{z}}_\mathbf{x} = \mathbf{y}_\mathbf{x}^T \hat{Y}$, and is equivalent to that obtained via (9). However, storing the whole $\hat{Y}$ can be expensive for very large values of N and C. We propose to sparsify $\hat{Y}$ by keeping only the largest values in each row, column, or over the whole matrix.

Note that [14] proposes a low-rank decomposition of the inverse graph Laplacian for the task of retrieval. Our solution is tailored to zero-shot classification, and we choose to obtain and sparsify the first C rows of L_{inv} instead of approximating the whole matrix. Additionally, our solution requires one (sparse) vector to matrix multiplications at test-time instead of two.

3.3. LP for zero-shot VLM classification

We are given a set of classes $\mathcal{C}$ with extracted VLM representations $\{\mathbf{w}_1, ..., \mathbf{w}_C\}$ and a set of unlabeled images $\mathcal{U}$ with extracted VLM representations $\{\mathbf{u}_1, ..., \mathbf{u}_M\}$. We use them as nodes $\{\mathbf{w}_1, ..., \mathbf{w}_C, \mathbf{u}_1, ..., \mathbf{u}_M\}$ of the graph for label propagation. Nodes of class representations (text nodes) are labeled and image nodes unlabeled.

To construct the adjacency matrix S, we need to perform the k-nearest neighbor search between nodes. However, it is known that there exists a large modality gap between image and text representations coming from VLMs [21, 38, 47]. The respective similarity distributions for CLIP are shown in Figure 3a. This modality gap makes standard kNN search between nodes not useful for label propagation; image nodes mostly get connected to image nodes, and text nodes mostly get connected to text nodes. As a consequence, few edges exist between labeled and unlabeled nodes.

To alleviate this problem, we perform the kNN search *separately* for connecting image nodes to image nodes and for connecting image nodes and text nodes. We do not perform the search using text nodes as queries, *i.e.* text nodes get linked only if they appear in the kNN list of an image. This way we also avoid linking text nodes with each other, which is beneficial as each of them is labeled to a different class. Formally, the values of the adjacency matrix are

$$s_{ij} = \begin{cases} \mathbf{u}_i^T \mathbf{u}_j, & \text{if } \mathbf{u}_j \in \text{kNN}_\mathbf{u}(\mathbf{u}_i) \\ \mathbf{u}_i^T \mathbf{w}_j, & \text{if } \mathbf{w}_j \in \text{kNN}_\mathbf{w}(\mathbf{u}_i) \\ 0, & \text{otherwise,} \end{cases} \tag{10}$$

where $\text{kNN}_\mathbf{u}$ and $\text{kNN}_\mathbf{w}$ denote that the search is performed within the image or class features only, respectively.

Moreover, during inductive inference for image $\mathbf{u}$, we perform the kNN search in a similar way to construct indicator vector $\mathbf{y}_\mathbf{u}$ whose elements are given by

$$\mathbf{y}_\mathbf{u}(i) = \begin{cases} \mathbf{u}^T \mathbf{u}_j, & \text{if } \mathbf{u}_j \in \text{kNN}_\mathbf{u}(\mathbf{u}) \\ \mathbf{u}^T \mathbf{w}_j, & \text{if } \mathbf{w}_j \in \text{kNN}_\mathbf{w}(\mathbf{u}) \\ 0, & \text{otherwise.} \end{cases} \tag{11}$$

Due to the two types of edges, *i.e.* image-to-image and image-to-text, we use power function $h(v) = v^\gamma$ to transform the image-to-text (cross-modal) similarities. This way we effectively *balance* their contribution in the graph and the indicator vector. To that end, we use $h(\mathbf{u}_i^T \mathbf{w}_j)$ and $h(\mathbf{u}^T \mathbf{w}_j)$ in (10) and (11), respectively, instead of $\mathbf{u}_i^T \mathbf{w}_j$ and $\mathbf{u}^T \mathbf{w}_j$.

