Rectifying the Shortcut Learning of Background: Shared Object Concentration for Few-Shot Image Recognition

Anonymous Author(s)
Affiliation
Address
email

Abstract

Few-Shot image classification aims to utilize pretrained knowledge learned from a large-scale dataset to tackle a series of downstream classification tasks. Typically, each task involves only few training examples from brand-new categories. This requires the pretraining models to focus on well-generalizable knowledge, but ignore domain-specific information. In this paper, we observe that image background serves as a source of domain-specific knowledge, which is a shortcut for models to learn in the source dataset, but is harmful when adapting to brand-new classes. To prevent the model from learning this shortcut knowledge, we propose COSOC, a novel Few-Shot Learning framework, to automatically figure out foreground objects at both pretraining and evaluation stage. COSOC is a two-stage algorithm motivated by the observation that foreground objects from different images within the same class share more similar patterns than backgrounds. At the pretraining stage, for each class, we cluster contrastive-pretrained features of randomly cropped image patches, such that crops containing only foreground objects can be identified by a single cluster. We then force the pretraining model to focus on found foreground objects by a fusion sampling strategy; at the evaluation stage, among images in each training class of any few-shot task, we seek for shared contents and filter out background. The recognized foreground objects of each class are used to match foreground of testing images. Extensive experiments tailored to inductive FSL tasks on two benchmarks demonstrate the state-of-the-art performance of our method. Our code is available in the supplementary material.

1 Introduction

Through observing a few samples at a glance, humans can accurately identify brand-new objects. This advantage comes from years of experience accumulated by the human vision system. Inspired by such learning capabilities, Few-Shot Learning (FSL) is developed to tackle the problem of learning from limited data [21, 48]. A typical FSL framework has two separated stages, the pretraining stage and the evaluation stage. At the pretraining stage, FSL models absorb knowledge from a large-scale dataset; at the evaluation stage, the learned knowledge is leveraged to solve a series of downstream classification tasks, each of which contains very few training images from *brand-new categories*. The category difference leads to an overwhelming distribution gap between these two stages [58, 6]. This challenge places high demands for FSL to learn well-generalizable pretrained knowledge.

In this paper, we find that image background serves as one source of such domain-specific knowledge in FSL. Image background is disturbing information that usually does not determine the category of the image. But as pointed out in [52], there is spurious correlation between background and category of image (*e.g.*, birds usually stand on branches, and shells often lie on the beaches), which serves as

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

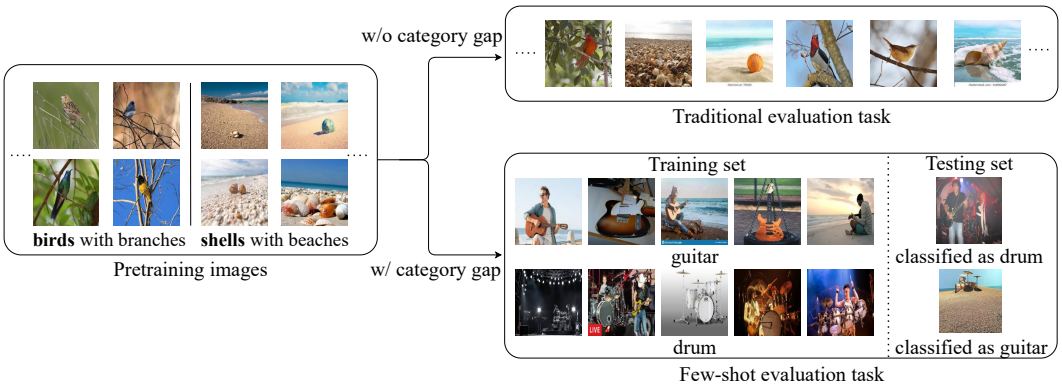

Figure 1: An illustrative example that demonstrates why background information is useful for regular classification but is harmful for Few-Shot Learning.

shortcut knowledge for models to learn. As illustrated in Fig. 1, background knowledge is useful only when no category gap (*i.e.*, the classes remain unchanged) exists between pretraining and evaluation datasets. However, in FSL, the correlation cannot generalize to brand-new classes and will probably mislead the predictions. For more thorough empirical study, please refer to Sec. 3.

A very natural solution is to force the model to concentrate on foreground objects at both two stages, but this is not easy since we do not have any prior knowledge of the entity and position of the foreground object in images. Our key insight is that foreground objects from images of the same class share more common patterns compared to background. Motivated by this observation, we customize algorithms separately for the pretraining and evaluation stages.

At the pretraining stage, we seek foreground objects by clustering random crops of images within each class. In this way, crops containing only foreground objects are more likely to form a single cluster, thus can be identified easily. However, the premise is that we have a feature extractor that can map foreground objects into compact regions in the feature space. To this end, we adopt the contrastive pretrained feature extractor [14, 62], which is revealed to be good at identifying foreground objects as shown in Sec. 3.2. After figuring out clusters that contain crops with foreground objects only, we define the foreground score of each image crop as the feature distance to its closest cluster center. Crops with the highest foreground score from each image are picked up and treated as potential foreground objects of the image. We call this framework **C**lustering-based **O**bject **S**eeker (COS). During pretraining, foreground objects obtained by COS are used to replace the original image with a probability depending on the foreground score. In this way, we aim to maximally suppress shortcut learning of background and improve the transferability of models.

At the few-shot evaluation stage, we propose **S**hared **O**bject **C**oncentrator (SOC) to grasp the shared information (i.e., foreground) among a few training images from each class. This is achieved by repeatedly seeking one part for each image such that the sum of pair-wise distance between these parts is minimized. The average features of obtained parts are regarded as representations of foreground objects and are used to locate foreground objects of testing images by feature matching. In this way, we remove irrelevant background information, such that reliable predictions can be made via comparisons between foreground objects of testing and training images in the few-shot task.

In summary, our contributions can be summarized as follows. *i*) By conducting empirical studies on image foreground and background in FSL, we reveal that image background serves as a source of shortcut knowledge which harms the evaluation performance. *ii*) To solve this problem, we propose COSOC, a two-stage framework combining COS and SOC, which can draw the model's attention to image foreground at both pretraining and evaluation stages. *iii*) Extensive experiments for non-transductive FSL tasks on two benchmarks have demonstrated the state-of-the-art performance of our method.

2 Related Works

Few-shot Image Classification. Few-shot learning [49] focuses on recognizing samples from new classes with only scarce labeled examples. This is a challenging task because of the risk of overfitting.

