

PROTO-CLIP: Vision-Language Prototypical Network for Few-Shot Learning

Jishnu Jaykumar P, Kamalesh Palanisamy, Yu-Wei Chao, Xinya Du, Yu Xiang

Abstract—We propose a novel framework for few-shot learning by leveraging large-scale vision-language models such as CLIP [1]. Motivated by unimodal prototypical networks for few-shot learning, we introduce PROTO-CLIP which utilizes image prototypes and text prototypes for few-shot learning. Specifically, PROTO-CLIP adapts the image and text encoder embeddings from CLIP in a joint fashion using few-shot examples. The embeddings from the two encoders are used to compute the respective prototypes of image classes for classification. During adaptation, we propose aligning the image and text prototypes of the corresponding classes. Such alignment is beneficial for few-shot classification due to the reinforced contributions from both types of prototypes. PROTO-CLIP has both training-free and fine-tuned variants. We demonstrate the effectiveness of our method by conducting experiments on benchmark datasets for few-shot learning, as well as in the real world for robot perception¹.

I. INTRODUCTION

Building autonomous robots that can help people perform various tasks is the dream of every roboticist. Nowadays, most robots are working in factories and warehouses by performing pre-programmed repetitive tasks such as assembling and delivering. In the future, we believe that there will be intelligent robots that can perform tasks in human environments autonomously. For example, people can instruct a robot by saying “bring me a bottle of water” or “wash the mug on the table”, then the robot will execute the instructions accordingly. In these scenarios, robots need to recognize objects from sensory data in order to understand the required tasks. In this work, we develop a novel few-shot learning method that can enable robots to recognize novel objects from just a few example images per object. We believe that few-shot learning [2] is a promising paradigm to enable robots to recognize a large number of objects. The appeal lies in the ease of data collection—just a few example images is sufficient for teaching a robot a novel object. On the contrary, object model-based approaches build 3D models of objects and then use these 3D models [3] for object recognition. Object category-based approaches focus on recognizing category labels of objects such as 80 categories in the MSCOCO dataset [4]. The limitation of model-based object recognition is the difficulty of obtaining a large number of 3D models for many objects in the real world. Current 3D scanning techniques cannot deal well with metal objects or transparent

objects. For category-based object recognition, it is difficult to obtain a large number of images for each category in robotic settings. Large-scale datasets for object categories such as ImageNet [5] and Visual Genome [6] are collected from the Internet. These Internet images are not very suitable for learning object representations for robot manipulation due to domain differences. Due to the limitations of model-based and category-based object recognition, if a robot can learn to recognize a new object from a few images of the object, it is likely to scale up the number of objects that the robot can recognize in the real world.

The main challenge in few-shot learning is how to achieve generalization with very limited training examples. Learning good visual representations is the key to achieve good performance in few-shot learning [7]. Although the Internet images are quite different from robot manipulation settings, they can be used to learn powerful visual representations. Recently, the CLIP (Contrastive Language–Image Pre-training) model [1] trained with a large number of image-text pairs from the Internet achieves promising *zero-shot* image recognition performance. Using the visual and language representations from CLIP, several few-shot learning approaches [8], [9], [10] are proposed to improve the zero-shot CLIP model. [9], [10] adapt the CLIP image encoder to learn better feature representations, while [8] learns prompts for the CLIP model. On the other hand, few-shot learning approaches are studied in the meta-learning framework [11]. These approaches are aimed at generalizing to novel classes after training. A notable method is Prototypical Network [12] and its variants [13], [14], which demonstrate effective performance for few-shot learning. However, these methods do not leverage the powerful feature representation of CLIP.

These observations motivate us to leverage CLIP in prototypical networks for few-shot learning. We notice that existing methods for adapting CLIP models in few-shot learning adapt the image encoder [9], [10] or the text encoder [8] in CLIP. We argue that if we can use both the image encoder and the text encoder for classification and jointly adapt them using few-shot training images, we can improve the few-shot classification performance. To achieve this goal, we propose PROTO-CLIP, a new model motivated by the traditional unimodal Prototypical Networks [12]. PROTO-CLIP utilizes image prototypes and text prototypes computed from adapted CLIP encoders for classification. In addition, we propose to align the image prototype and the text prototype of the same class during adaptation. In this way, both the image encoder and the text encoder can contribute to the classification while achieving agreement between their predictions. Fig. 1

Jishnu Jaykumar P, Kamalesh Palanisamy, Xinya Du and Yu Xiang are with the Department of Computer Science, University of Texas at Dallas, Richardson, TX 75080, USA {firstname.lastname}@utdallas.edu

Yu-Wei Chao is with NVIDIA, Seattle, WA 98105, USA ychao@nvidia.com

¹Project page: <https://irvlutd.github.io/Proto-CLIP>

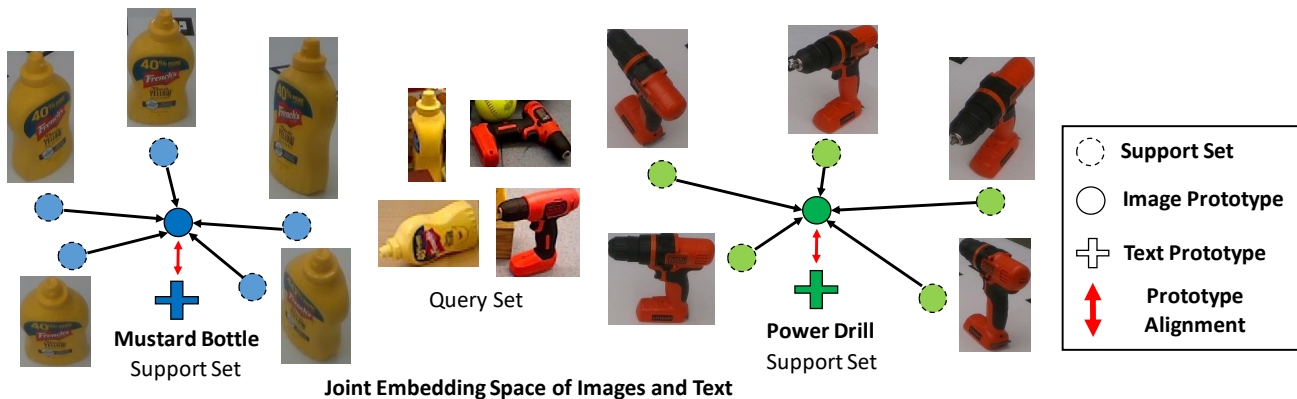


Fig. 1: Our PROTO-CLIP model learns a joint embedding space of images and text, where image and text prototypes formed using support sets are learned and aligned for few-shot classification.

illustrates the concept of learning the joint embedding space of images and text from PROTO-CLIP.