We refer to the proposed method described above as **Z**ero-shot classification with **La**bel **P**ropagation (**ZLaP**). We further denote the variant of our method after sparsifying the $\hat{Y}$ matrix for inductive inference as **ZLaP***.

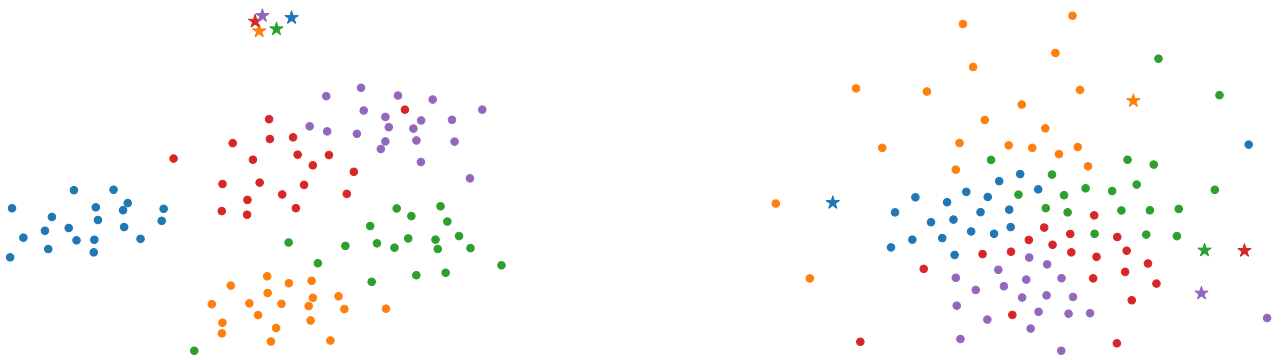
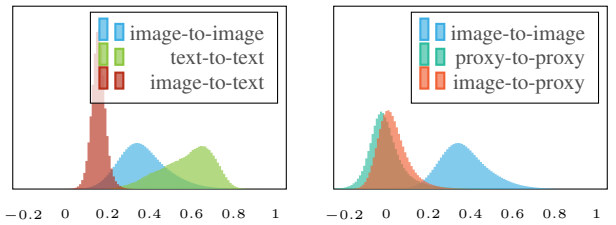


Figure 2. **t-SNE visualization** for the original CLIP features (left) and our geodesic similarity (right). The former is estimated with the features as input, while the latter with the L_{inv} used as a pairwise similarity matrix. $\star$: class representation, $\bullet$: image representation. Figure generated for five random classes from the CUB dataset.



(a) Using text prompts (b) Using proxies from InMaP [30]

Figure 3. **Similarity distributions** among features of the same or different modality, using 7 textual templates [38] (left) or the InMaP proxies (right) as class representations.

t-SNE visualization of the bi-modal space. In Figure 2 we visualize the bi-modal feature space for CLIP features using t-SNE [39] in two cases, *i.e.* the Euclidean case and using geodesic similarities obtained by L_{inv}, *i.e.* after label propagation. When using Euclidean affinities (left), we see that due to the large differences in the similarity distributions (text-text, image-to-image, and text-to-image, as shown in Figure 3a) all class representations (stars) are clustered together far from the image nodes. However, using the geodesic affinities from L_{inv} (right) we see that class representations are more spread.

4. Experiments

In this section, we first present the datasets we use, our experimental setup and competing methods. We then present component analysis for ZLaP and results for transductive and inductive zero-shot classification on 14 datasets.

4.1. Datasets

We evaluate the proposed method on 14 diverse image classification datasets: ImageNet ILSVRC2012 [33], Describable Textures Dataset (DTD) [5], EuroSAT [10], FGVC-Aircraft [23], Oxford Flowers 102 [25], Food-101 [2], Oxford-IIIT Pet [28], SUN397 [42], Stanford Cars [17], Caltech101 [6], UCF101 [36], CIFAR10 [18], CIFAR100 [18], CUB-200-2011 [41]. For the first 11 datasets we borrow the train and test splits from CoOp[45]. We use the official training and test splits for CIFAR10, CIFAR100 and CUB-200-2011.