Plenty of previous works tackled this problem in meta-learning framework [15, 46], where a model learns experience about how to solve few-shot learning tasks by tackling pseudo few-shot classification tasks constructed from the pretraining dataset. Existing methods that ultilize meta-learning can be generally divided into three groups: (1) Optimization-based methods learn the experience of how to optimize the model given few training samples. This kind of methods either meta-learn a good model initialization point [8, 41, 36, 63, 17] or meta-learn the whole optimization process [37, 54, 30, 24] or meta-learn both [2, 33]. (2) Hallucination-based methods [13, 50, 42, 61, 22, 5, 23, 34] learn to augment similar training samples in few-shot tasks, thus can greatly alleviate the low-shot problem. (3) Metric-based methods [48, 44, 45, 16, 59, 57, 55, 53, 6] learn to map images into a metric feature space and classify testing images by computing feature distances to training images. Among them, several recent works [16, 59, 53, 6] sought correspondence between images either by attention or meta-filter, in order to obtain a more reasonable similarity measure. Similar with our SOC algorithm in Sec. 4.2, these methods can also align foreground objects in pairs of images at the evaluation stage. **However, such pair-wise alignment can be easily misled by similar background, leading to unreliable similarity measures between images**. Instead of pair-wise matching, SOC seeks for shared contents among all images within the same class, thus can reliably remove background.

Contrastive learning. Recent success on contrastive learning of visual representations has greatly promoted the development of unsupervised learning [4, 14, 12, 3]. The promising performance of contrastive learning relies on the instance-level discrimination loss which maximizes agreement between transformed views of the same image and minimizes agreement between transformed views of different images. Recently there have been some attempts [27, 9, 6, 29, 32] at integrating contrastive learning into the framework of FSL. Although achieving good results, these works don't have an in-depth understanding of why contrastiveness is useful in FSL. To this end, one target of our work is to reveal the advantages of contrastive learning over traditional supervised FSL models in identifying core objects of images and then show how to use this to improve the performance of FSL.

3 Empirical Investigation

Problem Definition. Few-shot learning consists of a pretraining dataset $\mathcal{D}_B$ and an evaluation set $\mathcal{D}_v$ which share no overlapping classes. $\mathcal{D}_B$ contains a large amount of labeled data and is usually used at first to pretrain a backbone network $f_\theta(\cdot)$. After pretraining, a set of N-way K-shot classification tasks $\mathcal{T} = \{(\mathcal{S}_\tau, \mathcal{Q}_\tau)\}_{\tau=1}^{N_\mathcal{T}}$ are constructed, each by first sampling N classes in D_v and then sampling K and M images from each class to constitute $\mathcal{S}_\tau$ and $\mathcal{Q}_\tau$, respectively. In each task τ, given the learned backbone $f_\theta(\cdot)$ and a small training set $\mathcal{S}_\tau = \{(x^\tau_{k,n}, y^\tau_{k,n})\}_{k,n=1}^{K,N}$ consisting of K images $x^\tau_{k,n}$ and corresponding labels $y^\tau_{k,n}$ from each of N classes, a few-shot classification algorithm is designed to classify MN images from the test set $\mathcal{Q}_\tau = \{(x^\tau_{mn})\}_{m,n=1}^{M,N}$.

Preparation. To investigate the influence of background and foreground in few-shot learning, we create a subset $\mathcal{D}_{\text{new}} = (\mathcal{D}_B, \mathcal{D}_v)$ of *mini*ImageNet [48] and crop each image manually in $\mathcal{D}_{\text{new}}$ so that backgrounds are removed at all and only the core object for recognition is remained. We denote the uncropped version of the subset as $(\mathcal{D}_B\text{-Ori}, \mathcal{D}_v\text{-Ori})$, and the cropped version as $(\mathcal{D}_B\text{-FG}, \mathcal{D}_v\text{-FG})$. We carry out experiments on two well-known FSL baselines: Cosine Classifier (CC) [10] and Prototypical Networks (PN) [44]. The details of constructing $\mathcal{D}_{\text{new}}$ and introductions of CC and PN can be found in Appendix A.

3.1 The Role of Foreground and Background in Few-Shot Image Classification

Fig. 2(a) shows the average of 5-way 5-shot classification accuracy obtained by pretraining CC and PN on $\mathcal{D}_B$-Ori and $\mathcal{D}_B$-FG, and evaluating on $\mathcal{D}_v$-Ori and $\mathcal{D}_v$-FG, respectively. Additional 5-way 1-shot experiments can be found in Appendix F.

Category gap disables generalization of background knowledge. It can be first noticed that, under any condition, the performance is consistently and significantly improved if background is removed at the evaluation stage (switch from $\mathcal{D}_v$-Ori to $\mathcal{D}_v$-FG). The result implies that background information at the evaluation stage in FSL is disturbing information. This is the opposite of that reported in [52] which shows removing background harms the performance of traditional classification task, where no category gap exists between pretraining and evaluation. Thus we can infer that the class/distribution gap in FSL disables generalization of background knowledge and degrades performance.

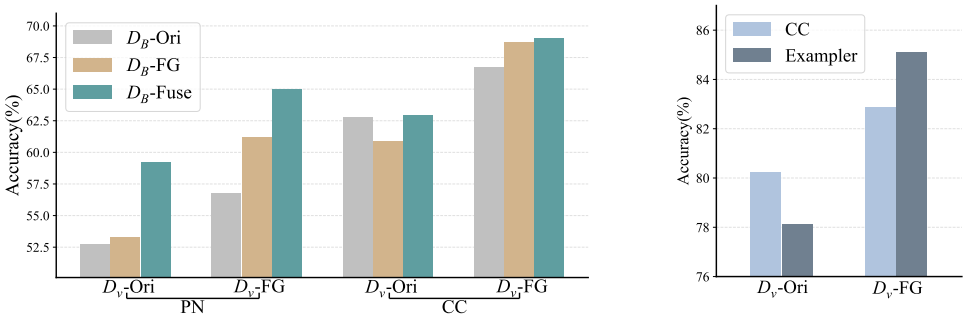

Figure 2: 5-way 5-shot FSL performance under different variants of pretraining and evaluation datasets which are described in Sec. 3. (a) Empirical exploration of image foreground and background in FSL using two models: PN and CC. (b) Comparison between CC and Exampler pretrained on full *mini*ImageNet and evaluated on $\mathcal{D}_v$-Ori and $\mathcal{D}_v$-FG.