To verify the effectiveness of PROTO-CLIP, we have conducted experiments on commonly used benchmarks for few-shot image classification, as well as the FewSOL dataset introduced for few-shot object learning in robotic environments [15]. In addition, we have built a robotic system that integrates Automatic Speech Recognition (ASR), few-shot object recognition using PROTO-CLIP and robotic grasping to demonstrate the robotic application of PROTO-CLIP.

II. RELATED WORK

In the context of image recognition, few-shot learning indicates using a few images per image category. The problem is usually formulated as “ N -way, K -shot”, i.e., N classes with K images per class. In the traditional image classification setup, these NK images are used as training images. Once a model is trained, it can be used to test images among N classes. Recent CLIP-based few-shot learning methods fall into this setting.

CLIP-based Few-Shot Learning. The CLIP [1] model applies contrastive learning to image-text pairs from the Internet. It consists of an image encoder and a text encoder for the extraction of features from images and text, respectively. Its training objective is to maximize the similarity between the corresponding image and text in a pair in a high-dimensional joint feature space. After training, CLIP can be used for zero-shot image classification by comparing image features with text embeddings of novel class names. This model is denoted as zero-shot CLIP. When a few training images are available for each class, several approaches are proposed to improve zero-shot CLIP. The linear-probe CLIP model [1] trains a logistic regression classifier using CLIP image features. CoOp [8] proposes to use learnable vectors as a prompt for the CLIP text encoder for few-shot learning. CLIP-Adapter [9] learns two layers of linear transformations on top of the image encoder and the text encoder with residual connections, respectively, to adapt CLIP features for few-shot learning. Tip-Adapter [10] builds a key-value cache model, where keys are CLIP image features and values are one-hot vectors of the class labels. Given a query image, its image feature is

compared with the cache keys to combine the value labels for classification. Tip-Adapter can also fine-tune the keys by treating them as learnable parameters, which further improves the few-shot classification accuracy. Sus-X [16] leverages the power of Stable Diffusion [17] to create support sets and aims to address the issue of uncalibrated intra-modal embedding distances in TIP-Adapter [10] by utilizing inter-modal distances as a connecting mechanism. Table I compares our proposed method with existing CLIP-model-based few-shot learning methods. By using the image prototypes and text prototypes for classification, our method can adapt both the image embeddings and text embeddings from CLIP. In addition, the model aligns the image prototypes and the text prototypes, which serves as a regularization term in adapting the feature embeddings. We empirically verify our model by conducting experiments on benchmark datasets for few-shot learning.

Meta-learning-based Few-Shot Learning. In parallel with these efforts to adapt CLIP for few-shot learning, meta-learning-based approaches are also proposed for few-shot learning. While previous CLIP-based models are tested on the same classes in training, the focus here is to learn a model on a set of training classes C_{train} that can generalize to novel classes C_{test} in testing. Each class contains a support set and a query set. During training, the class labels for both sets are available. During testing, only the class labels of the support set are available, and the goal is to estimate the labels of the query set. Meta-learning-based approaches train a meta-learner with the training classes C_{train} that can be adapted to the novel classes C_{test} using their support sets. Non-episodic approaches use all the data in C_{train} for training such as k -NN and its ‘Finetuned’ variants [18], [19], [20], [7]. Episodic approaches construct episodes, i.e., a subset of the training classes, to train the meta-learner. Representative episodic approaches include Prototypical Networks [12], Matching Networks [21], Relation Networks [22], Model Agnostic Meta-Learning (MAML) [11], Proto-MAML [13] and CrossTransformers [14]. The META-DATASET [13] was introduced to benchmark few-shot learning methods in this setting. In this work, we consider training and testing in the same classes following previous CLIP-based few-shot learning

Method	Use Support Sets	Adapt Image Embedding	Adapt Text Embedding	Align Image and Text
Zero-shot CLIP [1]	✗	✗	✗	✓
Linear-probe CLIP [1]	✓	✓	✗	✗
CoOp [8]	✓	✗	✓	✗
CLIP-Adapter [9]	✓	✓	✓	✗
Tip-Adapter [10]	✓	✓	✓	✗
Sus-X [16]	✓	✓	✗	✗
PROTO-CLIP (Ours)	✓	✓	✓	✓

TABLE I: Comparison of our proposed method with the existing CLIP-based few-shot learning methods. “Use Support Sets” indicates if a method uses support training sets for fine-tuning. “Adapt Image/Text Embedding” indicates if a method adapts the image/text embeddings obtained from CLIP. “Align Image and Text” indicates if a method *specifically* aligns images and text in the feature space.

methods [8], [9], [10].

III. METHOD

We consider the N -way K -shot classification problem. In few-shot settings, $K \ll N$. The image set with class labels is considered as the *support set*: $\mathcal{S} = \{\mathbf{x}_i^s, y_i^s\}_{i=1}^M$, where \mathbf{x}_i^s denotes a support image, $y_i^s \in \{1, 2, \dots, N\}$ denotes the class label of the support image, and M is the size of the support set. In N -way K -shot settings, $M = NK$. The goal of few-shot classification is to classify the *query set* $\mathcal{Q} = \{\mathbf{x}_j^q\}_{j=1}^L$, i.e., L test images without class labels. Specifically, we want to estimate the conditional probability $P(y = k | \mathbf{x}^q, \mathcal{S})$ that models the probability distribution of the class label y given a query image \mathbf{x}^q and the support set \mathcal{S} .

Our PROTO-CLIP model (Fig. 2). We propose to leverage both the image encoder and the text encoder in the CLIP model [1] to estimate the conditional probability of class label as

$$P(y = k | \mathbf{x}^q, \mathcal{S}) = \underbrace{\alpha P(y = k | \mathbf{x}^q, \mathcal{S}_x)}_{\text{image probability}} + (1 - \alpha) \underbrace{P(y = k | \mathbf{x}^q, \mathcal{S}_y)}_{\text{text probability}}, \quad (1)$$

where $\mathcal{S}_x = \{\mathbf{x}_i^s\}_{i=1}^M$ and $\mathcal{S}_y = \{y_i^s\}_{i=1}^M$ denote the image set and the label set of the support set \mathcal{S} , respectively, and $\alpha \in [0, 1]$ is a hyper-parameter to combine the two probabilities. To model the probability distributions conditioned on \mathcal{S}_x or \mathcal{S}_y , we leverage the prototypical networks [12]:

$$P(y = k | \mathbf{x}^q, \mathcal{S}_x) = \frac{\exp(-\beta \|g_{\mathbf{w}_1}(\mathbf{x}^q) - \mathbf{c}_k^x\|_2^2)}{\sum_{k'=1}^N \exp(-\beta \|g_{\mathbf{w}_1}(\mathbf{x}^q) - \mathbf{c}_{k'}^x\|_2^2)}, \quad (2)$$

$$P(y = k | \mathbf{x}^q, \mathcal{S}_y) = \frac{\exp(-\beta \|g_{\mathbf{w}_1}(\mathbf{x}^q) - \mathbf{c}_k^y\|_2^2)}{\sum_{k'=1}^N \exp(-\beta \|g_{\mathbf{w}_1}(\mathbf{x}^q) - \mathbf{c}_{k'}^y\|_2^2)}, \quad (3)$$