4.2. Experimental setup

In the transductive (inductive) inference setup, unlabeled nodes in the graph are the test (train) images. We always measure classification accuracy over the test images.

VLMs and textual prompts. We report results using the publicly available ResNet50 and ViT-B/16 CLIP [31] models. We adopt the 7 templates from SuS-X [38] as class prompts for all results apart from Table 3 where we utilize the LLM generated prompts from [29].

Compared methods. Our baseline is zero-shot recognition with **CLIP** [31] using text encoder features as class representations. **TPT** [35] is based on test-time prompt tuning such that different image augmentations produce consistent predictions. The aforementioned methods do not exploit unlabeled data; their performance is therefore unchanged in both inference setups. **CLIP-DN** [46] normalizes feature distributions during test-time and assumes access to the mean feature vector of the target distribution. In the transductive (inductive) setup the mean vector is estimated on the test (training) set. **InMaP** [30] is a concurrent work that extracts updated class representations using pseudo-labels on the unlabeled set. In the transductive (inductive) setup the learning is performed on the test (training) images.

Implementation details. We reproduce results for CLIP[2], CLIP-DN[3], and InMaP[4] using their public implementations. For TPT [35] we report the numbers provided in [30]. We run InMaP using a single set of hyper-parameters for all 14 datasets, *i.e*. the default values reported in the official implementation[4]. We also fix the values of k, γ, and α for ZLaP across all datasets to 5, 5.0, and 0.3, respectively, for CLIP, and 10, 3.0, and 0.3, respectively, for InMaP.

ZLaP variants. We refer to ZLaP using text class representations as **CLIP + ZLaP**. Since InMaP is complementary to our work, we further evaluate the performance of ZLaP when $\{\mathbf{w}_1, \ldots, \mathbf{w}_C\}$ are the InMaP proxies. We refer to this as **InMaP + ZLaP** in the results. We refer to ZLaP with a sparse $\hat{Y}$ for inductive inference as **ZLaP***.

4.3. Components of ZLaP

Bi-modal graph adjustments. In Table 1 we show the importance of two design choices to adapt LP to bi-modal graphs, *i.e*. separating the nearest neighbor search across modalities using (10) and (11), and transforming cross-modal similarities using a power function $h(\cdot)$. We see that separate search is crucial; without it LP is not effective at all. The power function gives an extra boost in both setups, especially in the case of transductive inference. In Table 2 we report the percentage of images that are connected to their groundtruth class nodes within a path of length n, with and without our adjustments. We see that for any such paths to exist for the case without adjustments, k needs to be extremely high. With adjustments, $k = 5$ is enough for 71.4% of the nodes to be connected to the correct class nodes.

Sparsifying matrix $\hat{Y}$ for inductive inference. We explore three ways of approximating $\hat{Y}$ by sparsification, *i.e*. wither keeping only the largest ξ columns per row, the largest ξ rows per column, or the largest ξ elements of the whole matrix. In all cases, the rest of the elements are set to zero. In Figure 4 we show the influence that these three variants have on performance. Not only these variants speed-up inference, but we can also see improvements in performance when sparsification percentage is high, *i.e*. low percentage of non-zero elements. We attribute this to the fact that less confident predictions in $\hat{Y}$, many of them erroneous, are now set to zero. Although the best variant to choose seems to vary per dataset, we found that keeping the top element per row performs well across different datasets. We therefore use the $\xi = 1$ top element per row for our experiments. This amounts to different percentages of sparsity per dataset; we are keeping approximately 2.3% on average

[2] https://github.com/OpenAI/CLIP

[3] https://github.com/fengyuli-dev/distribution-normalization

[4] https://github.com/idstcv/InMaP

Eq.(10)	$h(\cdot)$	ImageNet	DTD	CUB
✗	✗	0.1	2.1	0.5
✗	✓	0.1	2.1	0.5
✓	✗	50.2	32.3	41.6
✓	✓	61.8	41.9	52.1

