

---

# Rethinking Generalization in Few-Shot Classification

## Supplementary Material

---

Markus Hiller\*<sup>1</sup> Rongkai Ma\*<sup>2</sup> Mehrtash Harandi<sup>2</sup> Tom Drummond<sup>1</sup>

<sup>1</sup>School of Computing and Information Systems, The University of Melbourne

<sup>2</sup>Department of Electrical and Computer Systems Engineering, Monash University

markus.hiller@student.unimelb.edu.au

{rongkai.ma, mehrtash.harandi}@monash.edu

tom.drummond@unimelb.edu.au

### A Selecting helpful patches at inference time in 1-shot scenarios

Figure 6 in the main paper demonstrates that our approach is able to successfully learn at inference time which image regions should be considered to classify the unknown query images in a 5-way 5-shot scenario. We additionally present the visualization of the token importance weights for the query images of a 5-way 1-shot scenario in Figure A1. It can be clearly observed that the brighter regions representing higher importance of the respective image patches strongly relate to the actual objects that are to be classified, even in the case of smaller objects (2nd and 4th from the right). While our method only has access to significantly less information in the here presented 1-shot than in the case of 5-shot scenarios (see details in Section 2.4), our proposed way of masking the neighborhood of each pixel during the online optimization procedure still enables selection of the most helpful areas characteristic for the respective classes.

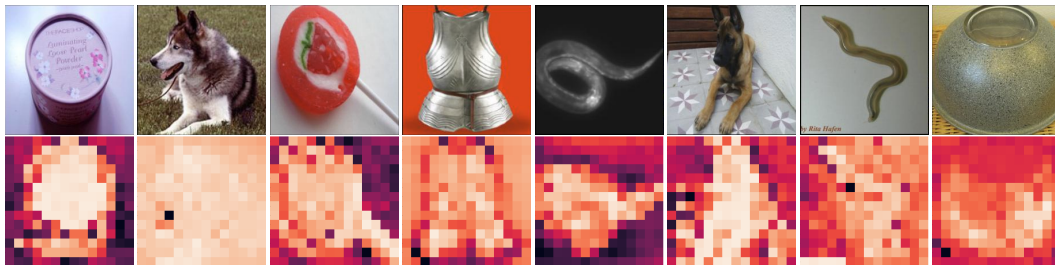


Figure A1: **Learning token importance at inference time.** Visualized importance weights learnt via online optimization for support set samples in a 5-way 1-shot task on the *miniImageNet* test set.

### B Discussion on model size and performance

Related works have shown that model size seems to not be a good indicator for few-shot performance, most likely since training datasets are comparably small (e.g. 38.4K images in *miniImageNet* [20] vs. standard ImageNet with 1.28M [16]) and big networks are thus much more prone to overfit. Chen *et al.* [2] demonstrate in Figure 3 of their paper that the performance gains due to larger backbones plateau across all methods for backbones bigger than ResNet10 in their experiments and only offer diminishing gains (if any at all). The investigations of Mangla *et al.* [11] yielded similar results,

---

\*Joint first authorship

where the performance on the *miniImageNet* and *tieredImageNet* datasets even decreased by around 0.5-1% when scaling up from ResNet18 to ResNet34 (Table 2). We thus conclude that increased number of parameters on its own does not lead to better few-shot performance, and the tendency of many recent works to choose the established ResNet12 (12.4M) over bigger backbones is highly likely a result of this.

To gauge the influence of model size in FewTURE, we additionally investigate the use of the significantly smaller ViT-tiny architecture with only 5M parameters [19]. Results in Table A1 show that our method achieves a competitive accuracy of 81.10% on the *miniImageNet* test dataset with less than one seventh of the number of parameters of a WRN-28-10, but is (in contrast to many other methods like e.g. [22]) able to leverage increased model sizes to further boost performance.

Table A1: Investigating model size and performance. Average classification accuracy on the *miniImageNet* test set, evaluated in a 5-way 5-shot scenario with a ViT-small backbone.

