Rethinking Generalization in Few-Shot Classification

Anonymous Author(s)
Affiliation
Address
email

Abstract

One-hot image-level annotations only correctly describe an often small subset of an image's content, particularly when complex real-world scenes are depicted. While this is tolerable in many classification scenarios, it poses a significant challenge for applications where the set of classes differs significantly between training and test time. In this paper, we take a closer look at the implications in the context of *few-shot learning*. Splitting the input samples into patches and encoding these via the help of Vision Transformers allows us to establish semantic correspondences between local regions across images and independent of their respective class. The most informative patch embeddings for the task at hand are then determined as a function of the support set via online optimization at inference time, additionally providing visual interpretability of '*what matters most*' in the image. We build on recent advances in unsupervised training of networks via masked image modelling to overcome the lack of fine-grained labels and learn the more general statistical structure of the data while avoiding negative image-level annotation influence, *aka* supervision collapse. Experimental results show the competitiveness of our approach, achieving new state-of-the-art results on four popular few-shot classification benchmarks for 5-shot and 1-shot scenarios.

1 Introduction

Images depicting real-world scenes are usually comprised of several different entities, e.g., a person walking their dog in a park surrounded by trees, or a dog playing with a ball (Figure 1). Nevertheless, popular computer vision datasets like ImageNet [35] assign a single *image-level* annotation to classify their entire content. Hence, such a label only correctly applies to an often small subset of the actual image. As a result, models trained on such data via gradient-based methods learn to ignore all seemingly irrelevant information, particularly entities that occur across differently labelled images. While this is acceptable for conventional classification methods that encounter a diverse number of training examples for all classes they are expected to classify, it poses a major but often overlooked challenge for applications where the set of classes differs between training and test time. One such affected area is few-shot learning (FSL) where approaches are expected to correctly classify entirely new classes at test time that have never been encountered during training, just by being provided with a few (e.g., one or five) samples for each of these new categories. During test time, entities that have not been part of the set of training classes and have possibly been perceived as irrelevant might very well be part of the set of test classes – yet, the method was taught to ignore these. Similarly, a method might overemphasize the importance of certain image patterns learned during training that are however of no relevance for the test classes, resulting in *supervision collapse* [8].

Few previous works [8; 17] partially tackle the above challenges. Particularly, CTX [8] proposes to learn the spatial and semantic alignment between CNN-extracted query and support features using a transformer-style attention mechanism. The authors further show that self-supervised learning tasks (i.e., SimCLR) can be integrated into episodic training along with normal supervised tasks to learn

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

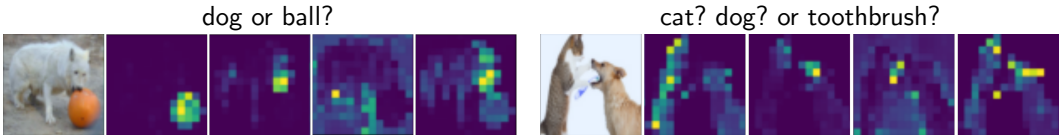


Figure 1: **Challenge of one-hot image-level annotations and our approach to solving it.** Labels assigned to real-world images with multiple entities only correctly describe a (small) subset of the depicted content, leading to supervision collapse for tasks where the set of classes differs between training and test time. The depicted attention maps demonstrate our method's ability to extract information regarding all major entities present in the image including 'background', avoiding supervision collapse.

more generalized features, which benefits solving unseen tasks and mitigates supervision collapse. CAN [17] achieves this in a similar manner by performing cross-attention between class prototypes and query feature maps, highlighting the region of a feature map important for classification during inference. While both methods propose important contributions towards tackling supervision collapse, there exist important drawbacks. Firstly, both methods build their ideas around aligning prototypes based on each query. Such prototypes are merely class-aware and ignore all inter-class information present in the support set – a part that has been shown to be crucial for few-shot learning [32; 46]. Furthermore, learning query-aligned class representations requires performing the same operation for each query, rendering such approaches rather inefficient at inference time.

Summarizing our and previous works' observations, we aim to address what we see as the two main criteria: 1) building an understanding about the image's structure and content that *generalizes* towards new classes, and 2) providing the ability to interpret the provided samples *in context*, i.e., finding the intra-class similarities and inter-class differences while jointly considering all available information.

Our work. To alleviate the negative influence of image-level annotations and to avoid supervision collapse, we decompose the images into patches representing local regions, each having a higher likelihood of being dominated by only one entity. To overcome the lack of such fine-grained annotations we employ self-supervised training with Masked Image Modelling as pretext task [57] and use a Vision Transformer architecture [9] as encoder due to its patch-based nature. We build our classification around the concept of learning task-specific similarity between local regions as a function of the support set at inference time. To this extend, we first create a prior similarity map by establishing semantic patch correspondences between all support set samples irrespective of their class, i.e. also between entities that might not be relevant or potentially even harmful for correct classification. Assume a reduced support set with only two classes: 'person' and 'cat'. Thinking back to Figure 1, the right image is part of our support set for 'cat' – and the dog just happens to be in the image. Now in the query sample that shall be classified, assume the image depicts a person walking their dog. We will thus correctly detect a correspondence of the two dogs across those two images, as well as between the person patches and the other samples of the person support set class. While the correspondences between the person regions are helpful, there is no 'dog' class is in the actual support set (i.e., 'dog' is out-of-task information), rendering this correspondence harmful for classification since it would indicate that the query is connected to the image with the 'cat' label. This is where our *token importance reweighting* comes into play. We infer an importance weight for each token based on its contribution towards correct classification of the other support set samples, actively strengthening intra-class similarities and inter-class differences by jointly considering all available information – in other words, we learn which tokens 'help' or 'harm' our classification objective. These importance-reweighted support set embeddings are then used as basis for our similarity-based query sample classification. Our main contributions include the following.