Removing background at pretraining stage prevents shortcut learning. When only foreground is given during at evaluation stage ($\mathcal{D}_v$-FG), the models pretrained with only foreground ($\mathcal{D}_B$-FG) perform much better than that pretrained with the original image ($\mathcal{D}_B$-Ori). This indicates that models pretrained with original image do not pay enough attention to the foreground object which really matters for classification. Background information at the pretraining stage serves as a shortcut for the model to learn and cannot generalize to brand-new classes. In contrast, models pretrained with only foreground "learn to compare" different objects, which is the desirable ability for reliable generalization to downstream few-shot learning with out-of-domain classes.

Pretraining with background helps models to handle complex scenes. When evaluating on $\mathcal{D}_v$-Ori, the models pretrained with original dataset $\mathcal{D}_B$-Ori are slightly better than that with foreground dataset $\mathcal{D}_B$-FG. We attribute this to a kind of domain shift: models trained with $\mathcal{D}_B$-FG never meet images with complex background and do not know how to handle with it. In Appendix D.1 we show evaluation accuracy of each class under the above two pretraining situations which further verify the assertion. Note that since we apply random crop augmentation during pretraining, the same domain shift does not exist if the model is pretrained on $\mathcal{D}_B$-Ori and evaluated on $\mathcal{D}_v$-FG instead.

Simple fusion sampling combines advantages of both sides. We want to cut off shortcut learning of background while maintaining adaptability of model to complex scenes. A simple way is to mix the data sampling process: given an image as input, choose its foreground version with probability p, and its original version with probability $1 - p$. p is simply set as 0.5. We denote the dataset using this sampling strategy as $\mathcal{D}_B$-Fuse. From Fig. 2(a), we can observe that models trained by this way indeed combine advantages of both sides: achieve relatively good performance on both $\mathcal{D}_v$-Ori and $\mathcal{D}_v$-FG. In Appendix C, we compare the training curves of PN models pretrained on three versions of datasets to further investigate the effectiveness of fusion sampling.

The analysis above gives us new insights about how to improve few-shot learning further: (1) During pretraining, fusion sampling is required for enhancing the transferability of pretrained models. (2) Because background information disturbs during evaluation, it is needed to figure out where the foreground object is and then utilize only these parts for evaluation. Therefore, a reliable foreground object identification mechanism is required at both pretraining (for fusion sampling) and evaluation stages.

3.2 Contrastive Learning is Good at Identifying Objects

In this subsection, we reveal the potential of contrastive learning in identifying foreground objects without the interference of background, which will be utilized in Sec. 4.1. Given one transformed view of an image, contrastive learning tends to identify another transformed view of the same image in thousands of views of different images. We give a comprehensive introduction of contrastive learning in Appendix B. The two augmented views of the same image always cover the same object, but probably with different parts, sizes and colors. To discriminate two augmented patches from

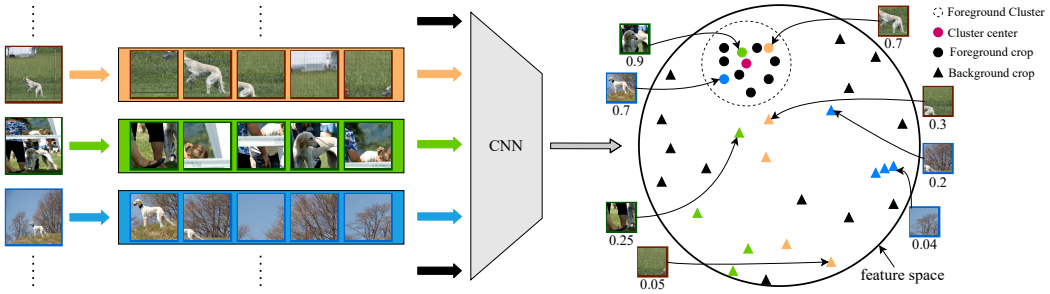

Figure 3: Simplified schematic illustration of COS algorithm. We show how we obtain foreground objects of three exemplified images. The number under each crop denotes its foreground score.

thousands of other image patches, the model has to learn how to identify the key discriminative information of the object under varying environment. In this manner, contrastive learning explicitly models semantic relations among crops of images, thereby clustering semantically similar contents automatically. The features of different images are pushed away, but features of similar objects in different images are pulled closer. Thus we speculate that contrastive learning enables models with better identification of single object than supervised models.

To verify this, we pretrain CC and contrastive learning on the whole *mini*ImageNet ($\mathcal{D}_B$-Full) and compare their accuracy on $\mathcal{D}_v$-Ori and $\mathcal{D}_v$-FG. The contrastive learning method we use is Exampler [62], a modified version of MoCo [14]. Unsurprisingly, Fig 2(b) shows that Exampler performs much better when only foregrounds of testing images are given, which affirms that contrastive learning has a better discriminative ability of single object. The evaluation accuracy of Exampler on $\mathcal{D}_v$-Ori is slightly worse than that of CC. Contrastive learning aims to discriminate between every image, without knowing what foreground and background are, thus the model judges that two images are similar only if both foreground and background are similar. This may lead to misclassification when the foreground object is the same but the background differs. In Appendix D.2, we provide a more in-depth analysis of why contrastive learning has such properties and find that the shape bias and viewpoint invariance of contrastive learning may play an important role.

4 Rectifying Shortcut Learning of Background

Given the analysis in Sec. 3.1, we want the model to focus more on image foreground both at pretraining and evaluation stages. And in Sec. 3.2 we reveal the potential of contrastive learning in discriminating foreground objects. Thus in this section, we propose COSOC, a two-stage framework which ultilizes contrastive learning to draw the model's attention to the foreground objects of images.

4.1 Pretraining Stage: Clustering-based Object Seeker (COS) with Fusion Sampling

Since contrastive learning is good at discriminating foreground objects, we utilize it to make up for deficiencies of supervised few-shot learning models. The first step is to pretrain a backbone $f_\theta(\cdot)$ using Exempler [62]. Then COS algorithm is used to extract "objects" identified by the pretrained model. The basic idea is that features of foreground objects in images within one class extracted by contrastive learning models are similar, thereby can be identified via a clustering algorithm, see Fig. 3. All images within the i-th class in $\mathcal{D}_B$ form a set $\{\mathbf{x}_n^i\}_{n=1}^N$. We will omit the class index i for brevity. The scheme of seeking for foreground objects in the i-th class can be described as follows:

1) For each image $\mathbf{x}_n$, we randomly crop it L times to obtain L image patches $\{\mathbf{p}_{n,m}\}_{m=1}^L$. Each image patch $\mathbf{p}_{n,m}$ is then passed through the pretrained model f_θ and gets a normalized feature vector $\mathbf{v}_{n,m} = \frac{f_\theta(\mathbf{p}_{n,m})}{||f_\theta(\mathbf{p}_{n,m})||_2} \in \mathbb{R}^d$.