where $g_{\mathbf{w}_1}(\cdot)$ denotes the CLIP image encoder plus an adapter network with learnable parameters \mathbf{w}_1 used to compute the feature embeddings of query images. The CLIP image encoder is pretrained and then frozen. \mathbf{c}_k^x and \mathbf{c}_k^y are the “prototypes” for class k computed using images and text, respectively. $\beta \in \mathbb{R}^+$ is a hyperparameter to sharpen the probability distributions. We have the prototypes as

$$\mathbf{c}_k^x = \frac{1}{M_k} \sum_{y_i^s=k} \phi_{\text{Image}}(\mathbf{x}_i^s) \quad (4)$$

$$\mathbf{c}_k^y = \frac{1}{\tilde{M}_k} \sum_{j=1}^{\tilde{M}_k} \phi_{\text{Text}}(\text{Prompt}_j(y_i^s = k)), \quad (5)$$

where M_k is the number of examples with label k , and \tilde{M}_k is the number of prompts for label k . To compute text embeddings, we can either directly input the class names such as “mug” and “plate” into the text encoder, or convert the class names to phrases such as “a photo of mug” and “a photo of plate” and then input the phrases into the text encoder. These phrases are known as *prompts* of the vision-language models. We can use multiple prompts for each class label. $\phi_{\text{Image}}(\mathbf{x}_i^s)$ and $\phi_{\text{Text}}(\text{Prompt}_j(y_i^s = k))$ denote the image embedding and the j th text embedding of the image-label pair (\mathbf{x}_i^s, y_i^s) computed using the CLIP image encoder and the text encoder, respectively. These embeddings with dimension C of the support set form the image memory and the text memory, as shown in Fig. 2. They are learnable embedding vectors initialized by the computed embeddings using the CLIP image encoder and text encoder. We use \mathbf{c}_k^x and \mathbf{c}_k^y to denote the mean of the embeddings of the images and the prompts for class k , respectively. Since the image embeddings and the text embeddings are of the same dimension, we can compute the distance between the text prototype \mathbf{c}_k^y and the image embedding $g_{\mathbf{w}_1}(\mathbf{x}^q)$ in Eq. 3. As we can see, our model leverages prototypical networks with image encoder and text encoder from CLIP. We name it “PROTO-CLIP”.

Learning the memories and the adapter. During training, we can construct a support set $\mathcal{S} = \{\mathbf{x}_i^s, y_i^s\}_{i=1}^M$ and a query set with ground truth labels $\mathcal{Q} = \{\mathbf{x}_j^q, y_j^q\}_{j=1}^L$. Then we can use \mathcal{S} and \mathcal{Q} to learn the weights in PROTO-CLIP. First, the support set is used to initialize the image memory $\mathbf{W}_{\text{image}}$ and the text memory \mathbf{W}_{text} . Second, the weights in the adapter network applied to the query images $g_{\mathbf{w}_1}(\cdot)$ need to be learned. Fig. 3 shows two designs of the adapter network, i.e., an MLP-based adapter as in [9] and a convolution-based adapter that we introduce. The convolution-based adapter has fewer weights to learn compared to the MLP-based one. We found that the two adapters have their own advantages on different datasets in our experiments. Finally, motivated by the CLIP-Adapter [9], we do not fine-tune the weights in the image encoder and text encoder by freezing these weights during training. In this way, we can reuse the weights of CLIP trained on a large

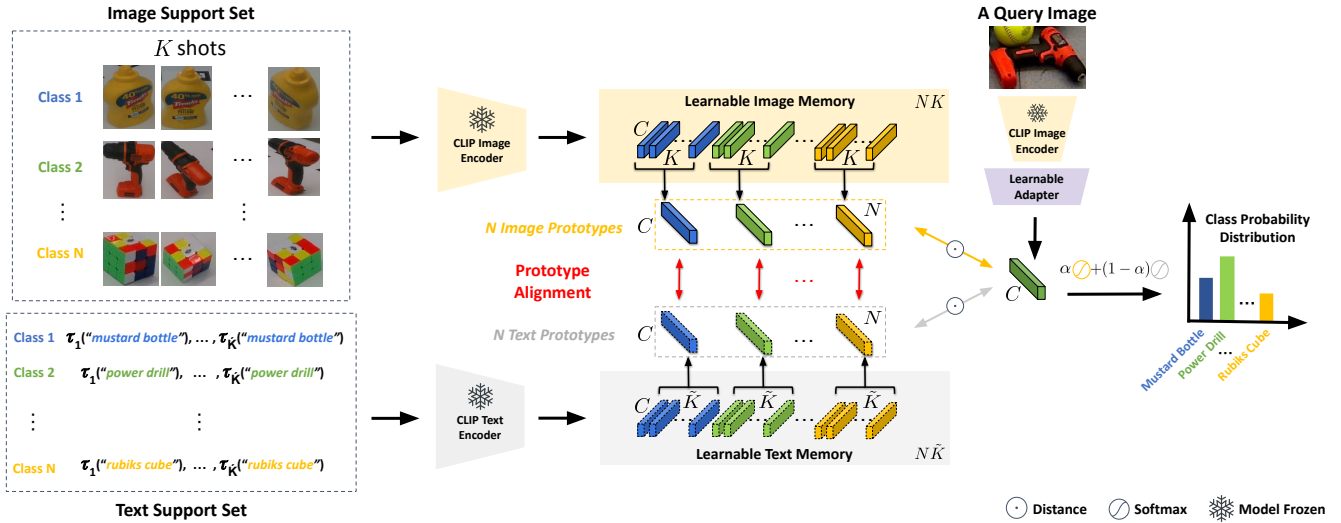


Fig. 2: Overview of our proposed Proto-CLIP model. The image memory, the text memory and the adapter network are learned. Given a class name, τ_i returns the i^{th} out of \tilde{K} predefined text prompts.

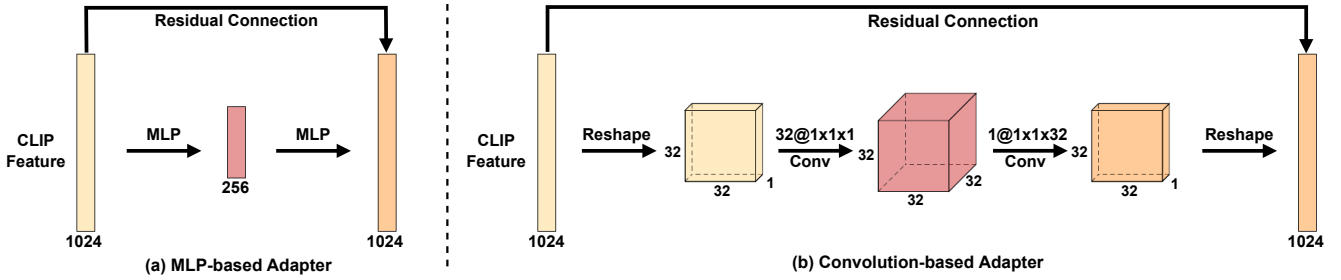


Fig. 3: Two designs of the adapters. (a) A Multi-layer perceptron-based adapter as in [9]. (b) A convolution-based adapter that we introduce. The feature dimension is for CLIP ResNet50 backbone.

number of image-text pairs and adapt the image embeddings and the text embeddings.