(a) Transductive inference

Eq.(10)-(11)	$h(\cdot)$	ImageNet	DTD	CUB
✗	✗	0.1	2.1	0.5
✗	✓	0.1	2.1	0.5
✓	✗	60.8	42.4	49.6
✓	✓	62.2	42.8	49.7

(b) Inductive inference

Table 1. **Adjusting LP to bi-modal graphs.** Impact of using separate kNN search for constructing the graph (Eq.(10)) or the indicator vector (Eq.(11)), as well as power function $h(\cdot)$ for balancing the contributions of two types of edges in the graph.

	Joint kNN search			Separate kNN search		
k	n=1	n=2	n=3	n=1	n=2	n=3
5	0.0	0.0	0.0	71.4	85.1	100.0
10	0.0	0.0	0.0	82.9	95.4	100.0
100	40.1	100.0	100.0	100.0	100.0	100.0

Table 2. **Impact of the separate kNN search** on the shortest paths between image nodes and the text node of their class. We report the percentage of images whose shortest path to the text node of their ground-truth class has length equal to or less than n. *Left:* the vanilla approach. *Right:* our separate kNN search using Eq. (10). Analysis on DTD for the transductive setup.

across all datasets. Regarding the inference speed-up, the primal solution takes $\sim$ 2.6 sec per image, the dual takes $\sim$ 4.4 ms, while the sparsified approach takes $\sim$ 0.6 ms, measured on ImageNet dataset.

Using the class proxies from InMaP. We observe that many class-to-image similarities (*e.g*. $\mathbf{u}_i^\top \mathbf{w}_j$ in (10)) become negative on some datasets when using ZLaP with InMaP proxies (see Figure 3b). We therefore perform min-max normalization in range $[0, 1]$ after constructing adjacency matrix S or the indicator vector, for the transductive and inductive inference setups.

4.4. Results

Transductive inference. We present results for transductive zero-shot classification in Figure 5. ZLaP improves the zero-shot performance of CLIP significantly on all datasets. It also outperforms the recent TPT and CLIP-DN approaches on the vast majority of cases, with large gains