1. We demonstrate that Transformer-only architectures in conjunction with self-supervised pretraining can be successfully used in few-shot settings without the need of convolutional backbones or any additional data.
2. We show that meta fine-tuning of Vision Transformers combined with our inner loop token importance reweighting can successfully use the supervision signal of provided support set labels while avoiding supervision collapse.
3. We provide insights into how establishing general similarities across images independent of classes followed by our optimization-based selection at inference time can boost generalization while allowing visual interpretability at the same time, and show the efficacy of our method by achieving new state-of-the-art results on four popular public benchmarks.

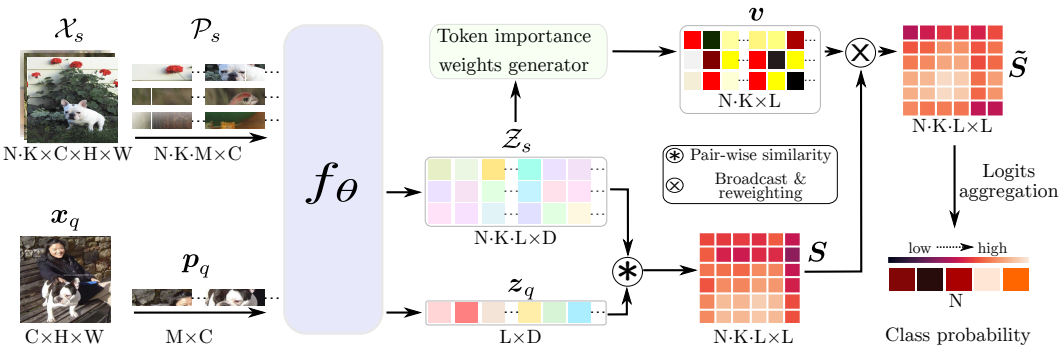


Figure 2: **Illustration of the proposed method FewTURE**. Support and query set images are split into patches and encoded by our Transformer backbone. Classification of query set images is performed by using the reweighted similarity of the encoded patches w.r.t. the support set tokens.

2 Few-shot classification via reweighted embedding similarity

We start this section by briefly introducing the problem setting we are tackling in this work: inductive few-shot classification. We then provide an overview of our proposed method *FewTURE* – Few-shot classification with Transformers Using Reweighted Embedding similarity (Figure 2) before elaborating on the main elements in more detail.

Problem definition. Inductive N-way K-shot few-shot classification aims to generalize knowledge learned during training on $\mathcal{D}_{train}$ to unseen test data $\mathcal{D}_{test}$, with classes $\mathcal{C}_{train} \cap \mathcal{C}_{test} = \emptyset$, using only a few labelled samples. We follow the meta-learning protocol of previous works [46] to formulate the few-shot classification problem with episodic training and testing. An episode $\mathcal{E}$ is composed of a support set $\mathcal{X}_s = \{(\boldsymbol{x}_s^{nk}, \boldsymbol{y}_s^{nk}) | n = 1, \dots, N; k = 1, \dots, K; \boldsymbol{y}_s^{nk} \in \mathcal{C}_{train}\}$, where $\boldsymbol{x}_s^{nk}$ denotes the k-th sample of class n with label $\boldsymbol{y}_s^{nk}$, and a query set $\mathcal{X}_q = \{(\boldsymbol{x}_q^{n}, \boldsymbol{y}_q^{n}) | n = 1, \dots, N\}$, where $\boldsymbol{x}_q^{n}$ denotes a query sample[1] of class n with label $\boldsymbol{y}_q^{n}$.

2.1 Overview on FewTURE

As depicted in Figure 2, we encode the image patches $\mathcal{P}_s$ of the support set samples along with the query sample patches $\mathbf{p}_q$ via f_θ and obtain corresponding sets of tokens $\mathcal{Z}_s$ and $\boldsymbol{z}_q$, respectively. It is to be noted that while we choose to illustrate our method via the use of one single query sample, the classification of all query samples is computed at the same time in one single pass in practice. We retrieve our 'prior' correspondence map $\boldsymbol{S}$ expressing the token-wise similarity between the encoded semantic content of the local regions in the query sample and all patches of all support samples, allowing us to consider all available information jointly without incurring information loss due to averaging or similar operations. This *'prior' similarity map* represents correspondences between regions of samples irrespective of their individual class, i.e. also between entities that might not be relevant or potentially even harmful for correct classification. Using the annotated support set samples, we now infer a task-specific importance weight factor v^j for each support token z_s^j representing its contribution to correctly classify other samples in the support set via online optimization at inference time (Section 2.4). We then reweight the prior similarities to obtain our classification result for the query sample, jointly considering all available information.

2.2 Self-supervised pretraining against supervision collapse

To overcome the problem of supervision collapse induced by image-level annotations, we split the input images into smaller parts where each region has a higher likelihood of only containing one major entity and hence a more distinct semantic meaning. Since no labels are available for this more fine-grained data, we encode the information of each local region via an unsupervised method.

[1] Without loss of generality, we present our method for the case of one query sample per class to improve ease of understanding. The exact number of query samples per class in generally unknown in practice.

We build our approach around the recently introduced idea of using *Masked Image Modeling* (MIM) [3; 57] as a pretext task for self-supervised training of Vision Transformers. In contrast to previous unsupervised approaches [5; 7] which focused mainly on global image-level representations, MIM randomly masks a number of patch embeddings (*tokens*) and aims to reconstruct them given the remaining information of the image. The in this way introduced token constraints help our Transformer backbone to learn an embedding space that yields semantically meaningful representations for each individual image patch. We then leverage the information of the provided labels through fine-tuning the pretrained backbone in conjunction with our inner loop token importance weighting described in the following sections while successfully avoiding supervision collapse (see experimental results in Section 3.2).