Loss Functions. The first loss function is the negative log-probability of the true label for a query image: $\mathcal{L}_1(\mathbf{W}_{\text{image}}, \mathbf{W}_{\text{text}}, \mathbf{w}_1) = -\log P(y^q = k | \mathbf{x}^q, \mathcal{S})$, where $P(y^q = k | \mathbf{x}^q, \mathcal{S})$ is defined in Eq. 1. Minimizing \mathcal{L}_1 learns the weights to classify the query images correctly. Second, we propose aligning the image prototypes and the text prototypes in training. Let $\{\mathbf{c}_1^x, \mathbf{c}_2^x, \dots, \mathbf{c}_N^x\}$ be the image prototypes computed from the image embeddings for the N classes and $\{\mathbf{c}_1^y, \mathbf{c}_2^y, \dots, \mathbf{c}_N^y\}$ be the corresponding text prototypes. We would like to learn the model weights such that \mathbf{c}_k^x is close to \mathbf{c}_k^y and far from other prototypes in the embedding space. We utilize the InfoNCE loss for contrastive learning [23]:

$$\mathcal{L}_2^k(\mathbf{c}_k^x, \{\mathbf{c}_{k'}^y\}_{k'=1}^N) = -\log \frac{\exp(\mathbf{c}_k^x \cdot \mathbf{c}_k^y)}{\sum_{k'=1}^N \exp(\mathbf{c}_k^x \cdot \mathbf{c}_{k'}^y)} \quad (6)$$

$$\mathcal{L}_3^k(\mathbf{c}_k^y, \{\mathbf{c}_{k'}^x\}_{k'=1}^N) = -\log \frac{\exp(\mathbf{c}_k^y \cdot \mathbf{c}_k^x)}{\sum_{k'=1}^N \exp(\mathbf{c}_k^y \cdot \mathbf{c}_{k'}^x)} \quad (7)$$

for $k = 1, \dots, N$, where \cdot indicates dot-product. Here, $\mathcal{L}_2^k(\mathbf{c}_k^x, \{\mathbf{c}_{k'}^y\}_{k'=1}^N)$ compares an image prototype \mathbf{c}_k^x with the text prototypes $\{\mathbf{c}_{k'}^y\}_{k'=1}^N$, while $\mathcal{L}_3^k(\mathbf{c}_k^y, \{\mathbf{c}_{k'}^x\}_{k'=1}^N)$ compares a text prototype \mathbf{c}_k^y with the image prototypes $\{\mathbf{c}_{k'}^x\}_{k'=1}^N$. In this way, we can align the image prototypes and the text prototypes for the N classes. This alignment can facilitate

classification, since the class conditional probabilities are computed using the image prototypes and the text prototypes as in Eqs. 2 and 3. The total loss function for training is:

$$\mathcal{L} = -\frac{1}{L} \sum_{j=1}^L \log P(y_j^q = k | \mathbf{x}_j^q, \mathcal{S}) + \frac{1}{N} \sum_{k=1}^N (\mathcal{L}_2^k(\mathbf{c}_k^x, \{\mathbf{c}_{k'}^y\}_{k'=1}^N) + \mathcal{L}_3^k(\mathbf{c}_k^y, \{\mathbf{c}_{k'}^x\}_{k'=1}^N)) \quad (8)$$

for a query set $Q = \{\mathbf{x}_j^q, y_j^q\}_{j=1}^L$. Following previous CLIP-based few-shot learning methods [8], [9], [10], the support set and the query set are the same during training in our experiments, i.e., $\mathcal{S} = Q$ meaning any of the support samples can act as a query sample during training.

IV. EXPERIMENTS

Datasets and Evaluation Metric. Following previous CLIP-based few-shot learning methods [8], [9], [10], we conduct experiments on the following datasets for evaluation: ImageNet [5], StanfordCars [25], UCF101 [26], Caltech101 [27], Flowers102 [28], SUN397 [29], DTD [30], EuroSAT [31], FGVCaircraft [32], OxfordPets [33], and Food101 [34]. In addition, we also include the FewSOL dataset [15] recently introduced for few-shot object recognition in robotic environments in order to improve object

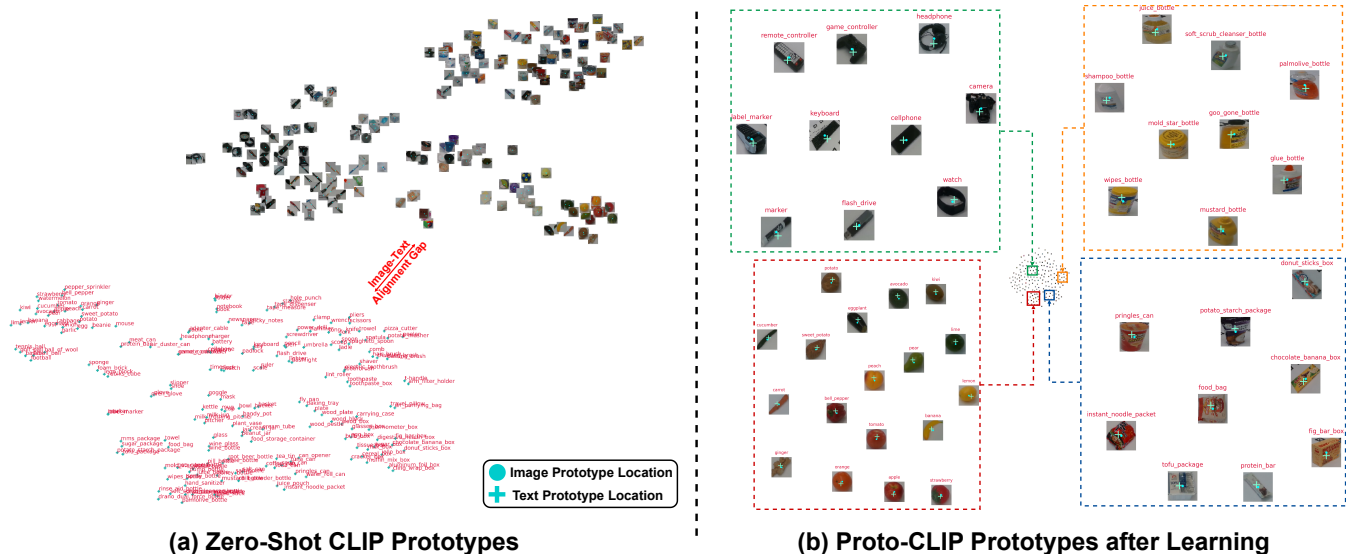


Fig. 4: Barnes-Hut t-SNE visualization [24] using the FewSOL dataset [15]. (a) Image and text prototypes from zero-shot CLIP, which are not aligned. (b) Aligned image and text prototypes from Proto-CLIP- F .

classification for robot manipulation tasks. In the N -way K -shot classification setting, K images for each class will be sampled from each dataset for training. A validation set of each dataset is reserved for hyper-parameter tuning, and a test set is used for evaluation. We evaluate using test set classification accuracy, as in related works.