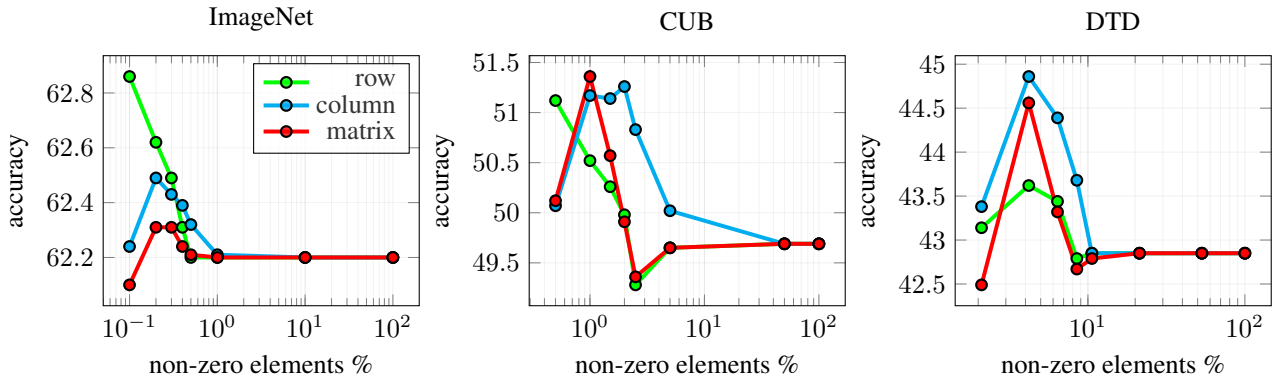
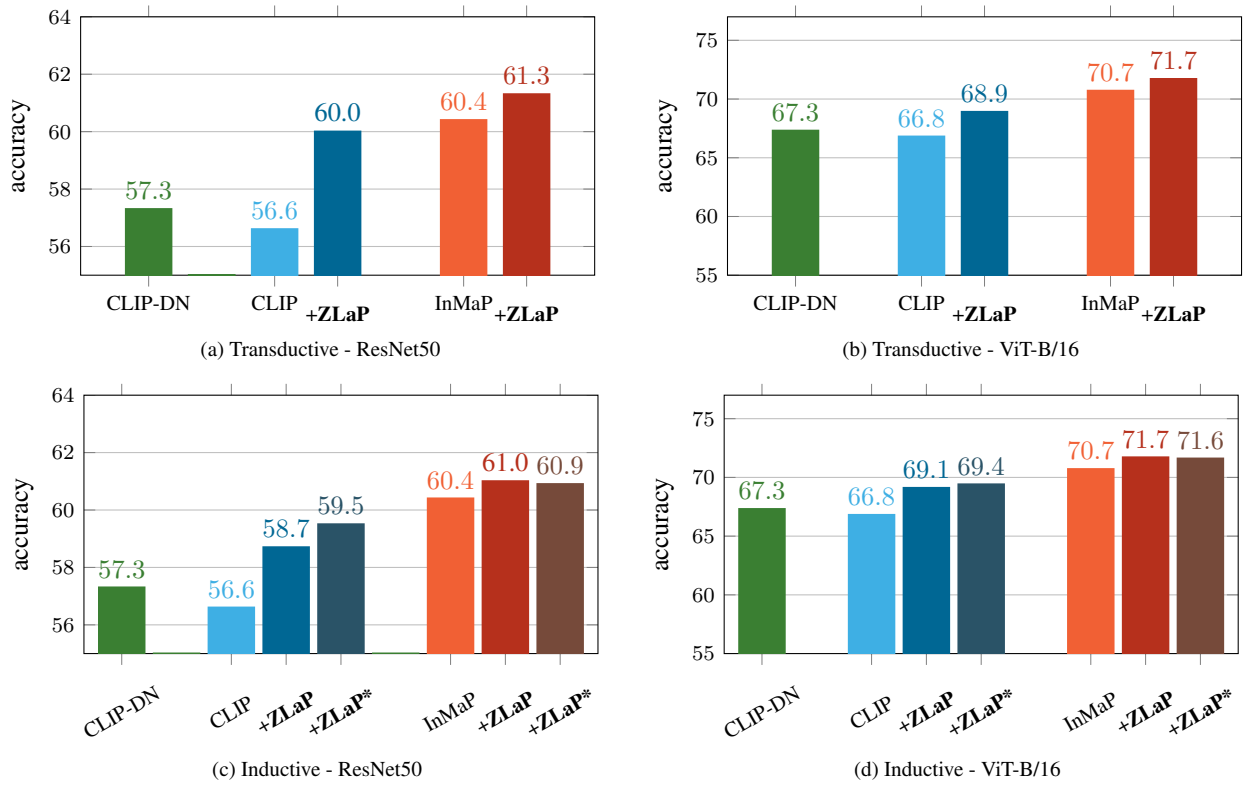


Figure 4. **Sparcifying matrix $\hat{Y}$ for inductive CLIP+ZLaP:** effect of maintaining only the top elements per row/column/matrix.



(a) Transductive - ResNet50

(b) Transductive - ViT-B/16

(c) Inductive - ResNet50

(d) Inductive - ViT-B/16

Figure 5. **Zero-shot classification accuracy averaged over 14 datasets** for the transductive (top) and inductive (bottom) setups. Results per dataset are reported in the supplementary material.

in average accuracy. Compared to InMaP, ZLaP offers lower accuracy on average. However, by incorporating InMaP's class representations to our graph, we can improve our results even further and outperform all other methods, for an improvement of approximately +5% over CLIP with both backbones.