2.3 Classification through reweighted token similarity

As illustrated in Figure 2, we split each input image $\boldsymbol{x} \in \mathbb{R}^{H \times W \times C}$ into a sequence of $M = {}^{H \cdot W}/_{P^2}$ patches $\mathbf{p} = \{p^i\}_{i=1}^{M}$, with each patch $p^i \in \mathbb{R}^{P^2 \times C}$. We then flatten and pass all patches of the support and query images as input to the Transformer architecture, obtaining the set of support tokens $\mathcal{Z}_s = f_\theta(\mathcal{P}_s)$ with $\mathcal{Z}_s = \{\boldsymbol{z}_s^{nk} | n = 1, \dots, N, k = 1, \dots, K\}$, $\boldsymbol{z}_s^{nk} = \{z_s^{nkl} | l = 1, \dots, L; z_s^{nkl} \in \mathbb{R}^D\}$ and query tokens $\boldsymbol{z}_q = f_\theta(\mathbf{p}_q)$ with $\boldsymbol{z}_q = \{z_q^l | l = 1, \dots, L; z_q^l \in \mathbb{R}^D\}$. Vision Transformers like ViT [9; 43] satisfy $L = M$ whereas hierarchical Transformers like Swin [25] generally emit a reduced number of tokens $L < M$ due to internal merging strategies.

Having obtained all patch embeddings, we establish semantic correspondences by computing the pair-wise patch similarity matrix between the set of support tokens $\mathcal{Z}_s$ and query tokens $\boldsymbol{z}_q$ as $\boldsymbol{S} \in \mathbb{R}^{N \cdot K \cdot L \times L}$, where each element in $\boldsymbol{S}$ is obtained by $s_{nk}^{l_s, l_q} = \mathrm{sim}(z_s^{nkl_s}, z_q^{l_q})$, where $l_s = 1, \dots, L$ and $l_q = 1, \dots, L$. Note that local image regions representing similar entities exhibit higher scores. While any distance metric can be used to compute the similarity (sim), we found cosine similarity to work particularly well for this task. We then use the task-specific token importance weights $\boldsymbol{v} \in \mathbb{R}^{N \cdot K \cdot L \times 1}$ inferred via online optimization based on the annotated support set samples (see Section 2.4) to reweight the similarities through column-wise addition and obtain our task-specific similarity matrix as $\tilde{\boldsymbol{S}} = \boldsymbol{S} + \left[\boldsymbol{v} \cdot \mathbb{1}^{1 \times L}\right]$, with elements $\tilde{s}_{nk}^{l_s, l_q}$. Note that this addition of our reweighting logits corresponds to multiplicative reweighting in probability space. We temperature-scale the adapted similarity logits with ${}^{1}/_{\tau_S}$ and aggregate the token similarity values across all elements belonging to the same support set class via a *LogSumExp* operation, i.e. aggregating $K \cdot L^2$ logits per class followed by a softmax – resulting in the final class prediction $\hat{\boldsymbol{y}}_q$ for the query sample $\boldsymbol{x}_q$ as

$$\hat{\boldsymbol{y}}_q = \mathrm{softmax}\left(\left\{\hat{y}_q^n\right\}_{n=1}^{N}\right) = \mathrm{softmax}\left(\left\{\log \sum_{k=1}^{K} \sum_{l_q=1}^{L} \sum_{l_s=1}^{L} \exp\left({}^{\tilde{s}_{nk}^{l_s, l_q}}/_{\tau_S}\right)\right\}_{n=1}^{N}\right). \quad (1)$$

2.4 Learning token importance at inference time

We use all samples of a task's support set together with their annotations to learn the importance for each individual patch token via online optimization at inference time. As depicted in Figure 3, we formulate the same classification objective as in the previous section but aim to classify the support set samples instead of a query sample. In other words, we use two versions of the support set samples: one with labels as 'support' $\mathcal{Z}_s$ and one without as 'pseudo-query' $\mathcal{Z}_{sq}$, and obtain the similarity matrix $\boldsymbol{S}_s \in \mathbb{R}^{N \cdot K \cdot L \times N \cdot K \cdot L}$. The token importance weights are initialized to $\boldsymbol{v}^0 = \mathbf{0} \in \mathbb{R}^{N \cdot K \cdot L \times 1}$ and column-wise added to form $\tilde{\boldsymbol{S}}_s = \boldsymbol{S}_s + \left[\boldsymbol{v}^0 \cdot \mathbb{1}^{1 \times N \cdot K \cdot L}\right]$.

The goal is now to determine which tokens of $\mathcal{Z}_s$ are most helpful in contributing towards correctly classifying $\mathcal{Z}_{sq}$, and which ones negatively affect this objective. To prevent tokens from simply classifying themselves, we device the following strategy for our N-way K-shot tasks. For scenarios with $K > 1$ samples per class, we apply block-diagonal masking with blocks of size $L \times L$ to the similarity matrix $\tilde{\boldsymbol{S}}_s$ – meaning that we enforce classification of each token in $\mathcal{Z}_{sq}$ exclusively based on information from other images. Since there are no other in-class examples available in 1-shot scenarios, we slightly weaken the constraint and apply local masking in a $m \times m$ window around each patch, forcing the token to be classified based on the remaining information in the image. We found a local window of $m = 5$ to work well throughout our experiments for both architectures.

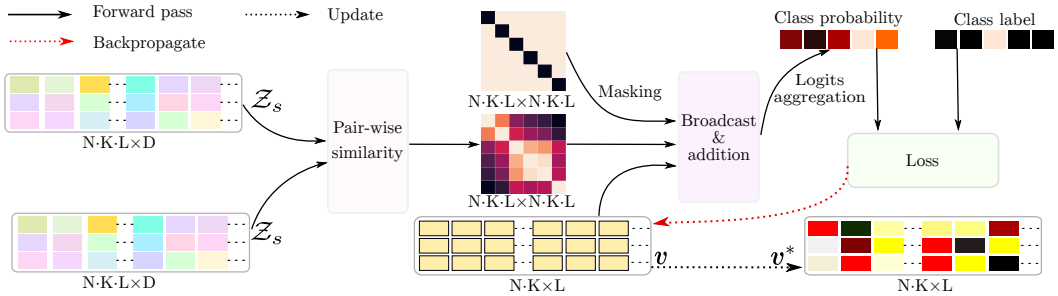


Figure 3: **Inner loop token importance weight generator**. The most helpful tokens for the task at hand are determined as a function of the support set via inner-loop optimization at inference time by reweighting all token similarities based on their contribution towards a correct classification result.