Method	Backbone	#Params	Test Accuracy
ProtoNet [18]	ResNet-12	≈ 12.4M	79.46±0.48
FEAT [22]	ResNet-12	≈ 12.4M	82.05±0.14
DeepEMD [23]	ResNet-12	≈ 12.4M	82.41±0.56
COSOC [10]	ResNet-12	≈ 12.4M	85.16±0.42
Meta DeepBDC [21]	ResNet-12	≈ 12.4M	84.46±0.28
LEO [17]	WRN-28-10	≈ 36.5M	77.59±0.12
CC+rot [7]	WRN-28-10	≈ 36.5M	79.87±0.33
FEAT [22]	WRN-28-10	≈ 36.5M	81.11±0.14
PSST [4]	WRN-28-10	≈ 36.5M	80.64±0.32
MetaQDA [24]	WRN-28-10	≈ 36.5M	84.28±0.69
OM [13]	WRN-28-10	≈ 36.5M	85.29±0.41
FewTURE (ours)	ViT-Tiny	≈ 5.0M	81.10±0.61
FewTURE (ours)	ViT-Small	≈ 22.0M	84.51±0.53
FewTURE (ours)	Swin-Tiny	≈ 29.0M	<b>86.38±0.49</b>

## C Discussion on self-supervised vs. supervised pretraining

**Performance in few-shot learning.** We demonstrate in Figure 4 of the main paper that self-supervised pretraining with masked image modelling as pretext task provides a significant advantage over supervised pretraining for our approach – a finding that differs from prior non-few-shot literature where self-supervised methods only moderately outperform their supervised counterparts [25] or even perform worse in some cases [3]. We provide our interpretation and insights regarding this in the following.

Few-shot classification is distinctively different from ‘conventional’ classification (like investigated in [3]) in one important aspect: novel previously unseen classes are encountered at test time. As such, supervised learning induces a tendency of the representation space to overfit to the structure of the classes observed during training. In other words, the representation space is created and condensed to easily separate observed training classes, but at the expense of distorting other dimensions that might be crucial to correctly distinguish yet unseen classes. This is known in the few-shot literature as ‘*supervision collapse*’ [5]. Since no class labels are provided during the self-supervised pretraining, we expect the method to create a more general/less distorted representation space that is significantly better suited to generalize to yet unseen classes and avoid collapse. These intuitions are supported by the results we have obtained (Fig 4.). We further observe that self-supervised training is helpful to prevent early overfitting when learning from small few-shot datasets (e.g. 38.4K *miniImageNet* [20] vs. 1.2M ImageNet1K [16]).

**Training details of supervised pretraining.** For adequate comparison to related work in few-shot learning, we follow the widely adopted pretraining scheme used in FEAT [22] and other works (e.g.

DeepEMD [23]) for our supervised pretraining. In detail, we train the network with a cross-entropy loss on the training set of the respective dataset to solve a standard classification task (e.g. for *miniImageNet*: 64 classes) – i.e., using the exact same data we use for self-supervised pretraining. Like [22] we use the representations of the penultimate layer (before the classifier) to evaluate the performance and quality of the embeddings. To judge suitability of the encoder for few-shot tasks, an N-way 1-shot task is commonly solved (e.g. N=16 for *miniImageNet* due to the 16 classes in the validation set) – and we tried three different variants here:

1. & 2. One sample per class is encoded to produce a class-embedding (‘prototype’), and classification performance is evaluated using 15 queries per class (as used in recent related works). To retrieve one embedding per sample, we use the average over all patch tokens produced by the Transformer architecture. For fairness regarding metrics, we evaluate both:
  1. embedding distance (MSE) and
  2. embedding similarity (cosine) to perform classification.
3. We additionally use our own patch-based classifier to evaluate the few-shot setting using all patch embeddings (as we later do during fine-tuning & evaluation).