Inductive inference. We report results for the inductive inference setup in Figure 5. ZLaP achieves a noticeable improvements over the CLIP baseline in this setup as well. Gains are more prominent for the case of ZLaP using the InMaP proxies, where gains over CLIP are +4.4% and +4.9% for the two backbones. We also observe that, although InMaP slightly outperforms ZLaP when used over CLIP, the combination of the two achieves *state-of-the-art performance* in this case as well. We further see that ZLaP* retains the state-of-the-art performance of our method, while sparsifying $\hat{Y}$ offers significant speed-up at inference time.

	Transductive	Inductive
Results with RN50		
CLIP	63.0	63.0
+ **ZLaP**	64.6	64.2
InMaP	64.8	64.6
+ **ZLaP**	**65.8**	**65.0**
Results with ViT-B-16		
CLIP	71.9	71.9
+ **ZLaP**	72.6	73.3
InMaP	73.9	74.0
+ **ZLaP**	**74.8**	**74.2**

Table 3. **Zero-shot classification using prompts generated by LLMs [29].** We report average accuracy on 12 datasets using prompts from CuPL [29] together with our 7 standard prompts. Results per dataset are reported in the supplementary material.

Leveraging LLM generated prompts. In Table 3 we report average zero-shot classification accuracy for ZLaP using the prompts recently proposed in CuPL [29]. These are prompts generated by LMMs that are available on the CuPL Github page[5] for 12 of the datasets we use (all datasets besides CUB and Eurosat). ZLaP improves zero-shot performance in this case as well, for both the transductive and inductive setups. This verifies that our method is complementary to improved prompt engineering.

Multi-label classification. We apply ZLaP for multi-label classification on the MS-COCO [22] dataset. ZLaP improves the zero-shot performance of CLIP by +6.0% mAP (56.8% vs. 50.8%) for inductive inference without any modification of the approach or its hyper-parameters.

Web-crawled unlabeled images. All previous experiments use unlabeled images that come from the target distribution, *i.e.* they are known to depict one of the classes of interest but their labels are discarded. To see the impact of ZLaP in a more realistic setup using web-crawled images we rely on LAION-400M [34] composed by image-caption pairs. We construct the set of unlabeled images with 10,000 images per class that are chosen either randomly, or based on proximity of their image or text features to the class representation. Random selection fails, but the other two options provide some improvement compared to CLIP, with the caption-based neighbors being a bit better. The complete set of results is presented in the supplementary material.

[5] https://github.com/sarahpratt/CuPL

	Transductive	Inductive
BLIP [20]	54.6	54.6
+ **ZLaP**	**59.6**	**57.9**
ALBEF [19]	36.0	36.0
+ **ZLaP**	**41.2**	**46.8**
EVA-CLIP-8B [37]	83.6	83.6
+ **ZLaP**	**84.6**	**84.5**
EVA-CLIP-18B [37]	83.9	83.9
+ **ZLaP**	**84.8**	**84.7**

Table 4. **Accuracy on ImageNet using different VLMs.**

Different VLMs We use CLIP as the VLM of choice throughout our experiments. In Table 4, we present results when ZLaP is applied on top of four recent VLMs, namely BLIP [20], ALBEF [19], and two versions of EVA-CLIP [37]. We use the implementations of BLIP and ALBEF that are available in the LAVIS library[6], while for EVA-CLIP we use implementation from the official Github repository[7]. ZLaP improves the results of all four different VLMs in both transductive and inductive setups.

5. Conclusions

Label propagation is an intuitive way of encoding the global structure of unlabeled data into geodesic distances over a locally Euclidean space. In this paper, we show that this method can be successfully tailored to both transductive and inductive zero-shot classification with vision-language models, and achieve state-of-the-art performance on both setups. To that end, we show that it is highly important to take proper care of the peculiarities of the bi-modal nature of the task during graph construction. We further carefully design an efficient variant of label propagation for the inductive inference case, that may enable label propagation to be applied to other tasks beyond zero-shot classification.