We use temperature-scaling and aggregate all modified similarity logits across all elements belonging to the same support set class of the annotated $\mathcal{Z}_s$ for each element $\boldsymbol{z}_{sq}^{nk}$, apply a softmax operation and obtain the predicted class probabilities $\hat{\boldsymbol{y}}_s^{nk}$ for each support set sample (*cf.* Equation (1)). Given that the prediction is now dependent on the initialized token similarity weights $\boldsymbol{v}$, we can formulate an online optimization objective by using the support set labels $\boldsymbol{y}_s^{nk}$ as

$$\arg\min_{\boldsymbol{v}} \sum_{n=1}^{N} \sum_{k=1}^{K} \mathcal{L}_{\text{CE}} \left(\boldsymbol{y}_s^{nk},\ \hat{\boldsymbol{y}}_s^{nk}(\boldsymbol{v}) \right) . \tag{2}$$

It is to be noted that by using column-wise addition of $\boldsymbol{v}$, we share the importance weights of each support token across all 'pseudo-query' tokens and thus constrain the optimization to jointly learn token importance with respect to all available information. In other words, task-specific intra-class similarities will be strengthened while inter-class ones will be penalized accordingly. We further do not require any second order derivatives like other methods during meta fine-tuning, since the optimization of $\boldsymbol{v}$ is decoupled from the network's parameters.

3 Experiments and discussion

3.1 Implementation details

Our training strategy is divided into two parts: self-supervised pretraining followed by meta fine-tuning. It is to be noted that each architecture is exclusively trained on the training set data of the respective dataset that is to be evaluated, and no additional data is used.

Datasets. We train and evaluate our methods using four popular few-shot classification benchmarks, namely *mini*ImageNet [46], *tiered*ImageNet [34], CIFAR-FS [4] and FC-100 [32].

Architectures. We compare two different Transformer architectures in this work, the monolithic ViT architecture [9; 43] in its 'small' form (*ViT-small aka DeiT-small*) and the multi-scale Swin architecture [25] in its 'tiny' version (*Swin-tiny*).

Self-supervised pretraining. We employ the strategy proposed by [57] to pretrain our Transformer backbones and mostly stick to the hyperparameter settings reported in their work. Our ViT and Swin architectures are trained with a batch size of 512 for 1600 and 800 epochs, respectively. We use 4 Nvidia A100 GPUs with 40GB each for our ViT and 8 such GPUs for our Swin models.

Meta fine-tuning. We use meta fine-tuning to further refine our embedding space by using the available image-level labels in conjunction with our token-reweighting method. We generally train for up to 200 epochs but find most architectures to converge earlier. We evaluate at each epoch on 600 randomly sampled episodes from the respective validation set to select the best set of parameters. During test time, we randomly sample 600 episodes from the test set to evaluate our model.

Token importance reweighting and classifier. We use cosine similarity to compute $\boldsymbol{S}$. While the temperature τ_S used to scale the logits can be learnt during meta fine-tuning, we found $\tau_S = 1/\sqrt{d}$ to be a good default value (or starting point if learnt, see supplementary material). We use SGD as optimizer with a learning rate of 0.1 for the token importance weight generation.

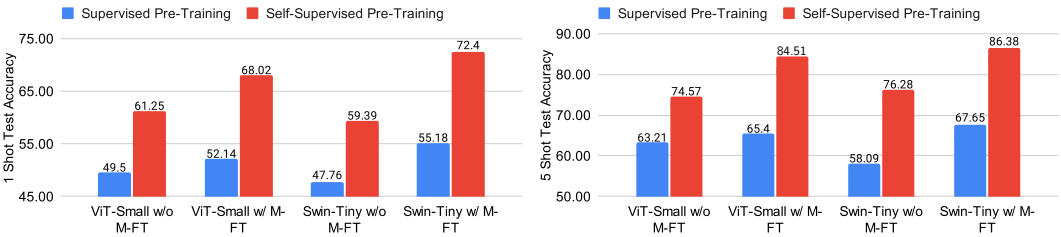


Figure 4: **Supervised vs. self-supervised pretraining.** Average test accuracies on *mini*ImageNet for our method with different pretraining methods, with (*w/*) and without (*w/o*) meta fine-tuning (M-FT).

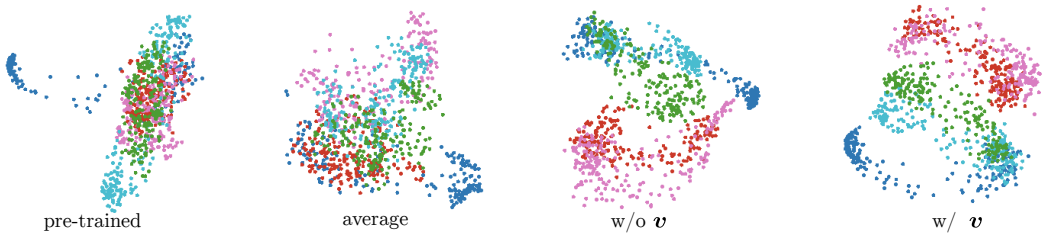


Figure 5: **Instance embeddings after meta fine-tuning.** Visualised are the projected tokens of 5 instances of the same novel support set class for different meta fine-tuning (M-FT) methods (after self-supervised pretraining). From left to right: self-supervised pretraining only, M-FT using an average embedding per class, M-FT using our classifier but without task-specific token reweighting, M-FT using our classifier with 15 reweighting steps. (Projection via PCA and t-SNE [45].)

Please refer to supplementary material of this paper for a more detailed discussion regarding datasets, implementation and hyperparameters.