2) We run a clustering algorithm $\mathcal{A}$ on all features vectors and obtain H clusters $\{\mathbf{z}_j\}_{j=1}^H = \mathcal{A}(\{\mathbf{v}_{n,m}\}_{n,m=1}^{N,L})$, where $\mathbf{z}_j$ is the center feature of the j-th cluster.

3) We say an image $\mathbf{x}_n \in \mathbf{z}_j$, if there exists $k \in [L]$ s.t. $\mathbf{v}_{n,k} \in \mathbf{z}_j$, where $[L] = \{1, 2, \ldots, L\}$. Let $l(\mathbf{z}_j) = \frac{\#\{\mathbf{x}|\mathbf{x}\in\mathbf{z}_j\}}{N}$ be the proportion of images in class c that belong to $\mathbf{z}_j$. If $l(\mathbf{z}_j)$ is small, then the cluster $\mathbf{z}_j$ is not representative for the whole class and is possibly background. Thus we remove

Figure 4: Examples of obtained objects from pretraining set of *mini*ImageNet. The first row shows the original images. The second row shows the picked patch with the highest foreground score.

all the clusters $\mathbf{z}$ with $l(\mathbf{z}) < \gamma$, where γ is a threshold that controls the generality of clusters. The remaining h clusters $\{\mathbf{z}_j\}_{j=\alpha_1}^{\alpha_h}$ are "objects" of the class that we find.

4) The foreground score of image patch $p_{n,m}$ is defined as $s_{n,m} = 1 - \min_{j\in[h]} ||\mathbf{v}_{n,m} - \mathbf{z}_{\alpha_j}||_2/\eta$, where $\eta = \max_{n,m} \min_{j\in[h_c]} ||\mathbf{v}_{n,m} - \mathbf{z}_{\alpha_j}||_2$ is used to normalize the score into $[0, 1]$. Then top-k scores of each image $\mathbf{x}_n$ are obtained as $\{s_{n,m}\}_{m=\beta_1}^{\beta_k} = \underset{m\in[L]}{\text{Topk}}(s_{n,m})$. The corresponding patches $\{\mathbf{p}_{n,m}\}_{m=\beta_1}^{\beta_k}$ are seen as possible crops of the foreground object in image $\mathbf{x}_n$, and the foreground scores $\{s_{n,m}\}_{m=\beta_1}^{\beta_k}$ as the confidence.

We visualize some obtained "objects" in Fig. 4, and display more results in Appendix G. From the result, we can see that the model indeed extracts relatively reliable crops of foreground objects. We then use it as prior knowledge to rectify shortcut learning of background of supervised models.

The strategy resembles the simple fusion strategy introduced in Sec. 3.1. For an image $\mathbf{x}_n$, the probability that we choose the original version is $1 - \max_{i\in[k]} s_{n,\beta_i}$, and the probability of choosing $\mathbf{p}_{n,m}$ is $(s_{n,m}/\sum_{i\in[k]} s_{n,\beta_i}) \cdot \max_{i\in[k]} s_{n,\beta_i}$. Then in any circumstances, we randomly crop the chosen image patch and make sure that the least area proportion to the original image is a constant. We use this strategy to pretrain a few-shot learning model $f_\theta(\cdot)$ from scratch or finetune from Exampler-pretrained model to maximally improve the ability of model to discriminate foreground objects while retain the ability to adapt to complex scenes.

4.2 Downstream Stage: Few-shot Learning with Shared Object Concentrator (SOC)

As shown in Sec. 3.1, if the accurate crop of the foreground object in the image is given at the evaluation stage, the performance of the model will be improved by a large margin, serving as an upper bound of the model performance. To approach this upper bound, we propose SOC, an algorithm that imitates human inferential thinking, to capture foreground objects by seeking shared contents among training images from the same class and testing images.

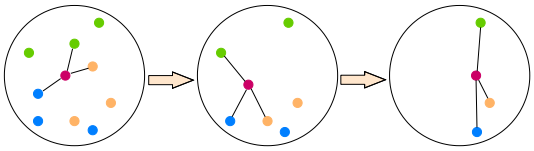

Figure 5: The overall pipeline of step 1 in SOC. Points in one color represent features of crops from one image. The red points are $\boldsymbol{\omega}_1, \boldsymbol{\omega}_2$ and $\boldsymbol{\omega}_3$.

Step 1: Shared Content Searching within Each Class. For each image $\mathbf{x}_k$ within one class c from training set $\mathcal{S}_\tau$, we randomly crop it V times and obtain corresponding candidates $\{\mathbf{p}_{k,n}\}_{n=1,..,V}$. Each patch $\mathbf{p}_{k,n}$ is individually sent to the learned backbone f_θ to obtain a normalized feature vector $\mathbf{v}_{k,n}$. Thus we have totally $K \times V$ feature vectors within a class c. Our goal is to obtain a feature vector $\boldsymbol{\omega}_1$ that can maximally represent the shared information of all images in class c. Ideally, $\boldsymbol{\omega}_1$ represents the center of the most similar K image patches, each from one image, which can be formulated as

$$\boldsymbol{\omega}_1 = \frac{1}{K}\sum_{k=1}^{K} \mathbf{v}_{k,\lambda_{opt}(k)}, \quad \lambda_{opt} = \underset{\lambda\in[K]^{[V]}}{\arg\min} \sum_{1\le i<j\le K} \cos(\mathbf{v}_{i,\lambda(i)}, \mathbf{v}_{j,\lambda(j)}), \tag{1}$$

where $\cos(\cdot,\cdot)$ denotes cosine distance and $[K]^{[V]}$ denotes the set of functions that take $[K]$ as domain and $[V]$ as range. While λ_{opt} can be obtained by enumerating all possible combinations of image patches, the computation complexity of this brute-force method is $\mathcal{O}(V^K)$, which is computation prohibitive when V or K is large. Thus when the computation is not affordable, we turn to use a

simplified method that leverages iterative optimization. Instead of seeking for the closest image patches, we directly optimize $\boldsymbol{\omega}_1$ so that the sum of minimum distance to patches of each image is minimized, *i.e.*,

$$\boldsymbol{\omega}_1 = \arg\min_{\boldsymbol{\omega}\in\mathcal{R}^d} \sum_{k=1}^{K} \min_n [\cos(\boldsymbol{\omega}, \mathbf{v}_{k,n})], \tag{2}$$

which can be achieved by any optimization method. SGD is applied in our experiments. After optimization, we remove the most similar image patch with $\boldsymbol{\omega}_1$ in each image, and obtain $K \times (V-1)$ feature vectors. Then we repeatedly implement the above optimization process until no vectors are left, as shown in Fig. 5. We finally obtain V sorted feature vectors $\{\boldsymbol{\omega}_n\}_{n=1}^{V}$, which we use to represent possible objects inside class c. If shot $K = 1$, then we use the original V feature vectors.