Choosing the Hyper-parameters: α and β . From the experiments, we found that the two hyper-parameters α in Eq. 1 and β in Eq. 2 and Eq. 3 play a critical role in classification accuracy. Therefore, for each dataset, we conducted a grid search of the two parameters using the validation set. Then we finalize their values for all the runs in our experiments.

Proto-CLIP Variants. i) “Proto-CLIP”: we do not train the image memory and the text memory and do not use any adapter in Proto-CLIP (Fig. 2), we directly run inference using the pre-trained CLIP features. We term this variant the “training-free” version because it does not require training. This offers a convenient way to quickly test new datasets without the complexities of training, although it comes with the caveat of potential misalignment between visual and textual features. ii) “Proto-CLIP- F ”: we train the image memory and/or the text memory with the adapter. During training, for all the query images, we precompute their CLIP image features and directly use these stored features for training. This variant can be trained more quickly w.r.t. the following variant. Therefore, we use it for our ablation studies. iii) “Proto-CLIP- F - Q^T ”: During training, for each query image, we apply random data augmentation operations such as cropping and horizontal flip. Then we compute CLIP image features for the transformed query images during training.

A. Ablation Studies

Adapter Types and Learnable Text Memory. Since the 12 datasets have different characteristics, we found that varying adapter types and whether to learn the text memory or not

affect performance. Table III summarizes the result of this ablation study. Visual data plays a crucial role in image recognition when compared to textual information. Therefore, visual memory keys are consistently trained, regardless of the circumstances. The architectures of the MLP-based adapter and the convolution-based adapter are illustrated in Fig. 3. “2xConv” indicates using 2 convolution layers as shown in Fig. 3, while “3xConv” uses 3 convolution layers in the adapter where we add a $32@3 \times 3 \times 32$ convolution layer in the middle. By checking the best accuracy for each dataset, we can observe that there is no consensus on which adapter and trainable text memory setup to use among these datasets. Therefore, we select the best configuration on the adapter and learnable text memory for each dataset in the following experiments. Learning both image memory and text memory can help to yield aligned image-text prototypes. Fig. 4 visualizes the image-text prototypes in the FewSOL dataset [15] before and after training. For Proto-CLIP- F , unless specified otherwise, both the adapter and the visual memory keys are trained in all scenarios.

Loss functions. We have introduced three different loss functions in Sec. III: $\mathcal{L}_1, \mathcal{L}_2, \mathcal{L}_3$. We analyze the effects of these loss functions in Table IV. We can see that i) the \mathcal{L}_1 loss function is essential since it drives the classification of the query images; ii) Overall, both \mathcal{L}_2 and \mathcal{L}_3 loss functions for prototype alignment contribute to the performance, which verifies our motivation of aligning image and text prototypes for few-shot classification.

Backbones. Table V shows the results of using different backbone networks on the FewSOL dataset [15]. In general, better backbones can learn more powerful feature representations and consequently improve the classification accuracy. CLIP vision transformer backbones achieve better performance than CLIP ResNet backbones.

Shots. Table VI displays the results of using different numbers of shots on ImageNet [5] and FewSOL [15]. With

Dataset # classes	ImageNet 1000	FGVC 100	Pets 37	Cars 196	EuroSAT 10	Caltech101 100	SUN397 397	DTD 47	Flowers 102	Food101 101	UCF101 101	FewSOL 52
Zero-shot CLIP [1]	60.33	17.10	85.83	55.74	37.52	85.92	58.52	42.20	66.02	77.32	61.35	25.91
1 shot												
Linear-Probe CLIP [1]	22.07	12.89	30.14	24.64	51.00	70.62	32.80	29.59	58.07	30.13	41.43	-
CLIP-Adapter [9]	61.20	17.49	85.99	55.13	61.40	88.60	61.30	45.80	73.49	76.82	62.20	-
CoOp [8]	57.15	9.64	85.89	55.59	50.63	87.53	60.29	44.39	68.12	74.32	61.92	-
Tip [10]	60.70	19.05	86.10	57.54	54.38	87.18	61.30	46.22	73.12	77.42	62.60	27.30
Tip-F [10]	61.13	20.22	87.00	58.86	59.53	89.33	62.50	49.65	79.98	77.51	64.87	27.91
Proto-CLIP	60.31	19.59	86.10	57.29	55.53	87.99	60.81	46.04	76.98	77.36	63.15	27.09
Proto-CLIP-F	60.32	19.50	85.72	57.34	54.93	88.07	60.83	35.64	77.47	77.34	63.07	22.22
Proto-CLIP-F- Q^T	59.12	16.26	83.62	52.77	61.95	88.48	61.43	32.27	68.53	75.16	62.44	21.65
2 shots												
Linear-Probe CLIP [1]	31.95	17.85	43.47	36.53	61.58	78.72	44.44	39.48	73.35	42.79	53.55	-
CLIP-Adapter [9]	61.52	20.10	86.73	58.74	63.90	89.37	63.29	51.48	81.61	77.22	67.12	-
CoOp [8]	57.81	18.68	82.64	58.28	61.50	87.93	59.48	45.15	77.51	72.49	64.09	-
Tip [10]	60.96	21.21	87.03	57.93	61.68	88.44	62.70	49.47	79.13	77.52	64.74	26.22
Tip-F [10]	61.69	23.19	87.03	61.50	66.15	89.74	63.64	53.72	82.30	77.81	66.43	27.43
Proto-CLIP	60.64	22.14	87.38	60.01	64.89	89.05	63.12	51.06	83.39	77.34	67.46	28.35
Proto-CLIP-F	60.64	22.14	87.38	60.04	64.86	89.09	63.20	49.88	83.52	77.34	67.49	26.17
Proto-CLIP-F- Q^T	60.48	20.01	85.28	60.02	63.59	89.49	65.46	45.69	81.20	76.15	68.83	25.91
4 shots												
Linear-Probe CLIP [1]	41.29	23.57	56.35	48.42	68.27	84.34	54.59	50.06	84.80	55.15	62.23	-
CLIP-Adapter [9]	61.84	22.59	87.46	62.45	73.38	89.98	65.96	56.86	87.17	77.92	69.05	-
CoOp [8]	59.99	21.87	86.70	62.62	70.18	89.55	63.47	53.49	86.20	73.33	67.03	-
Tip [10]	60.98	22.41	86.45	61.45	65.32	89.39	64.15	53.96	83.80	77.54	66.46	28.70
Tip-F [10]	62.52	25.80	87.54	64.57	74.12	90.56	66.21	57.39	88.83	78.24	70.55	29.13
Proto-CLIP	61.30	23.25	87.19	63.33	68.67	89.57	65.51	55.91	88.23	77.58	69.50	29.13
Proto-CLIP-F	61.30	23.31	86.95	63.34	68.52	89.62	65.57	57.21	88.27	77.58	69.55	27.09
Proto-CLIP-F- Q^T	61.80	27.63	87.11	66.24	80.64	91.81	68.09	56.86	89.85	76.94	70.16	30.30
8 shots												
Linear-Probe CLIP [1]	49.55	29.55	65.94	60.82	76.93	87.78	62.17	56.56	92.00	63.82	69.64	-
CLIP-Adapter [9]	62.68	26.25	87.65	67.89	77.93	91.40	67.50	61.00	91.72	78.04	73.30	-
CoOp [8]	61.56	26.13	85.32	68.43	76.73	90.21	65.52	59.97	91.18	71.82	71.94	-
Tip [10]	61.45	25.59	87.03	62.93	67.95	89.83	65.62	58.63	87.98	77.76	68.68	29.22
Tip-F [10]	64.00	30.21	88.09	69.25	77.93	91.44	68.87	62.71	91.51	78.64	74.25	32.43
Proto-CLIP	62.12	27.63	88.04	64.93	69.42	90.22	67.37	59.34	92.08	77.90	71.08	29.83
Proto-CLIP-F	63.92	31.32	88.55	70.35	78.94	92.54	69.59	62.35	93.79	78.29	74.81	33.26
Proto-CLIP-F- Q^T	64.03	35.82	87.46	71.50	81.89	92.62	70.02	64.01	94.28	78.61	75.34	32.70
16 shots												
Linear-Probe CLIP [1]	55.87	36.39	76.42	70.08	82.76	90.63	67.15	63.97	94.95	70.17	73.72	-
CLIP-Adapter [9]	63.59	32.10	87.84	74.01	84.43	92.49	69.55	65.96	93.90	78.25	76.76	-
CoOp [8]	62.95	31.26	87.01	73.36	83.53	91.83	69.26	63.58	94.51	74.67	75.71	-
Tip [10]	62.02	29.76	88.14	66.77	70.54	90.18	66.85	60.93	89.89	77.83	70.58	28.87
Tip-F [10]	65.51	35.55	89.70	75.74	84.54	92.86	71.47	66.55	94.80	79.43	78.03	34.04
Proto-CLIP	62.77	29.67	88.61	68.11	72.95	91.08	68.09	61.64	92.94	78.11	73.35	29.96
Proto-CLIP-F	65.75	37.56	89.62	75.25	83.53	93.43	71.94	68.56	95.78	79.09	77.50	35.22
Proto-CLIP-F- Q^T	65.91	40.65	89.34	76.76	86.59	93.59	72.19	68.50	96.35	79.34	78.11	34.70