3.2 Self-supervised pretraining and token-reweighted fine-tuning improve generalization

In this section, we investigate the influence of self-supervised pretraining compared to its supervised counterpart using the provided image-level labels (de-facto standard in most current state-of-the-art methods). We only vary the training strategy of our backbone and use our introduced token similarity-based classifier with 15 inner-loop reweighting steps for all experiments. Figure 4 illustrates that both ViT-small and Swin-tiny with self-supervised pretraining alone (*w/o* meta fine-tuning) learn more generalizable features than their respective supervised version for both 1-shot (left-hand side) and 5-shot (right-hand side) scenarios. In fact, across all cases *w/o* meta fine-tuning, the self-supervised pretrained Transformers outperform their supervised counterparts by more than 10% and up to a significant 18.19% in the case of Swin-tiny. This clearly demonstrates that our self-supervised pretraining strategy explores information that is richer and beyond the labels. Another interesting insight is that the meta fine-tuning stage does not improve supervised backbones as much as their unsupervised counterparts. Specifically after unsupervised pretraining, meta fine-tuning of the Transformers in conjunction with our token-reweighting strategy is able to boost the performance by 6.77% (ViT-small) and 13.01% (Swin-tiny) for 1-shot, and 9.94% (ViT-small) and 10.10% (Swin-tiny) for 5-shot. While such significant improvements cannot be observed across the supervised networks, the Swin versions seem to generally start off lower after pretraining but benefit more from the fine-tuning than ViT. The observed results clearly indicate that our token-reweighted fine-tuning strategy is able to further improve the generalization of self-supervised Transformers, thus performing better on the novel tasks of the unseen test set. Figure 5 additionally depicts projected views of the tokens of 5 instances from a novel class in embedding space. While the representations obtained from the network meta fine-tuned by using common averaging over the embeddings ('*average*') does not exhibit any clear separation of the instances, the embeddings obtained with our classifier seem to retain the instance information ('*w/o* $\boldsymbol{v}$') and separation is improved when using token importance reweighting ('*w/* $\boldsymbol{v}$'). These results indicate that our similarity-based classifier coupled with task-specific token reweighting is able to better disentangle the embeddings of different instances from the same class, which prevents the network from supervision collapse and achieves the higher

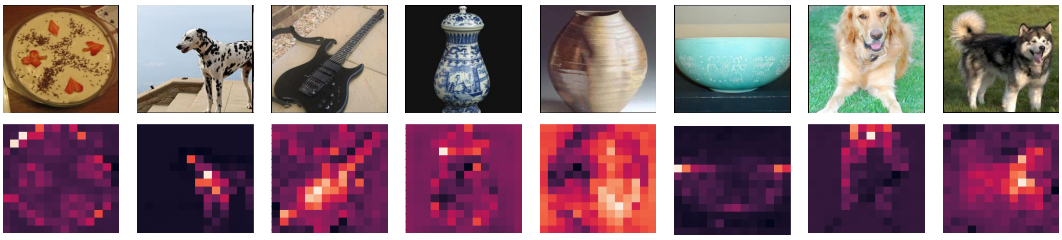


Figure 6: **Learning token importance at inference time.** Visualized importance weights learnt via online optimization for support set samples in a 5-way 5-shot task on the *mini*ImageNet test set.

performance observed on the benchmarks. They further show that self-supervised pretraining is helpful but not sufficient to achieve well-separated representations without supervision collapse that are suitable for few-shot classification.

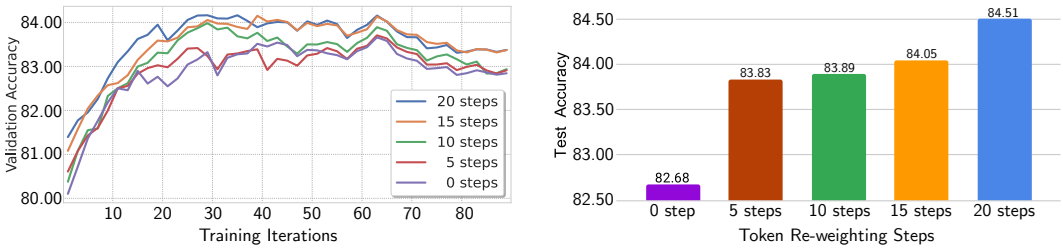
3.3 Selecting helpful patches at inference time

Figure 6 shows a visualisation of the patch importance weights $\boldsymbol{v}$ that are learned at inference time during the inner loop adaptation for the support set images. Brighter regions represent a higher importance weight, meaning that these patches will contribute most to the classification of query samples if matches with high similarity can be found. Judging from the visualized weights, *FewTURE* seems to consistently select characteristic regions of the depicted objects, e.g., the rim of the bowls, strings of the guitar or the dogs' facial area, and to exclude unimportant or out-of-task information.

We further investigate the influence of the number of optimization steps in our inner loop token importance weighting using ViT-small on *mini*ImageNet. The results in Figure 7 indicate that increasing the steps up to 20 aligns with increased performance, both during validation and testing. While the initial increase in test accuracy when using our token reweighting (steps > 0) is rather significant with 1.15%, the contribution of higher step numbers comes at the cost of higher computational complexity, and we generally found anything between 5 and 15 steps to be a good performance *vs*. inference-time trade-off.

3.4 Evaluation on few-shot classification benchmarks

We conduct experiments using the few-shot settings established in the community, namely 5-way 1-shot and 5-way 5-shot – meaning the network has to distinguish samples from 5 novel classes based on a provided number of 1 or 5 images per class. We evaluate our method *FewTURE* using two different Transformer backbones and compare our results against the current state of the art in Table 1 for the *mini*ImageNet and *tiered*ImageNet, and in Table 2 for the CIFAR-100 and FC100 datasets. It is to be noted that in contrast to previous works, we do not employ the help of any convolutional backbone but instead (and as far as we are aware for the first time) use a Transformer backbone together with our previously introduced token importance reweighting method to achieve these results. We are able to set new state of the art results across all four datasets in both 5-shot



(a) (b)

Figure 7: **Inner loop token reweighting.** Average classification accuracies on the *mini*ImageNet validation set (a), and test set (b) for varying inner loop optimization steps, evaluated with a ViT-small backbone and SDG with 0.1 as learning rate.

and 1-shot settings, improving particularly the 1-shot results on *mini*ImageNet, *tiered*Imagenet and CIFAR-FS by significant margins of 3.03%, 1.74% and 1.93%, respectively.

Table 1: Average classification accuracy for 5-way 1-shot and 5-way 5-shot scenarios. Reported are the mean and 95% confidence interval on the unseen test sets of *mini*ImageNet [46] and *tiered*ImageNet [34], using the established evaluation protocols.