Step 2: Locate Foreground Object of Testing Images. Once the core objects of each class are identified, the next step is to use them to locate foreground objects in the testing images in the testing set $\mathcal{Q}_\tau$. For each testing image $\mathbf{x}$, we again randomly crop it for V times and obtain V candidate features $\{\boldsymbol{\mu}_n\}_{n=1}^{V}$. For each class c, we have V feature vectors $\{\alpha^{n-1}\boldsymbol{\omega}_n\}_{n=1}^{V}$, where $\alpha \leq 1$ is an importance factor. We then match the most similar patches between testing features and class features, *i.e.*,

$$s_1 = \max_{1\leq i,j\leq V} [-\alpha^{j-1}\cos(\boldsymbol{\mu}_i, \boldsymbol{\omega}_j)], \tag{3}$$

similarly, the two matched features are removed and the above process repeats until no features left. Finally, the score of $\mathbf{x}$ w.r.t. class c is obtained as a weighted sum of all similarities, *i.e.*, $S_c = \sum_{n=1}^{V} \beta^{n-1} s_n$, where $\beta \leq 1$ is an importance factor controlling the belief of each crop being foreground objects. The predicted class of $\mathbf{x}$ is the one with the highest score.

5 Experiments

5.1 Experiment Setup

Dataset. We adopt two benchmark datasets which are the most representative in few-shot learning. The first is *mini*ImageNet [48], a small subset of ILSVRC-12 [40] that contains 600 images within each of the 100 categories. The categories are split into 64, 16, 20 classes for pretraining, validation and evaluation, respectively. The second dataset, *tiered*ImageNet [38], is a much larger subset of ILSVRC-12 and is more challenging. It is constructed by choosing 34 super-classes with 608 categories. The super-classes are split into 20, 6, 8 super-classes which ensures separation between pretraining and evaluation categories. The final dataset contains 351, 97, 160 classes for pretraining, validation and evaluation, respectively. On both datasets, the input image size is 84 × 84 for fair comparison.

Evaluation Protocols. We follow the 5-way 5-shot (1-shot) FSL evaluation setting. Specifically, 2000 tasks, each contains 15 testing images and 5 (1) training images per class, are randomly sampled from the evaluation set $\mathcal{D}_v$ and the average classification accuracy is computed. This is repeated 5 times and the mean of the average accuracy with a 95% confidence interval is reported.

Implementation Details. The backbone we use throughout the article is ResNet-12, which is widely used in few-shot learning. We use Pytorch [35] to implement all our experiments on two NVIDIA 1080Ti GPUs. We train the model using SGD with cosine learning rate schedule without restart to reduce the number of hyperparameters (Which epochs to decay the learning rate). The initial learning rate for training Exampler is 0.1, and for CC is 0.005. The batch size for Exampler, CC are 256 and 128, respectively. For *mini*ImageNet, we train Exampler for 150k iterations, and train CC for 6k iterations. For *tiered*ImageNet, we train Exampler for approximately 900k iterations, and train CC for 120k iterations. We choose k-means [28] as the clustering algorithm for COS. The threshold γ is set to 0.5, and top 3 out of 30 features are chosen per image at the pretraining stage. At the evaluation stage, we crop each image 7 times. The importance factors α and β are both set to 0.8.

5.2 Model Analysis

In this subsection, we show the effectiveness of each component of our method. Tab. 1 shows the ablation study conducted on the full *mini*ImageNet. Since the aim of SOC algorithm is to find foreground objects, it is not necessary to evaluate SOC on the foreground dataset $\mathcal{D}_v$-FG.

Table 1: Ablative study on *mini*ImageNet. All models are pretrained on full *mini*ImageNet.

CC	COS	SOC	$\mathcal{D}_v$-Ori		$\mathcal{D}_v$-FG	
			1-shot	5-shot	1-shot	5-shot
✓			62.67 $\pm$ 0.32	80.22 $\pm$ 0.24	66.69 $\pm$ 0.32	82.86 $\pm$ 0.19
✓	✓		65.05 $\pm$ 0.06	81.16 $\pm$ 0.17	**71.36** $\pm$ 0.30	**86.20** $\pm$ 0.14
✓		✓	64.41 $\pm$ 0.22	81.54 $\pm$ 0.28	-	-
✓	✓	✓	**69.29** $\pm$ 0.12	**84.94** $\pm$ 0.28	-	-

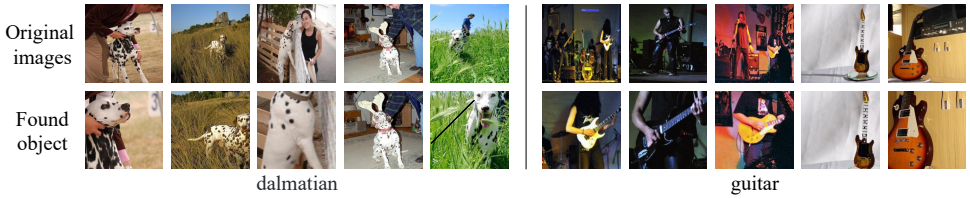

Figure 6: Visualization examples of the SOC algorithm.The first row displays 5 images that belong to dalmatian and guitar classes respectively from evaluation set of miniImageNet. The second row shows image patches that are picked up from the first round of SOC algorithm. Our method succesfully puts focus on the shared contents/foreground.

Importance of COS Algorithm. When COS is applied on CC, the performance is improved on both versions of datasets. Especially, it improves CC on the foreground version $\mathcal{D}_v$-FG by 3.45%, 2.75% under 1-shot and 5-shot settings, respectively. While not substantial, it still improves CC on the original version $\mathcal{D}_v$-Ori by 2.38%, 0.96% under 1-shot and 5-shot settings, respectively. This verifies that foreground objects obtained by the COS algorithm are relatively reliable, and the model pretrained with these objects using fusion strategy improves the ability of discriminating core objects while maintains the ability to handle complex backgrounds.