TABLE II: Few-shot classification results of various CLIP based few shot learning methods on different datasets across various shots using the CLIP ResNet50 backbone.

Adapter	Train-Text-Memory	ImageNet	FGVC	Pets	Cars	EuroSAT	Caltech101	SUN397	DTD	Flowers	Food101	UCF101	FewSOL
MLP	✗	61.06	35.31	85.61	72.19	83.47	92.58	68.54	63.89	95.01	74.05	76.16	28.65
MLP	✓	61.06	37.56	85.72	73.61	83.53	92.13	69.71	63.89	96.06	74.05	76.16	32.87
2xConv	✗	65.75	34.38	89.62	75.25	81.85	93.40	71.94	67.85	94.76	79.09	77.50	27.13
2xConv	✓	58.60	35.82	89.21	74.34	81.78	93.02	69.79	67.32	95.82	78.06	76.37	27.13
3xConv	✗	65.37	34.41	88.74	<u>75.25</u>	82.21	93.43	71.63	67.67	94.40	79.11	<u>77.50</u>	29.78
3xConv	✓	59.63	36.15	87.93	72.68	81.57	92.74	68.64	68.56	95.78	78.61	77.03	35.22

TABLE III: Results of ablation study of various query adapter types and textual memory bank training using the CLIP ResNet50 backbone with $K = 16$ on Proto-CLIP-F. In case of a tie, the underlined setup was selected randomly.

Loss	ImageNet	FGVC	Pets	Cars	EuroSAT	Caltech101	SUN397	DTD	Flowers	Food101	UCF101	FewSOL
\mathcal{L}_1	62.67	20.34	73.21	73.77	78.98	92.25	68.34	66.49	96.14	77.39	76.66	34.57
\mathcal{L}_2	62.29	4.71	0.00	0.00	38.95	0.28	66.93	67.38	10.31	77.71	57.41	32.70
\mathcal{L}_3	62.27	4.14	0.00	0.00	38.09	0.24	64.86	67.38	10.27	77.69	57.55	20.22
$\mathcal{L}_1 + \mathcal{L}_2$	65.39	36.24	88.58	75.39	82.78	93.71	71.65	68.09	96.06	78.69	77.29	33.48
$\mathcal{L}_2 + \mathcal{L}_3$	62.33	3.87	0.00	0.00	36.86	0.24	64.84	68.32	8.20	77.35	57.52	19.61
$\mathcal{L}_1 + \mathcal{L}_3$	65.43	36.84	88.58	75.51	82.84	93.35	71.44	68.32	96.14	78.80	77.53	33.43
$\mathcal{L}_1 + \mathcal{L}_2 + \mathcal{L}_3$	65.75	37.56	89.62	75.25	83.53	93.43	71.94	68.56	96.06	79.09	77.50	35.22

TABLE IV: Ablation study of various Loss functions using the CLIP ResNet50 backbone and $K = 16$. The best performing model architectures for each dataset from Table III are used here.

more shots for training, the classification accuracy is improved accordingly. The choice of $K=16$ for our experiments aligns with the prevalent practice in the field of vision-language few-shot learning. This specific value has been widely adopted, as

evidenced in various scholarly works such as [9], [8], [10]. Moreover, given our specific emphasis on the few-shot context, it appeared prudent to exercise caution when surpassing a particular threshold, specifically 16 in our case. As a result,