Vision-language models trained on billion-scale datasets are redefining computer vision research. The proposed ZLaP is a training-free approach able to improve the generalization performance of black-box VLMs using only unlabeled data, for an annotation-free, text-based and open-world classification paradigm that will inevitably be ubiquitous in the near future.

Acknowledgements. This work was supported by the Junior Star GACR GM 21- 28830M and the Czech Technical University in Prague grant No. SGS23/173/OHK3/3T/13. We thank Ahmet Iscen for many helpful comments.

[6] https://github.com/salesforce/LAVIS

[7] https://github.com/baaivision/EVA/tree/master/EVA-CLIP-18B

References

[1] Yoshua Bengio, Olivier Delalleau, and Nicolas Le Roux. Label propagation and quadratic criterion. In *Semi-Supervised Learning*, pages 192–216. The MIT Press, 2006. 4

[2] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101 – mining discriminative components with random forests. In *ECCV*, 2014. 5

[3] Siddhartha Chandra and Iasonas Kokkinos. Fast, exact and multi-scale inference for semantic image segmentation with deep Gaussian CRFs. In *ECCV*, 2016. 3

[4] Mehdi Cherti, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia Jitsev. Reproducible scaling laws for contrastive language-image learning. In *CVPR*, 2023. 2

[5] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, , and A. Vedaldi. Describing textures in the wild. In *CVPR*, 2014. 5

[6] 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 *CVPRW*, 2004. 5

[7] Leo Grady. Random walks for image segmentation. *pami*, 28(11):1768–1783, 2006. 3

[8] Ziyu Guo, Renrui Zhang, Longtian Qiu, Xianzheng Ma, Xupeng Miao, Xuming He, and Bin Cui. CALIP: zero-shot enhancement of CLIP with parameter-free attention. In *AAAI*, 2023. 1, 2

[9] Ruifei He, Shuyang Sun, Xin Yu, Chuhui Xue, Wenqing Zhang, Philip H. S. Torr, Song Bai, and Xiaojuan Qi. Is synthetic data from generative models ready for image recognition? In *ICLR*, 2023. 2

[10] 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*, 2019. 5

[11] Xuefeng Hu, Ke Zhang, Lu Xia, Albert Chen, Jiajia Luo, Yuyin Sun, Ken Wang, Nan Qiao, Xiao Zeng, Min Sun, Cheng-Hao Kuo, and Ram Nevatia. Reclip: Refine contrastive language image pre-training with source free domain adaptation. In *WACV*, 2024. 2

[12] Tony Huang, Jack Chu, and Fangyun Wei. Unsupervised prompt learning for vision-language models. *arXiv preprint arXiv:2204.03649*, 2022. 2

[13] Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, Teddy Furon, and Ondřej Chum. Efficient diffusion on region manifolds: Recovering small objects with compact CNN representations. In *CVPR*, 2017. 3, 4

[14] Ahmet Iscen, Yannis Avrithis, Giorgos Tolias, Teddy Furon, and Ondrej Chum. Fast spectral ranking for similarity search. In *CVPR*, 2018. 4

[15] Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, and Ondrej Chum. Label propagation for deep semi-supervised learning. In *CVPR*, 2019. 3

[16] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V. Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *ICML*, 2021. 2

[17] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *ICCVW*, 2013. 5

[18] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009. 5

[19] Junnan Li, Ramprasaath R. Selvaraju, Akhilesh Gotmare, Shafiq R. Joty, Caiming Xiong, and Steven Chu-Hong Hoi. Align before fuse: Vision and language representation learning with momentum distillation. In *NeurIPS*, 2021. 8

[20] Junnan Li, Dongxu Li, Caiming Xiong, and Steven C. H. Hoi. BLIP: bootstrapping language-image pre-training for unified vision-language understanding and generation. In *ICML*, 2022. 8

[21] Weixin Liang, Yuhui Zhang, Yongchan Kwon, Serena Yeung, and James Y. Zou. Mind the gap: Understanding the modality gap in multi-modal contrastive representation learning. In *NeurIPS*, 2022. 4