In Fig. 7, we show the curves of training and validation error during pretraining of CC with and without COS. Both models are pretrained and validated on the full *mini*ImageNet. As we can see, CC sinks into overfitting: the training accuracy drops to zero, and validation accuracy stops improving before the end of the pretraining. Meanwhile, the COS algorithm slows down convergence and prevents training accuracy from reaching zero. This makes validation accuracy comparable at first but higher at the end. Our COS with fusion strategy weakens the "background shortcut" for learning and draws model's attention toward discriminating between foreground objects, which increases the difficulty of the task and improves generalization.

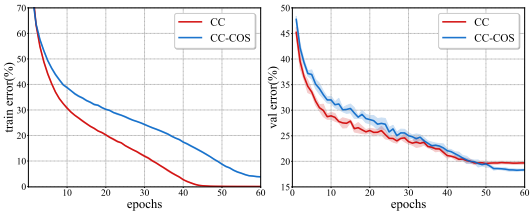

Figure 7: Comparison of training and validation curves between CC with and without COS.

Effectiveness of SOC Algorithm. The result in Tab. 1 shows that the SOC algorithm is the key to maximally exploit the potential of good object-discrimination ability: it further boosts performance on $\mathcal{D}_v$-Ori by 4.24%, 3.78% under 1-shot and 5-shot settings, respectively. This even approaches the upper bound performance obtained by evaluating the model on $\mathcal{D}_v$-FG, whose images perfectly bound foreground objects, which implies that the model with the SOC algorithm successfully concentrates on the foreground objects.

Note that if we apply only the SOC algorithm on CC, the performance degrades. This indicates that COS and SOC are both necessary: COS provides the discrimination ability of foreground objects and SOC leverages it to maximally boost the performance.

Visualization of SOC. Fig. 6 demonstrates two visualization examples of SOC. In each example, we show images within one class from a 5-shot task. The first row shows the original images, and the second row shows the corresponding automatically-picked patches. The original images are from unknown classes—"dalmatian" and "guitar"—which the model has never met and contain noisy backgrounds. For instance, the first and third guitar images do not only describe guitars but in fact display scenes where one band is playing on stage. It just happened that guitars appear in the scenes, taking up a small space. If without any prior knowledge, the model does not know what to

Table 2: Comparisons with state-of-the-art models on *mini*ImageNet and *tiered*ImageNet. The average **inductive** 5-way few-shot classification accuracies with 95 confidence interval are reported.

Model	backbone	*mini*ImageNet		*tiered*ImageNet	
		1-shot	5-shot	1-shot	5-shot
MatchingNet [48]	ResNet-12	63.08 $\pm$ 0.80	75.99 $\pm$ 0.60	68.50 $\pm$ 0.92	80.60 $\pm$ 0.71
TADAM [31]	ResNet-12	58.50 $\pm$ 0.30	76.70 $\pm$ 0.30	-	-
LEO [41]	WRN-28-10	61.76 $\pm$ 0.08	77.59 $\pm$ 0.12	66.33 $\pm$ 0.05	81.44 $\pm$ 0.09
wDAE-GNN [11]	WRN-28-10	61.07 $\pm$ 0.15	76.75 $\pm$ 0.11	68.18 $\pm$ 0.16	83.09 $\pm$ 0.12
MetaOptNet [19]	ResNet-12	62.64 $\pm$ 0.82	78.63 $\pm$ 0.46	65.99 $\pm$ 0.72	81.56 $\pm$ 0.53
DC [25]	ResNet-12	62.53 $\pm$ 0.19	79.77 $\pm$ 0.19	-	-
CTM [22]	ResNet-18	64.12 $\pm$ 0.82	80.51 $\pm$ 0.13	68.41 $\pm$ 0.39	84.28 $\pm$ 1.73
CAM [16]	ResNet-12	63.85 $\pm$ 0.48	79.44 $\pm$ 0.34	69.89 $\pm$ 0.51	84.23 $\pm$ 0.37
AFHN [23]	ResNet-18	62.38 $\pm$ 0.72	78.16 $\pm$ 0.56	-	-
DSN [43]	ResNet-12	62.64 $\pm$ 0.66	78.83 $\pm$ 0.45	66.22 $\pm$ 0.75	82.79 $\pm$ 0.48
AM3+TRAML [20]	ResNet-12	67.10 $\pm$ 0.52	79.54 $\pm$ 0.60	-	-
FEAT [56]	ResNet-12	66.78 $\pm$ 0.20	82.05 $\pm$ 0.14	-	-
DeepEMD [59]	ResNet-12	68.77 $\pm$ 0.29	84.13 $\pm$ 0.53	74.29 $\pm$ 0.32	86.98 $\pm$ 0.60
Net-Cosine [26]	ResNet-12	63.85 $\pm$ 0.81	81.57 $\pm$ 0.56	-	-
RFS-Distill [47]	ResNet-12	64.82 $\pm$ 0.60	82.14 $\pm$ 0.43	71.52 $\pm$ 0.69	86.03 $\pm$ 0.49
CA [1]	WRN-28-10	65.92 $\pm$ 0.60	82.85 $\pm$ 0.55	**74.40 $\pm$ 0.68**	86.61 $\pm$ 0.59
MABAS [18]	ResNet-12	65.08 $\pm$ 0.86	82.70 $\pm$ 0.54	-	-
ConsNet [55]	ResNet-12	64.89 $\pm$ 0.23	79.95 $\pm$ 0.17	-	-
IEPT [60]	ResNet-12	67.05 $\pm$ 0.44	82.90 $\pm$ 0.30	72.24 $\pm$ 0.50	86.73 $\pm$ 0.34
MELR [7]	ResNet-12	67.40 $\pm$ 0.43	83.40 $\pm$ 0.28	72.14 $\pm$ 0.51	87.01 $\pm$ 0.35
IER-Distill [39]	ResNet-12	67.28 $\pm$ 0.80	84.78 $\pm$ 0.52	72.21 $\pm$ 0.90	87.08 $\pm$ 0.58
LDAMF [53]	ResNet-12	67.76 $\pm$ 0.46	82.71 $\pm$ 0.31	71.89 $\pm$ 0.52	85.96 $\pm$ 0.35
FRN [51]	ResNet-12	66.45 $\pm$ 0.19	82.83 $\pm$ 0.13	72.06 $\pm$ 0.22	86.89 $\pm$ 0.14
COSOC (ours)	ResNet-12	**69.28 $\pm$ 0.49**	**85.16 $\pm$ 0.42**	73.57 $\pm$ 0.43	**87.57 $\pm$ 0.10**

focus on in the images, thus has to embed the whole image, and the resulting feature will contain a lot of class-irrelevant information. From the second row, we can observe that SOC successfully focuses on the foreground and extract corresponding patches from the images by searching for shared information from all 5 images. More visualization results can be found in Appendix G.