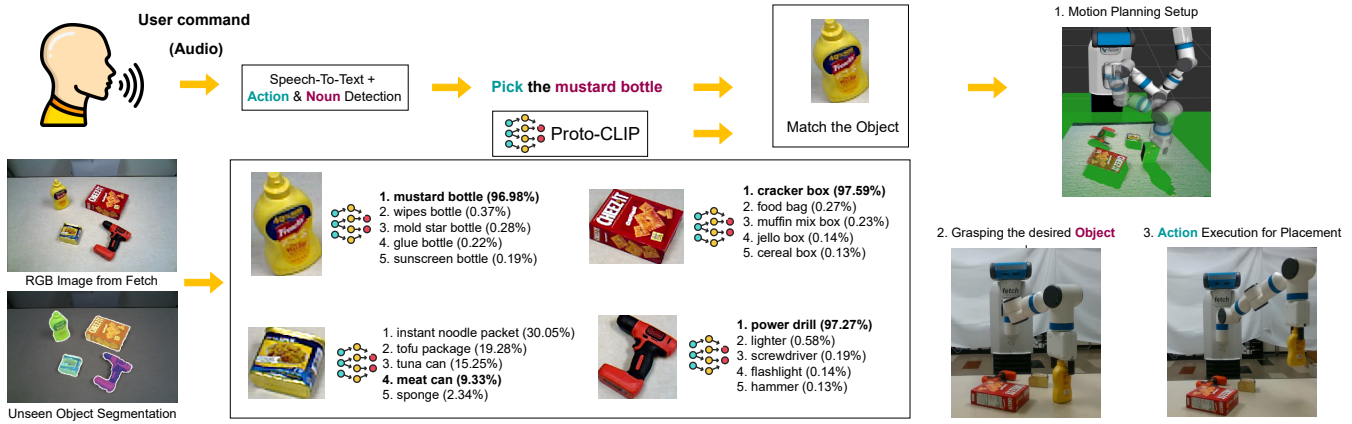


Fig. 5: Results for the real world setup with top-5 predictions from the Proto-CLIP- F (ViT-L/14) model trained on FEWSOL-198 [15]. The Speech-To-Text is performed via Whisper [35].

Model	Adapter	TTM	Backbone				
			RN50	RN101	ViT-B/16	ViT-B/32	ViT-L/14
Zero-Shot-CLIP [1]	-	-	25.91	32.96	40.70	41.87	54.57
Tip [10]	-	-	29.74	37.43	47.00	41.48	56.78
Tip-F [10]	-	-	32.52	41.43	50.17	45.48	60.17
Proto-CLIP- F	MLP	✗	33.48	39.04	47.96	41.91	58.65
Proto-CLIP- F	MLP	✓	34.83	40.74	47.43	42.13	58.91
Proto-CLIP- F	2xConv	✗	35.04	41.04	50.83	46.52	63.74
Proto-CLIP- F	2xConv	✓	35.04	42.52	49.26	43.43	61.61
Proto-CLIP- F	3xConv	✗	34.13	42.83	51.91	46.87	62.35
Proto-CLIP- F	3xConv	✓	35.22	44.09	50.39	46.57	60.39

TABLE V: Backbone ablation study. Dataset=FEWSOL-52 [15]. $K = 16$. TTM=‘Train-Text-Memory’.

Dataset	Method	1	2	4	8	16	32	64
ImageNet [5]	Tip [10]	60.70	60.96	60.98	61.45	62.01	62.51	62.88
	Proto-CLIP	60.31	60.64	61.30	62.12	62.77	62.98	63.23
	Tip-F [10]	61.13	61.69	62.52	64.00	65.51	66.58	67.96
FEWSOL-52 [15]	Proto-CLIP- F	60.32	60.64	61.30	63.92	65.75	66.47	65.36
	Proto-CLIP- F - Q^T	59.12	60.48	61.80	64.03	65.91	66.71	66.90
	Tip [10]	27.30	26.22	28.70	29.22	28.87	✗	✗
FEWSOL-52 [15]	Proto-CLIP	27.09	28.35	29.13	29.83	29.96	✗	✗
	Tip-F [10]	27.91	27.43	29.13	32.43	34.04	✗	✗
	Proto-CLIP- F	22.22	26.17	27.09	33.26	35.22	✗	✗
	Proto-CLIP- F - Q^T	21.65	25.91	30.30	32.70	34.70	✗	✗

TABLE VI: Shots ablation results. Backbone=‘CLIP ResNet50’.

we embarked on an ablation study involving the ImageNet [5] dataset. This particular dataset holds the largest number of classes (1000) and thus provided a suitable platform for investigating shots values beyond 16, such as 32 and 64. Despite our intention to explore 128 shots, our experimental hardware’s memory limitations prohibited us from pursuing this avenue. Additionally, FEWSOL is valuable for few-shot object learning, especially in robotics. We capped shots at 16 for FEWSOL as average number of samples per class in FEWSOL hovers around 15. Consequently, we conjectured that going beyond might yield diminishing learning returns. These insights are detailed in Table VI.

B. Comparison with Other Methods

Table II shows the performance of Proto-CLIP compared to the state-of-the-art methods using CLIP for few-shot learning in the literature: Linear-Probe CLIP [1], CoOp [8], CLIP-Adapter [9] and Tip-Adapter [8]. We follow these methods and use CLIP’s ResNet50 backbone for this comparison. The fine-tuned variant of Tip-Adapter ‘‘Tip-F’’ is the most competitive method compared to ours. The performance of Proto-CLIP on very few shots, i.e., 1 shot and 2 shots is inferior compared to Tip-F. When the number of shots increases to 4, 8 and

16, the fine-tuned variants of Proto-CLIP outperform Tip-F. The enhanced performance of our proposed Proto-CLIP method can be attributed to its reliance on robust image and textual prototypes, which subsequently leads to improved classification accuracy. Therefore, our model benefits from more than 4 shots, while it is not as good as Tip-F when using 1 shot and 2 shots. Proto-CLIP- F - Q^T performs better than Proto-CLIP- F on most datasets by using the data augmentation of query images during training².

C. Real World Experiments

As an application, we have built a robotic system to verify the effectiveness of Proto-CLIP for object recognition in the real world. Fig. 5 illustrates our pipeline for the system. It takes human instruction in the form of voice commands as input such as ‘‘pick something’’ or ‘‘grasp something’’. The system first applies Automatic Speech Recognition (ASR) to convert voice input to text using OpenAI Whisper [35]. Then the system grounds the noun in the human instruction into a target object observed from an input image. This is achieved by joint object segmentation and classification. We utilize unseen object instance segmentation [36] to segment objects in cluttered scenes and then classify each segmented object with Proto-CLIP. By matching the noun with the class labels, the system can ground the target in the image. Once the target object is recognized, we use Contact-GraspNet [37] for grasp planning and MoveIt motion planning toolbox [38] to pick and place the target².

V. CONCLUSIONS

We have introduced a novel method for few-shot learning based on the CLIP [1] vision-language model. Our method learns image prototypes and text prototypes from few-shot training examples and aligns the corresponding image-text prototypes for classification. The model is equipped with learnable image memory and text memory for support images and a learnable adapter for query images. Compared to previous CLIP-based few-shot learning methods, our method

²For more details, please see the supplementary material on the project page.

is flexible in configuring these learnable components, resulting in powerful learned models. Good feature representation is the key in few-shot learning. Future work includes how to further improve feature representation learning compared to CLIP models. One idea is to adapt more powerful vision-language models such as GPT variants. The FewSOL [15] dataset also provides multiview and depth information about objects. Exploring this 3D information in few-shot object recognition is also a promising direction.