We perform validation over 200 such few-shot tasks after every epoch during training and pick the best-performing model regarding highest average validation accuracy. We encountered clear signs of overfitting during this type of supervised training, with the training accuracy consistently improving to convergence, but validation accuracy plateauing (or decreasing) rather early on (~350-500ep), independent of the variant we used to evaluate on the validation set.

## D Ablation studies on components of FewTURE

In this section, we provide further insights into our approach and the design choices we made.

### D.1 Ablation on inner loop token reweighting

A more detailed version of the average classification test accuracies achieved with a meta fine-tuned ViT backbone on the *miniImageNet* dataset used for the visualization of the contribution for different numbers of token reweighting steps during online optimisation (main paper, Figure 7) is presented in Table A2, including the respective 95% confidence intervals. As discussed in the main paper, we observed a strong initial increase of 1.15% when using our proposed adaptation via online optimization (steps > 0). While a higher number of inner-loop updates seems to still lead to increased accuracy across all our test runs, this benefit brings along higher computational cost as can be seen in the second row of Table A2. We generally found settings between 5 and 15 steps to be a good accuracy vs. inference-time trade-off. Our experiments were conducted using an Nvidia-2080ti GPU and the stated inference times have been averaged over 1800 query sample classifications. It is to be noted that the code has not been specifically optimized for fast inference times, and these values should rather be interpreted in a relative manner.

Table A2: Average classification accuracy and inference times on the *mini-ImageNet* test set for varying inner loop optimization steps, evaluated in a 5-way 5-shot scenario with a ViT-small backbone and SDG with 0.1 as learning rate. Experiments were conducted using an Nvidia-2080ti and runtimes were averaged over 1800 query sample classifications.

	0 steps	5 steps	10 steps	15 steps	20 steps
<b>Accuracy</b>	82.68±0.59	83.83±0.59	83.89±0.57	84.05±0.55	84.51±0.53
<b>Inference time [ms]</b>	156.86±2.16	159.86±2.12	162.11±2.11	165.62±2.06	168.62±2.22

### D.2 Ablation on token aggregation and similarity metrics

As discussed in the main paper, we use the *logsumexp* operation to aggregate our similarity logits as it poses a rigorous and numerically stable way of combining individual class probabilities (one for each token) to a valid overall probability distribution over classes for each image, independent of how the individual token (log) probability scores are obtained. Table A3 (a) shows the results of additional

experiments (training and testing) using our method (ViT-small) and 15 token reweighting steps with the only change being aggregation of the logits via *mean*, and we found it to underperform our chosen *logsumexp* method of aggregation. Direct addition without normalization (i.e. just summing up all logits) proved unstable due to large logit values and was thus not included in this table.

We further investigated the use of alternate metrics to compute the similarity between different tokens. Both the use of the negative Euclidean distance and unscaled dot-product yielded inferior results compared to the temperature-scaled cosine distance we use in FewTURE (Table A3 (b)).

Table A3: Ablation on token aggregation method and similarity metric. Reported are the average classification accuracies on the *miniImageNet* test set evaluated in a 5-way 5-shot scenario with a ViT-small backbone.

(a) Token aggregation		(b) Similarity metrics	
Aggregation method	Test Accuracy	Metric	Test Accuracy
logsumexp	<b>84.05 ± 0.53</b>	cosine similarity	<b>84.05 ± 0.53</b>
mean logits	80.13 ± 0.60	neg. Euclidean dist.	81.85 ± 0.58
		unscaled dot-prod.	37.60 ± 0.64

### D.3 Ablation regarding temperature scaling of embedding similarity logits

As reported in the main paper, we use the temperature  $\tau_S$  to rescale the logits of our task-specific similarity matrix  $\tilde{S}$  via division (or the original similarity matrix  $S$  in case no task-specific adaptation shall be used). We investigate two different ways of temperature scaling: (i) the possibility of using a fixed temperature defined as  $1/\sqrt{d}$  where  $d$  is the dimension of the patch embeddings of the respective architecture, and (ii) learning the appropriate temperature during the meta fine-tuning procedure. In practice, we learn  $\log(\tau_S)$  to ensure  $\tau_S \geq 0$ .