5.3 Comparison to State-of-the-Arts

Tab. 2 presents 5-way 1-shot and 5-shot classification results on *mini*ImageNet and *tiered*ImageNet. We compare with state-of-the-art few-shot learning methods which are sorted by the year of publishment. Our method uses a relatively shallower backbone (ResNet-12) but achieves comparable/stronger performance on both datasets. Particularly, our method outperforms the most recent work IER-Distill [39] by 2.0% in the 1-shot task and 0.38% in the 5-shot task on *mini*ImageNet, and on *tiered*ImageNet the improvement is 1.36% in the 1-shot task and 0.49% in the 5-shot task. Our method achieves state-of-the-art performance under all settings except for 1-shot task on *tiered*ImageNet, in which the performance of our method is slightly worse than CA [1], which uses WRN-28-10, a deeper backbone, as the feature extractor.

6 Conclusion

Few-shot image classification benefits from increasingly more complex network and algorithm design, but little attention has been focused on image itself. In this paper, we reveal that image background serves as a source of harmful knowledge that few-shot learning models easily absorb in. This problem is tackled by our COSOC method which successfully draws the model's attention to image foreground both at the pretraining and evaluation stage. However, random cropping in our algorithm brings instability into models: what if there are no crops accurately bounding the foreground objects? It is future work to consider solving this problem by identifying approximate object-based regions as potential cropping positions. This may be achieved by the use of unsupervised segmentation or detection algorithms.

References

[1] Arman Afrasiyabi, Jean-François Lalonde, and Christian Gagné. Associative alignment for few-shot image classification. In *ECCV*, 2020.

[2] Sungyong Baik, Myungsub Choi, Janghoon Choi, Heewon Kim, and Kyoung Mu Lee. Meta-learning with adaptive hyperparameters. In *NIPS*, 2020.

[3] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In *NIPS*, 2020.

[4] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020.

[5] Zitian Chen, Yanwei Fu, Yu-Xiong Wang, Lin Ma, Wei Liu, and Martial Hebert. Image deformation meta-networks for one-shot learning. In *CVPR*, 2019.

[6] Carl Doersch, Ankush Gupta, and Andrew Zisserman. Crosstransformers: spatially-aware few-shot transfer. In *NIPS*, 2020.

[7] Nanyi Fei, Zhiwu Lu, Tao Xiang, and Songfang Huang. MELR: meta-learning via modeling episode-level relationships for few-shot learning. In *ICLR*, 2021.

[8] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, 2017.

[9] Yizhao Gao, Nanyi Fei, Guangzhen Liu, Zhiwu Lu, Tao Xiang, and Songfang Huang. Contrastive prototype learning with augmented embeddings for few-shot learning. *arXiv preprint arXiv:2101.09499*, 2021.

[10] Spyros Gidaris and Nikos Komodakis. Dynamic few-shot visual learning without forgetting. In *CVPR*, 2018.

[11] Spyros Gidaris and Nikos Komodakis. Generating classification weights with GNN denoising autoencoders for few-shot learning. In *CVPR*, 2019.

[12] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Ávila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent - A new approach to self-supervised learning. In *NIPS*, 2020.

[13] Bharath Hariharan and Ross B. Girshick. Low-shot visual recognition by shrinking and hallucinating features. In *ICCV*, 2017.

[14] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross B. Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020.

[15] Timothy Hospedales, Antreas Antoniou, Paul Micaelli, and Amos Storkey. Meta-learning in neural networks: A survey. *arXiv preprint arXiv:2004.05439*, 2020.

[16] Ruibing Hou, Hong Chang, Bingpeng Ma, Shiguang Shan, and Xilin Chen. Cross attention network for few-shot classification. In *NIPS*, 2019.

[17] Muhammad Abdullah Jamal and Guo-Jun Qi. Task agnostic meta-learning for few-shot learning. In *CVPR*, 2019.

[18] Jaekyeom Kim, Hyoungseok Kim, and Gunhee Kim. Model-agnostic boundary-adversarial sampling for test-time generalization in few-shot learning. In *ECCV*, 2020.

[19] Kwonjoon Lee, Subhransu Maji, Avinash Ravichandran, and Stefano Soatto. Meta-learning with differentiable convex optimization. In *CVPR*, 2019.

[20] Aoxue Li, Weiran Huang, Xu Lan, Jiashi Feng, Zhenguo Li, and Liwei Wang. Boosting few-shot learning with adaptive margin loss. In *CVPR*, 2020.

[21] Fei-Fei Li, Robert Fergus, and Pietro Perona. One-shot learning of object categories. In *TPAMI*, 2006.

[22] Hongyang Li, David Eigen, Samuel Dodge, Matthew Zeiler, and Xiaogang Wang. Finding task-relevant features for few-shot learning by category traversal. In *CVPR*, 2019.

[23] Kai Li, Yulun Zhang, Kunpeng Li, and Yun Fu. Adversarial feature hallucination networks for few-shot learning. In *CVPR*, 2020.

[24] Zhenguo Li, Fengwei Zhou, Fei Chen, and Hang Li. Meta-sgd: Learning to learn quickly for few-shot learning. *arXiv preprint arXiv:1707.09835*, 2017.

[25] Yann Lifchitz, Yannis Avrithis, Sylvaine Picard, and Andrei Bursuc. Dense classification and implanting for few-shot learning. In *CVPR*, 2019.

[26] Bin Liu, Yue Cao, Yutong Lin, Qi Li, Zheng Zhang, Mingsheng Long, and Han Hu. Negative margin matters: Understanding margin in few-shot classification. In *ECCV*, 2020.

[27] Chen Liu, Yanwei Fu, Chengming Xu, Siqian Yang, and Jilin Li. Learning a few-shot embedding model with contrastive learning. In *AAAI*, 2021.

[28] James MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, 1967.

[29] Orchid Majumder, Avinash Ravichandran, Subhransu Maji, Marzia Polito, Rahul Bhotika, and Stefano Soatto. Revisiting contrastive learning for few-shot classification. *arXiv preprint arXiv:2101.11058*, 2021.