Model	**Backbone**	***mini*ImageNet**		***tiered*ImageNet**	
		1-shot	**5-shot**	**1-shot**	**5-shot**
ProtoNet [38]	ResNet-12	62.29±0.33	79.46±0.48	68.25±0.23	84.01±0.56
FEAT [51]	ResNet-12	66.78±0.20	82.05±0.14	70.80±0.23	84.79±0.16
DeepEMD [52]	ResNet-12	65.91±0.82	82.41±0.56	71.16±0.87	86.03±0.58
IEPT [54]	ResNet-12	67.05±0.44	82.90±0.30	72.24±0.50	86.73±0.34
MELR [11]	ResNet-12	67.40±0.43	83.40±0.28	72.14±0.51	87.01±0.35
FRN [47]	ResNet-12	66.45±0.19	82.83±0.13	72.06±0.22	86.89±0.14
CG [56]	ResNet-12	67.02±0.20	82.32±0.14	71.66±0.23	85.50±0.15
DMF [50]	ResNet-12	67.76±0.46	82.71±0.31	71.89±0.52	85.96±0.35
InfoPatch [23]	ResNet-12	67.67±0.45	82.44±0.31	-	-
BML [58]	ResNet-12	67.04±0.63	83.63±0.29	68.99±0.50	85.49±0.34
CNL [56]	ResNet-12	67.96±0.98	83.36±0.51	73.42±0.95	87.72±0.75
Meta-NVG [53]	ResNet-12	67.14±0.80	83.82±0.51	74.58±0.88	86.73±0.61
MetaQDA [55]	WRN-28-10	67.83±0.64	84.28±0.69	74.33±0.65	89.56±0.79
PAL [27]	ResNet-12	69.37±0.64	84.40±0.44	72.25±0.72	86.95±0.47
COSOC [26]	ResNet-12	69.28±0.49	85.16±0.42	73.57±0.43	87.57±0.10
Meta DeepBDC [49]	ResNet-12	67.34±0.43	84.46±0.28	72.34±0.49	87.31±0.32
FewTURE (ours)	ViT-Small	68.02±0.88	84.51±0.53	72.96±0.92	86.43±0.67
FewTURE (ours)	Swin-Tiny	**72.40±0.78**	**86.38±0.49**	**76.32±0.87**	**89.96±0.55**

Table 2: Average classification accuracy for 5-way 1-shot and 5-way 5-shot scenarios. Reported are the mean and 95% confidence interval on the unseen test sets of CIFAR-FS [4] and FC-100 [32], using the established evaluation protocols.

Model	**Backbone**	**CIFAR-FS**		**FC100**	
		1-shot	**5-shot**	**1-shot**	**5-shot**
ProtoNet [38]	ResNet-12	-	-	41.54±0.76	57.08±0.76
MetaOpt [20]	ResNet-12	72.00±0.70	84.20±0.50	41.10±0.60	55.50±0.60
MABAS [19]	ResNet-12	73.51±0.92	85.65±0.65	42.31±0.75	58.16±0.78
RFS [42]	ResNet-12	73.90±0.80	86.90±0.50	44.60±0.70	60.90±0.60
BML [58]	ResNet-12	73.45±0.47	88.04±0.33	-	-
CG [13]	ResNet-12	73.00±0.70	85.80±0.50	-	-
Meta-NVG [53]	ResNet-12	74.63±0.91	86.45±0.59	46.40±0.81	61.33±0.71
Meta-QDA [55]	WRN-28-10	75.83±0.88	88.79±0.75	-	-
RENet [18]	ResNet-12	74.51±0.46	86.60±0.32	-	-
TPMN [48]	ResNet-12	75.50±0.90	87.20±0.60	46.93±0.71	63.26±0.74
MixFSL [1]	ResNet-12	-	-	44.89±0.63	60.70±0.60
FewTURE (ours)	ViT-Small	76.10±0.88	86.14±0.64	46.20±0.79	63.14±0.73
FewTURE (ours)	Swin-Tiny	**77.76±0.81**	**88.90±0.59**	**47.68±0.78**	**63.81±0.75**

3.5 Limitations

Our introduced method is arguably more powerful for cases where multiple examples of the same class are provided, i.e. in N-way K-shot settings with $K > 1$. While FewTURE works well in 1-shot settings (Table 1 and 2), the inner loop adaptation procedure still aims to exclude cross-class similarities that hurt classification performance, but has less diverse information for selecting the most helpful regions due to the lack of other in-class comparison samples – which might thus yield slightly less-refined token selections compared to multi-shot scenarios (*supplementary material*).

While we did not face significant problems regarding the comparably very small size of our datasets (e.g. *mini*ImageNet with 38K compared to the usually used ImageNet with 1.28M training images), more specialized applications with highly limited training data might be negatively impacted and successfully training our method due to the lack of inductive bias in the Transformer architecture could prove challenging.

4 Related work

Over the past few years, the family of few-shot learning (FSL) has grown diverse and broad. Those closely related to this work can be categorized into two groups: metric-based methods [28; 29; 37; 38; 46; 51; 52] and optimisation-based methods [12; 20; 31; 36; 59]. Metric-based methods, such as ProtoNet [38], DeepEMD [52], and RelationNet [40] aim to learn a class representation (prototype) by averaging the embeddings belonging to the same class and employ a predefined (ProtoNet and DeepEMD) or learned (RelationNet) metric to perform prototype-query matching. FEAT [51] takes this a step further and uses a transformer head to adapt the features given the novel tasks, which results in a more adaptive method. Our method instead fully utilizes the embeddings of local image regions (patch tokens), preventing loss of information and supervision collapse occurring in the aforementioned prototype-based approaches (see supplementary material).

Optimisation-based methods such as MAML [12] and Reptile [31] propose to learn a set of initial model parameters that can quickly adapt to a novel task. However, updating all model parameters is often not feasible given large backbones and only few labeled samples during inference. To alleviate this so-called meta-overfitting problem, CAVIA [59] and LEO [36] propose to learn and adapt a lower dimensional representation that is mapped onto the network. Our method is inspired by such lower-dimensional adaptation strategies and learns a tiny set of context-aware re-weighting factors online for each novel task without requiring higher-order gradients, resulting in a flexible and efficient framework.