We observe throughout our 1-shot experiments depicted in Figure A2 (a) and (b) that the temperature converges towards our default values of  $1/\sqrt{d}$  shown as a dashed horizontal line. This is independent of the initial value of the temperature parameter  $\tau_S^{\text{init}}$ . For the 5-way 5-shot experiments presented in Figure A2 (c) and (d) however, we observe that while our default value still achieves good results, the learned temperature converges to a slightly lower value across all experiments.

### D.4 Development over the course of pretraining

We further present insights into the development of the accuracy during self-supervised pretraining. Since our pretraining procedure is entirely unsupervised and does hence not include any labels, we investigate models trained for a variety of different epochs and evaluate these on the test set using the proposed similarity-based classification method *with* ('5 steps' and '15 steps') and *without* ('None') and present the results in Table A4. Note that no meta fine-tuning was employed here. We observe that while the performance significantly increases over the first 50 epochs, there seems to be some saturation and even slight decrease in performance until above 500 epochs where the accuracy increases again and (mostly) achieves highest results in this study.

Table A4: **Development of test accuracy in self-supervised pretraining.** Results obtained for a 5-way 5-shot scenario on the *tieredImageNet* test set using our proposed classifier with a ViT-small backbone. For online optimisation (i.e., steps > 0), we use SGD with 0.1 as learning rate.

Reweighting steps	Epochs					
	1	50	100	250	500	800
None	39.20±0.69	73.30±0.75	73.63±0.73	72.84±0.72	71.51±0.72	73.83±0.74
5 steps	39.34±0.69	73.59±0.74	74.03±0.73	73.10±0.73	71.82±0.72	74.16±0.73
15 steps	39.43±0.69	73.86±0.73	74.48±0.74	73.41±0.75	72.16±0.73	74.42±0.74