[30] Tsendsuren Munkhdalai and Hong Yu. Meta networks. In *ICLR*, 2017.

[31] Boris N. Oreshkin, Pau Rodríguez López, and Alexandre Lacoste. TADAM: task dependent adaptive metric for improved few-shot learning. In *NIPS*, 2018.

[32] Yassine Ouali, Céline Hudelot, and Myriam Tami. Spatial contrastive learning for few-shot classification. *arXiv preprint arXiv:2012.13831*, 2020.

[33] Eunbyung Park and Junier B. Oliva. Meta-curvature. In *NIPS*, 2019.

[34] Seong-Jin Park, Seungju Han, Ji-Won Baek, Insoo Kim, Juhwan Song, Haebeom Lee, Jae-Joon Han, and Sung Ju Hwang. Meta variance transfer: Learning to augment from the others. In *ICML*, 2020.

[35] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *NIPS*, 2019.

[36] Aravind Rajeswaran, Chelsea Finn, Sham M. Kakade, and Sergey Levine. Meta-learning with implicit gradients. In *NIPS*, 2019.

[37] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. In *ICLR*, 2017.

[38] Mengye Ren, Eleni Triantafillou, Sachin Ravi, Jake Snell, Kevin Swersky, Joshua B. Tenenbaum, Hugo Larochelle, and Richard S. Zemel. Meta-learning for semi-supervised few-shot classification. In *ICLR*, 2018.

[39] Mamshad Nayeem Rizve, Salman Khan, Fahad Shahbaz Khan, and Mubarak Shah. Exploring complementary strengths of invariant and equivariant representations for few-shot learning. In *CVPR*, 2021.

[40] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Fei-Fei Li. Imagenet large scale visual recognition challenge. In *IJCV*, 2015.

[41] Andrei A. Rusu, Dushyant Rao, Jakub Sygnowski, Oriol Vinyals, Razvan Pascanu, Simon Osindero, and Raia Hadsell. Meta-learning with latent embedding optimization. In *ICLR*, 2019.

[42] Eli Schwartz, Leonid Karlinsky, Joseph Shtok, Sivan Harary, Mattias Marder, Abhishek Kumar, Rogério Schmidt Feris, Raja Giryes, and Alexander M. Bronstein. Delta-encoder: an effective sample synthesis method for few-shot object recognition. In *NIPS*, 2018.

[43] Christian Simon, Piotr Koniusz, Richard Nock, and Mehrtash Harandi. Adaptive subspaces for few-shot learning. In *CVPR*, 2020.

[44] Jake Snell, Kevin Swersky, and Richard S. Zemel. Prototypical networks for few-shot learning. In *NIPS*, 2017.

[45] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip H. S. Torr, and Timothy M. Hospedales. Learning to compare: Relation network for few-shot learning. In *CVPR*, 2018.

[46] Sebastian Thrun and Lorien Y. Pratt. Learning to learn: Introduction and overview. In *Learning to Learn*, 1998.

[47] Yonglong Tian, Yue Wang, Dilip Krishnan, Joshua B. Tenenbaum, and Phillip Isola. Rethinking few-shot image classification: A good embedding is all you need? In *ECCV*, 2020.

[48] Oriol Vinyals, Charles Blundell, Tim Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. In *NIPS*, 2016.

[49] Yaqing Wang, Quanming Yao, James T. Kwok, and Lionel M. Ni. Generalizing from a few examples: A survey on few-shot learning. In *ACM Comput. Surv.*, 2020.

[50] Yu-Xiong Wang, Ross B. Girshick, Martial Hebert, and Bharath Hariharan. Low-shot learning from imaginary data. In *CVPR*, 2018.

[51] Davis Wertheimer, Luming Tang, and Bharath Hariharan. Few-shot classification with feature map reconstruction networks. In *CVPR*, 2021.

[52] Kai Xiao, Logan Engstrom, Andrew Ilyas, and Aleksander Madry. Noise or signal: The role of image backgrounds in object recognition. In *ICLR*, 2021.

[53] Chengming Xu, Chen Liu, Li Zhang, Chengjie Wang, Jilin Li, Feiyue Huang, Xiangyang Xue, and Yanwei Fu. Learning dynamic alignment via meta-filter for few-shot learning. In *CVPR*, 2021.

[54] Jin Xu, Jean-Francois Ton, Hyunjik Kim, Adam R. Kosiorek, and Yee Whye Teh. Metafun: Meta-learning with iterative functional updates. In *ICML*, 2020.

[55] Weijian Xu, yifan xu, Huaijin Wang, and Zhuowen Tu. Attentional constellation nets for few-shot learning. In *ICLR*, 2021.

[56] Han-Jia Ye, Hexiang Hu, De-Chuan Zhan, and Fei Sha. Few-shot learning via embedding adaptation with set-to-set functions. In *CVPR*, 2020.

[57] Sung Whan Yoon, Jun Seo, and Jaekyun Moon. Tapnet: Neural network augmented with task-adaptive projection for few-shot learning. In *ICML*, 2019.

[58] Zhongqi Yue, Hanwang Zhang, Qianru Sun, and Xian-Sheng Hua. Interventional few-shot learning. In *NIPS*, 2020.

[59] Chi Zhang, Yujun Cai, Guosheng Lin, and Chunhua Shen. Deepemd: Few-shot image classification with differentiable earth mover's distance and structured classifiers. In *CVPR*, 2020.

[60] Manli Zhang, Jianhong Zhang, Zhiwu Lu, Tao Xiang, Mingyu Ding, and Songfang Huang. IEPT: instance-level and episode-level pretext tasks for few-shot learning. In *ICLR*, 2021.

[61] Ruixiang Zhang, Tong Che, Zoubin Ghahramani, Yoshua Bengio, and Yangqiu Song. Metagan: An adversarial approach to few-shot learning. In *NIPS*, 2018.

[62] Nanxuan Zhao, Zhirong Wu, Rynson WH Lau, and Stephen Lin. What makes instance discrimination good for transfer learning? In *ICLR*, 2021.

[63] Luisa M. Zintgraf, Kyriacos Shiarlis, Vitaly Kurin, Katja Hofmann, and Shimon Whiteson. Fast context adaptation via meta-learning. In *ICML*, 2019.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] Described in Sec. 6.
   (c) Did you discuss any potential negative societal impacts of your work? [No]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] In the supplemental material.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] Data splits and hyperparameters are described in Sec. 5.1. In Appendix E we discuss how they are chosen.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] See Fig. 7; or Fig. 2 in Appendix.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] Described in Sec. 5.1.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]