ACKNOWLEDGMENT

This work was supported in part by the DARPA Perceptually-enabled Task Guidance (PTG) Program under contract number HR00112220005.

REFERENCES

- [1] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *International Conference on Machine Learning*, 2021, pp. 8748–8763.
- [2] Y. Wang, Q. Yao, J. T. Kwok, and L. M. Ni, "Generalizing from a few examples: A survey on few-shot learning," *ACM computing surveys (CSUR)*, vol. 53, no. 3, pp. 1–34, 2020.
- [3] B. Calli, A. Walsman, A. Singh, S. Srinivasa, P. Abbeel, and A. M. Dollar, "Benchmarking in manipulation research: The YCB object and model set and benchmarking protocols," *arXiv preprint arXiv:1502.03143*, 2015.
- [4] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *European Conference on Computer Vision (ECCV)*. Springer, 2014, pp. 740–755.
- [5] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009, pp. 248–255.
- [6] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li *et al.*, "Visual genome: Connecting language and vision using crowdsourced dense image annotations," *International Journal of Computer Vision (IJCV)*, vol. 123, no. 1, pp. 32–73, 2017.
- [7] Y. Tian, Y. Wang, D. Krishnan, J. B. Tenenbaum, and P. Isola, "Rethinking few-shot image classification: a good embedding is all you need?" in *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16*. Springer, 2020, pp. 266–282.
- [8] K. Zhou, J. Yang, C. C. Loy, and Z. Liu, "Learning to prompt for vision-language models," *International Journal of Computer Vision*, vol. 130, no. 9, pp. 2337–2348, 2022.
- [9] P. Gao, S. Geng, R. Zhang, T. Ma, R. Fang, Y. Zhang, H. Li, and Y. Qiao, "Clip-adapter: Better vision-language models with feature adapters," *arXiv 2110.04544*, 2021.
- [10] R. Zhang, Z. Wei, R. Fang, P. Gao, K. Li, J. Dai, Y. Qiao, and H. Li, "Tip-adapter: Training-free adaption of clip for few-shot classification," *arXiv preprint arXiv:2207.09519*, 2022.
- [11] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *International Conference on Machine Learning (ICML)*, 2017, pp. 1126–1135.
- [12] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 30, 2017.
- [13] E. Triantafyllou, T. Zhu, V. Dumoulin, P. Lamblin, U. Evci, K. Xu, R. Goroshin, C. Gelada, K. Swersky, P.-A. Manzagol *et al.*, "Meta-dataset: A dataset of datasets for learning to learn from few examples," *arXiv preprint arXiv:1903.03096*, 2019.
- [14] C. Doersch, A. Gupta, and A. Zisserman, "Crosstransformers: spatially-aware few-shot transfer," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, pp. 21 981–21 993, 2020.
- [15] J. J. P. Y.-W. Chao, and Y. Xiang, "Fewsol: A dataset for few-shot object learning in robotic environments," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 9140–9146.
- [16] V. Udandarao, A. Gupta, and S. Albanie, "Sus-x: Training-free name-only transfer of vision-language models," *arXiv preprint arXiv:2211.16198*, 2022.
- [17] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 10 684–10 695.
- [18] S. Gidaris and N. Komodakis, "Dynamic few-shot visual learning without forgetting," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 4367–4375.
- [19] H. Qi, M. Brown, and D. G. Lowe, "Low-shot learning with imprinted weights," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 5822–5830.
- [20] W.-Y. Chen, Y.-C. Liu, Z. Kira, Y.-C. F. Wang, and J.-B. Huang, "A closer look at few-shot classification," *arXiv preprint arXiv:1904.04232*, 2019.
- [21] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra *et al.*, "Matching networks for one shot learning," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 29, 2016.
- [22] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. Torr, and T. M. Hospedales, "Learning to compare: Relation network for few-shot learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 1199–1208.
- [23] A. v. d. Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *arXiv preprint arXiv:1807.03748*, 2018.
- [24] L. Van Der Maaten, "Accelerating t-sne using tree-based algorithms," *The journal of machine learning research*, vol. 15, no. 1, pp. 3221–3245, 2014.
- [25] J. Krause, M. Stark, J. Deng, and L. Fei-Fei, "3d object representations for fine-grained categorization," in *Proceedings of the IEEE international conference on computer vision workshops*, 2013, pp. 554–561.
- [26] K. Soomro, A. R. Zamir, and M. Shah, "Ucf101: A dataset of 101 human actions classes from videos in the wild," *arXiv preprint arXiv:1212.0402*, 2012.
- [27] L. Fei-Fei, R. Fergus, and P. Perona, "Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories," in *2004 conference on computer vision and pattern recognition workshop*. IEEE, 2004, pp. 178–178.
- [28] M.-E. Nilsback and A. Zisserman, "Automated flower classification over a large number of classes," in *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*. IEEE, 2008, pp. 722–729.
- [29] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba, "Sun database: Large-scale scene recognition from abbey to zoo," in *2010 IEEE computer society conference on computer vision and pattern recognition*. IEEE, 2010, pp. 3485–3492.
- [30] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, and A. Vedaldi, "Describing textures in the wild," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 3606–3613.
- [31] P. Helber, B. Bischke, A. Dengel, and D. 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*, vol. 12, no. 7, pp. 2217–2226, 2019.
- [32] S. Maji, E. Rahtu, J. Kannala, M. Blaschko, and A. Vedaldi, "Fine-grained visual classification of aircraft," *arXiv preprint arXiv:1306.5151*, 2013.
- [33] O. M. Parkhi, A. Vedaldi, A. Zisserman, and C. Jawahar, "Cats and dogs," in *2012 IEEE conference on computer vision and pattern recognition*. IEEE, 2012, pp. 3498–3505.
- [34] L. Bossard, M. Guillaumin, and L. Van Gool, "Food-101—mining discriminative components with random forests," in *Computer Vision—ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part VI 13*. Springer, 2014, pp. 446–461.
- [35] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," *arXiv preprint arXiv:2212.04356*, 2022.
- [36] Y. Lu, Y. Chen, N. Ruozzi, and Y. Xiang, "Mean shift mask transformer for unseen object instance segmentation," *arXiv preprint arXiv:2211.11679*, 2022.
- [37] M. Sundermeyer, A. Mousavian, R. Triebel, and D. Fox, "Contact-graspnet: Efficient 6-dof grasp generation in cluttered scenes," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 13 438–13 444.
- [38] S. Chitta, I. Sucas, and S. Cousins, "Moveit! [ros topics]," *IEEE Robotics & Automation Magazine*, vol. 19, no. 1, pp. 18–19, 2012.