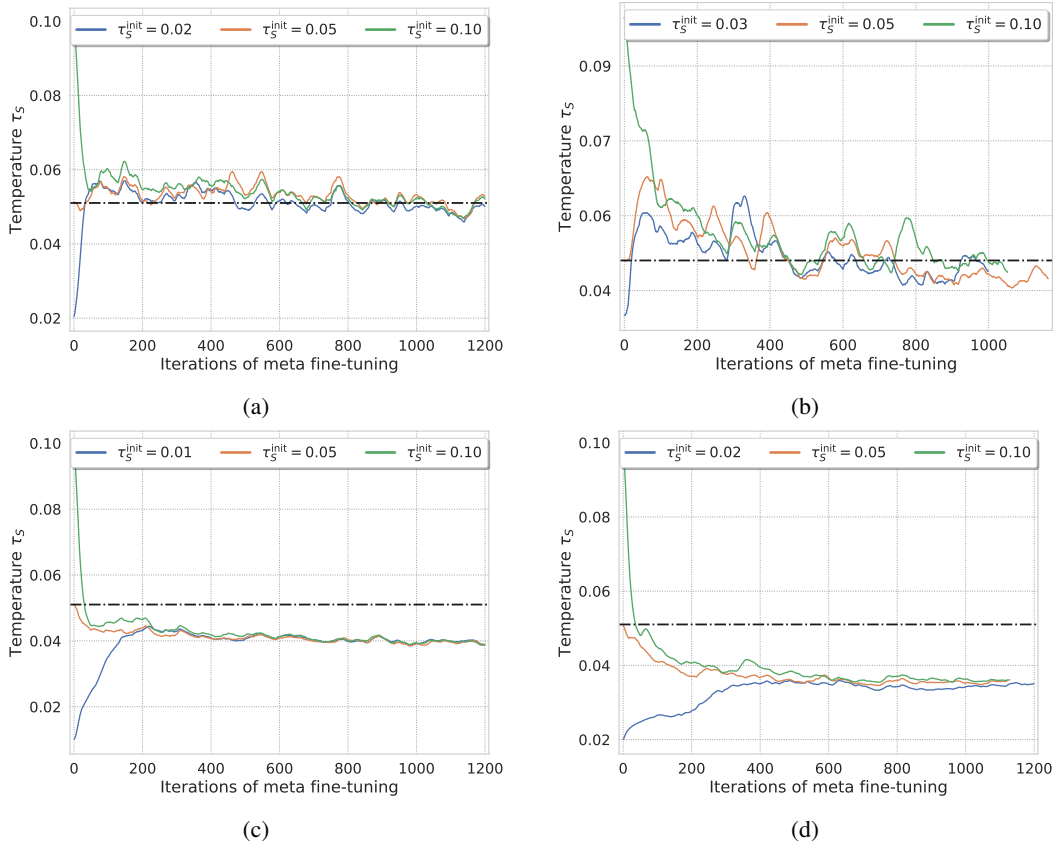


Figure A2: **Temperature for rescaling similarity logits.** (a) and (b) show the learned temperatures for 5-way 1-shot scenarios on *miniImageNet* and *tieredImageNet*, respectively. The corresponding 5-way 5-shot results are depicted in (c) and (d). All experiments have been conducted using a ViT-small architecture.

## E Further visualization of instance embeddings

Figure 5 in the main paper depicts instance and class embeddings visualized via PCA projection to the three dominant dimensions. Figure A3 additionally depicts a comparison of projected views of the tokens of 5 instances from a novel class in embedding space for different ways of meta training. While the representations obtained from the network meta fine-tuned by using common averaging over the embeddings (*‘average’*) do not exhibit any clear separation of the instances, the embeddings obtained with our classifier seem to retain the instance information (*‘w/o v’*) and separation is improved when using token importance reweighting (*‘w/ 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 further prevents the network from supervision collapse and helps to achieve the higher performance observed on the benchmarks.

## F Datasets used for evaluation

We train and evaluate our approach presented in the main paper on the following few-shot image classification datasets:

***miniImageNet*.** The *miniImageNet* dataset has been initially proposed by [20] with follow-up modifications by [14] and consists of a specific 100 class subset of ImageNet [16] with 600 images for each class. The data is split into 64 training, 16 validation and 20 test classes.

***tieredImageNet*.** Similar to the previous dataset, the *tieredImageNet* [15] is a subset of classes selected from the bigger ImageNet [16] dataset, however with a substantially larger set of classes and

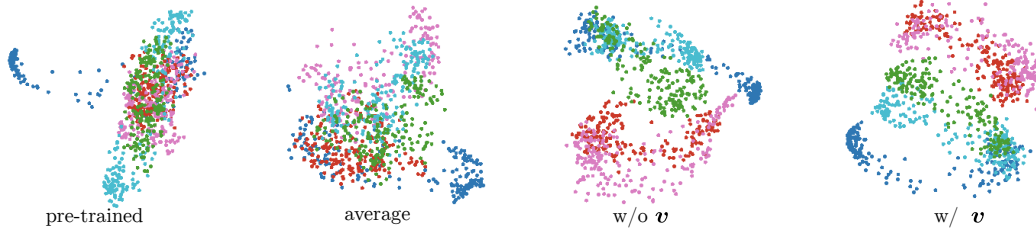


Figure A3: **Instance embeddings after meta fine-tuning.** Visualized are the projected tokens of 5 instances of the same novel support set class for different meta fine-tuning (M-FT) methods (after self-supervised pretraining). From left to right: self-supervised pretraining only, M-FT using an average 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 to main dimensions.)

different structure in mind. It comprises a selection of 34 super-classes with a total of 608 categories, totalling in 779,165 images that are split into 20,6 and 8 super-classes to achieve better separation between training, validation and testing, respectively.

**CIFAR-FS.** The CIFAR-FS dataset [1] contains the 100 categories with 600 images per category from the CIFAR100 [8] dataset which are split into 64 training, 16 validation and 20 test classes.