Self-supervised learning for FSL. Although self-supervised learning is underrepresented in the context of few-shot learning, some recent works [14; 30; 39] have shown that self-supervision via pretext tasks can be beneficial when integrated as auxiliary loss. S2M2 [30] employs rotation [15] and exemplars [10] along with common supervised learning during the pre-training stage. CTX [8] demonstrates that SimCLR [6] can be combined with supervised-learning tasks in an episodic-training manner to learn a more generalized model. In contrast, FewTURE demonstrates that the self-supervised pretext task (i.e., Masked Image Modelling [2; 3; 22; 16; 33; 41; 57]), can be used in a standalone manner to learn more generalized features for few-shot learning on small-scale datasets.

Vision Transformers in FSL. Transformers have been immensely successful in the field of Natural Language Processing. Recent studies [9; 21; 25; 44] suggest that encoding the long-range dependency of data via self-attention also yields promising results for vision tasks (e.g., image classification, joint vision-language modeling, etc.). However, there is an important trade-off between leveraging the rich representation capacity and the lack of inductive bias. Transformers have gained a reputation to generally require significantly more training data compared to convolutional neural networks (CNNs), since properties like translation invariance, locality and hierarchical structure of visual data have to be inferred from the data [24]. While this data-hungry nature largely prevented the use of Transformers in problems with scarce data like few-shot learning, some recent works demonstrate that a single transformer head can be successfully adopted to perform feature adaptation [8; 51]. To the best of our knowledge, *FewTURE* is the first approach that uses a fully Transformer-based architecture to obtain representative embeddings while being trained exclusively on training data of the respective few-shot dataset.

5 Conclusion

In this paper, we presented a novel approach to tackle the challenge of supervision collapse introduced by one-hot image-level labels in few-shot learning. We split the input images into smaller parts with higher probability of being dominated by only one entity and encode these local regions using a Vision Transformer architecture pretrained via Masked Image Modeling (MIM) in a self-supervised way to learn a representative embedding space beyond the pure label information. We devise a classifier based on patch embedding similarities and propose a token importance reweighting mechanism to refine the contribution of each local patch towards the overall classification result based around intra-class similarities and inter-class differences as a function of the support set information. Our obtained results demonstrate that our proposed method alleviates the problem of supervision collapse by learning more generalized features, achieving new state-of-the-art results on four popular few-shot classification datasets.

References

[1] Arman Afrasiyabi, Jean-François Lalonde, and Christian Gagn'e. Mixture-based feature space learning for few-shot image classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9041–9051, October 2021.

[2] Sara Atito, Muhammad Awais, and Josef Kittler. Sit: Self-supervised vision transformer. *arXiv preprint arXiv:2104.03602*, 2021.

[3] Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. BEit: BERT pre-training of image transformers. In *International Conference on Learning Representations*, 2022.

[4] Luca Bertinetto, Joao F Henriques, Philip Torr, and Andrea Vedaldi. Meta-learning with differentiable closed-form solvers. In *International Conference on Learning Representations*, 2019.

[5] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9650–9660, 2021.

[6] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning*, pages 1597–1607. PMLR, 2020.

[7] Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9640–9649, 2021.

[8] Carl Doersch, Ankush Gupta, and Andrew Zisserman. Crosstransformers: spatially-aware few-shot transfer. *Advances in Neural Information Processing Systems*, 33:21981–21993, 2020.

[9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[10] Alexey Dosovitskiy, Jost Tobias Springenberg, Martin Riedmiller, and Thomas Brox. Discriminative unsupervised feature learning with convolutional neural networks. *Advances in Neural Information Processing Systems*, 27, 2014.

[11] Nanyi Fei, Zhiwu Lu, Tao Xiang, and Songfang Huang. Melr: Meta-learning via modeling episode-level relationships for few-shot learning. In *International Conference on Learning Representations*, 2020.

[12] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, pages 1126–1135. PMLR, 2017.

[13] Zhi Gao, Yuwei Wu, Yunde Jia, and Mehrtash Harandi. Curvature generation in curved spaces for few-shot learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8691–8700, 2021.

[14] Spyros Gidaris, Andrei Bursuc, Nikos Komodakis, Patrick Pérez, and Matthieu Cord. Boosting few-shot visual learning with self-supervision. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8059–8068, 2019.

[15] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. In *International Conference on Learning Representations*, 2018.

[16] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. *arXiv preprint arXiv:2111.06377*, 2021.

[17] Ruibing Hou, Hong Chang, Bingpeng Ma, Shiguang Shan, and Xilin Chen. Cross attention network for few-shot classification. *Advances in Neural Information Processing Systems*, 32, 2019.

[18] Dahyun Kang, Heeseung Kwon, Juhong Min, and Minsu Cho. Relational embedding for few-shot classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[19] Jaekyeom Kim, Hyoungseok Kim, and Gunhee Kim. Model-agnostic boundary-adversarial sampling for test-time generalization in few-shot learning. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*, pages 599–617. Springer, 2020.

[20] Kwonjoon Lee, Subhransu Maji, Avinash Ravichandran, and Stefano Soatto. Meta-learning with differentiable convex optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10657–10665, 2019.

[21] Chunyuan Li, Jianwei Yang, Pengchuan Zhang, Mei Gao, Bin Xiao, Xiyang Dai, Lu Yuan, and Jianfeng Gao. Efficient self-supervised vision transformers for representation learning. *International Conference on Learning Representations (ICLR)*, 2022.

[22] Zhaowen Li, Zhiyang Chen, Fan Yang, Wei Li, Yousong Zhu, Chaoyang Zhao, Rui Deng, Liwei Wu, Rui Zhao, Ming Tang, et al. Mst: Masked self-supervised transformer for visual representation. *Advances in Neural Information Processing Systems*, 34, 2021.