[22] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. In *ECCV*, 2014. 8

[23] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew B. Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013. 5

[24] Sachit Menon and Carl Vondrick. Visual classification via description from large language models. In *ICLR*, 2023. 1, 2

[25] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *Indian Conference on Computer Vision, Graphics and Image Processing*, 2008. 5

[26] Zachary Novack, Julian J. McAuley, Zachary Chase Lipton, and Saurabh Garg. Chils: Zero-shot image classification with hierarchical label sets. In *ICML*, 2023. 2

[27] Omiros Pantazis, Gabriel J. Brostow, Kate E. Jones, and Oisin Mac Aodha. Svl-adapter: Self-supervised adapter for vision-language pretrained models. In *BMVC*, 2022. 2

[28] Omkar M. Parkhi, Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. Cats and dogs. In *CVPR*, 2012. 5

[29] Sarah M. Pratt, Rosanne Liu, and Ali Farhadi. What does a platypus look like? generating customized prompts for zero-shot image classification. In *ICCV*, 2023. 1, 2, 5, 8

[30] Qi Qian, Yuanhong Xu, and Juhua Hu. Intra-modal proxy learning for zero-shot visual categorization with clip. In *NeurIPS*, 2023. 1, 2, 5, 6

[31] 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. In *ICML*, 2021. 1, 2, 5

[32] 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*, 2023. 2

[33] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *IJCV*, 2015. 5

[34] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. LAION-5B: an open large-scale dataset for training next generation image-text models. In *NeurIPS*, 2022. 2, 8

[35] 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. In *NeurIPS*, 2022. 2, 5, 6

[36] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. UCF101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012. 5

[37] Quan Sun, Jinsheng Wang, Qiying Yu, Yufeng Cui, Fan Zhang, Xiaosong Zhang, and Xinlong Wang. EVA-CLIP-18B: scaling CLIP to 18 billion parameters. *arXiv preprint arXiv:2402.04252*, 2024. 8

[38] Vishaal Udandarao, Ankush Gupta, and Samuel Albanie. Sus-x: Training-free name-only transfer of vision-language models. In *ICCV*, 2023. 2, 4, 5

[39] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *JMLR*, 9(11), 2008. 5

[40] Matthew Wallingford, Vivek Ramanujan, Alex Fang, Aditya Kusupati, Roozbeh Mottaghi, Aniruddha Kembhavi, Ludwig Schmidt, and Ali Farhadi. Neural priming for sample-efficient adaptation. In *NeurIPS*, 2023. 2

[41] Peter Welinder, Steve Branson, Takeshi Mita, Catherine Wah, Florian Schroff, Serge Belongie, and Pietro Perona. Caltech-ucsd birds 200. 2010. 5

[42] Jianxiong Xiao, James Hays, Krista A. Ehinger, Aude Oliva, and Antonio Torralba. SUN database: Large-scale scene recognition from abbey to zoo. In *CVPR*, 2010. 5

[43] Renrui Zhang, Xiangfei Hu, Bohao Li, Siyuan Huang, Hanqiu Deng, Yu Qiao, Peng Gao, and Hongsheng Li. Prompt, generate, then cache: Cascade of foundation models makes strong few-shot learners. In *CVPR*, 2023. 2

[44] Dengyong Zhou, Olivier Bousquet, Thomas Navin Lal, Jason Weston, and Bernhard Schölkopf. Learning with local and global consistency. In *NIPS*, 2003. 3

[45] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *IJCV*, 2022. 5

[46] Yifei Zhou, Juntao Ren, Fengyu Li, Ramin Zabih, and Ser-Nam Lim. Test-time distribution normalization for contrastively learned vision-language models. In *NeurIPS*, 2023. 1, 2, 5

[47] Xiangyang Zhu, Renrui Zhang, Bowei He, Aojun Zhou, Dong Wang, Bin Zhao, and Peng Gao. Not all features matter: Enhancing few-shot clip with adaptive prior refinement. In *ICCV*, 2023. 4