**FC-100.** The FC-100 dataset [12] is also derived from CIFAR100 [8] but follows a splitting strategy similar to *tieredImageNet* to increase difficulty through higher separation, resulting in 60 training, 20 validation and 20 test classes.

## G Implementation details

We present further details regarding our implementation and used hyperparameters in the following.

### G.1 Pretraining

**GPU usage.** We pretrain our models with the use of 4 Nvidia A100 GPUs with 40GB each for our ViT [6, 19] and 8 such GPUs for our Swin [9] variants.

**Hyperparameter choice.** We follow the strategy introduced by [25] to pretrain our Transformer backbones and mostly stick to the hyperparameter settings reported in their work. We generally use two global crops and 10 local crops with crop scales of (0.4, 1.0) and (0.05, 0.4), respectively. We further use a patch size of 16 for our ViT models and a window size of 7 for Swin, corresponding to the default sizes for ViT-small [6, 19] and Swin-tiny [9]. We use an output dimension of 8192 for the projection heads across all models, and employ random Masked Image Modelling with prediction ratios (0, 0.3) and variances (0, 0.2). Our ViT and Swin architectures are trained with an image size of  $224 \times 224$  arranged in batches of size 512 samples for 1600 and 800 epochs, respectively, using a linearly ramped-up learning rate (over first 10 epochs) of  $5e-4 \times \text{batchsize}/256$ . For detailed information, we would like to refer the interested reader to the work by Zhou *et al.* [25] where more background information regarding the influence and justification of these hyperparameters is provided.

### G.2 Meta fine-tuning

**GPU usage.** During the meta fine-tuning (M-FT) stage, we use 1 and 2 Nvidia 2080-ti GPUs for ViT-small and Swin-tiny, respectively, across all 4 datasets.

**Hyperparameters.** We fix the input image size as  $224 \times 224$  for all datasets. We use the SGD optimizer along with a learning rate of  $2e-4$ , 0.9 as the momentum value and  $5e-4$  as the weight decay. Additionally, we employ a learning rate scheduler with cosine annealing for 5,000 iterations as one cycle, ramping down to  $5e-5$  at the end of each cycle.

**Online optimization.** During the online learning of the token importance reweighting vectors, we adopt the SGD optimizer with 0.1 as the learning rate. For online update steps, we generally choose

a default value of 15 steps across all datasets. For further details regarding the temperature scaling procedure used to rescale our task-specific similarity logits, please refer to Section D.3.

## References

- [1] Luca Bertinetto, Joao F Henriques, Philip Torr, and Andrea Vedaldi. Meta-learning with differentiable closed-form solvers. In *International Conference on Learning Representations*, 2019.
- [2] Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Wang, and Jia-Bin Huang. A closer look at few-shot classification. In *International Conference on Learning Representations*, 2019.
- [3] 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.
- [4] Zhengyu Chen, Jixie Ge, Heshen Zhan, Siteng Huang, and Donglin Wang. Pareto self-supervised training for few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13663–13672, 2021.
- [5] Carl Doersch, Ankush Gupta, and Andrew Zisserman. Crosstransformers: spatially-aware few-shot transfer. *Advances in Neural Information Processing Systems*, 33:21981–21993, 2020.
- [6] 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.
- [7] 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.
- [8] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.
- [9] 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.
- [10] 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.
- [11] 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.
- [12] 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.
- [13] Guodong Qi, Huimin Yu, Zhaohui Lu, and Shuzhao Li. Transductive few-shot classification on the oblique manifold. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8412–8422, 2021.
- [14] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. In *International Conference on Learning Representations*, 2017.
- [15] 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.
- [16] 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.
- [17] 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.
- [18] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. *Advances in Neural Information Processing Systems*, 30, 2017.
- [19] 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.
- [20] 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.
- [21] 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.
- [22] 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.
- [23] 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)*, 2020.
- [24] 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.
- [25] 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.