[23] Chen Liu, Yanwei Fu, Chengming Xu, Siqian Yang, Jilin Li, Chengjie Wang, and Li Zhang. Learning a few-shot embedding model with contrastive learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 8635–8643, 2021.

[24] Yahui Liu, Enver Sangineto, Wei Bi, Nicu Sebe, Bruno Lepri, and Marco Nadai. Efficient training of visual transformers with small datasets. *Advances in Neural Information Processing Systems*, 34, 2021.

[25] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021.

[26] Xu Luo, Longhui Wei, Liangjian Wen, Jinrong Yang, Lingxi Xie, Zenglin Xu, and Qi Tian. Rectifying the shortcut learning of background for few-shot learning. *Advances in Neural Information Processing Systems*, 34, 2021.

[27] Jiawei Ma, Hanchen Xie, Guangxing Han, Shih-Fu Chang, Aram Galstyan, and Wael Abd-Almageed. Partner-assisted learning for few-shot image classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10573–10582, 2021.

[28] Rongkai Ma, Pengfei Fang, Gil Avraham, Yan Zuo, Tom Drummond, and Mehrtash Harandi. Learning instance and task-aware dynamic kernels for few shot learning. *arXiv preprint arXiv:2112.03494*, 2021.

[29] Rongkai Ma, Pengfei Fang, Tom Drummond, and Mehrtash Harandi. Adaptive poincaré point to set distance for few-shot classification. *arXiv preprint arXiv:2112.01719*, 2021.

[30] Puneet Mangla, Nupur Kumari, Abhishek Sinha, Mayank Singh, Balaji Krishnamurthy, and Vineeth N Balasubramanian. Charting the right manifold: Manifold mixup for few-shot learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2218–2227, 2020.

[31] Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*, 2018.

[32] Boris Oreshkin, Pau Rodríguez López, and Alexandre Lacoste. Tadam: Task dependent adaptive metric for improved few-shot learning. *Advances in Neural Information Processing Systems*, 31, 2018.

[33] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2536–2544, 2016.

[34] 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. *arXiv preprint arXiv:1803.00676*, 2018.

[35] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[36] Andrei A Rusu, Dushyant Rao, Jakub Sygnowski, Oriol Vinyals, Razvan Pascanu, Simon Osindero, and Raia Hadsell. Meta-learning with latent embedding optimization. In *International Conference on Learning Representations*, 2018.

[37] Christian Simon, Piotr Koniusz, Richard Nock, and Mehrtash Harandi. Adaptive subspaces for few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4136–4145, 2020.

[38] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. *Advances in Neural Information Processing Systems*, 30, 2017.

[39] Jong-Chyi Su, Subhransu Maji, and Bharath Hariharan. When does self-supervision improve few-shot learning? In *European Conference on Computer Vision*, pages 645–666. Springer, 2020.

[40] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1199–1208, 2018.

[41] Hao Tan, Jie Lei, Thomas Wolf, and Mohit Bansal. Vimpac: Video pre-training via masked token prediction and contrastive learning. *arXiv preprint arXiv:2106.11250*, 2021.

[42] 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 *European Conference on Computer Vision*, pages 266–282. Springer, 2020.

[43] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pages 10347–10357. PMLR, 2021.

[44] Zhengzhong Tu, Hossein Talebi, Han Zhang, Feng Yang, Peyman Milanfar, Alan Bovik, and Yinxiao Li. Maxvit: Multi-axis vision transformer. *arXiv preprint arXiv:2204.01697*, 2022.

[45] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008.

[46] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. *Advances in Neural Information Processing Systems*, 29:3630–3638, 2016.

[47] Davis Wertheimer, Luming Tang, and Bharath Hariharan. Few-shot classification with feature map reconstruction networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8012–8021, June 2021.

[48] Jiamin Wu, Tianzhu Zhang, Yongdong Zhang, and Feng Wu. Task-aware part mining network for few-shot learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8433–8442, 2021.

[49] Jiangtao Xie, Fei Long, Jiaming Lv, Qilong Wang, and Peihua Li. Joint distribution matters: Deep brownian distance covariance for few-shot classification. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[50] Chengming Xu, Yanwei Fu, Chen Liu, Chengjie Wang, Jilin Li, Feiyue Huang, Li Zhang, and Xiangyang Xue. Learning dynamic alignment via meta-filter for few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5182–5191, 2021.

[51] Han-Jia Ye, Hexiang Hu, De-Chuan Zhan, and Fei Sha. Few-shot learning via embedding adaptation with set-to-set functions. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages

8808–8817, 2020.

[52] Chi Zhang, Yujun Cai, Guosheng Lin, and Chunhua Shen. Deepemd: Few-shot image classification with differentiable earth mover's distance and structured classifiers. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[53] Chi Zhang, Henghui Ding, Guosheng Lin, Ruibo Li, Changhu Wang, and Chunhua Shen. Meta navigator: Search for a good adaptation policy for few-shot learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9435–9444, 2021.

[54] 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 *International Conference on Learning Representations*, 2020.

[55] Xueting Zhang, Debin Meng, Henry Gouk, and Timothy M Hospedales. Shallow bayesian meta learning for real-world few-shot recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 651–660, 2021.

[56] Jiabao Zhao, Yifan Yang, Xin Lin, Jing Yang, and Liang He. Looking wider for better adaptive representation in few-shot learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 10981–10989, 2021.

[57] Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. ibot: Image bert pre-training with online tokenizer. *International Conference on Learning Representations (ICLR)*, 2022.

[58] Ziqi Zhou, Xi Qiu, Jiangtao Xie, Jianan Wu, and Chi Zhang. Binocular mutual learning for improving few-shot classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8402–8411, 2021.

[59] Luisa Zintgraf, Kyriacos Shiarli, Vitaly Kurin, Katja Hofmann, and Shimon Whiteson. Fast context adaptation via meta-learning. In *International Conference on Machine Learning*, pages 7693–7702. PMLR, 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] See Section 3.5
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (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] See Section 3.1 and supplementary material
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Section 3.1 and supplementary material
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] See Section 3.1 and supplementary material
   (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] See Section 3.1
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] See Section 3.1 and supplementary material
   (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]