BaseTransformers: Attention over base data-points for One Shot Learning

Mayug Maniparambil[1,2]
mayugmaniparambil@gmail.com
Kevin McGuinness[1,2]
kevin.mcguinness@dcu.ie
Noel E. O'Connor[1,2]
noel.oconnor@dcu.ie

[1] ML Labs
SFI Centre for Research Training, Dublin, Ireland
[2] Dublin City University, Dublin, Ireland

Abstract

Few shot classification aims to learn to recognize novel categories using only limited samples per category. Most current few shot methods use a base dataset rich in labeled examples to train an encoder that is used for obtaining representations of support instances for novel classes. Since the test instances are from a distribution different to the base distribution, their feature representations are of poor quality, degrading performance. In this paper we propose to make use of the well-trained feature representations of the base dataset that are closest to each support instance to improve its representation during meta-test time. To this end, we propose BaseTransformers, that attends to the most relevant regions of the base dataset feature space and improves support instance representations. Experiments on three benchmark data sets show that our method works well for several backbones and achieves state-of-the-art results in the inductive one shot setting. Code is available at github.com/mayug/BaseTransformers .

1 Introduction

The development of few shot learning models is important for real world deployment of artificial vision systems outside of controlled scenarios. Most previous works focus on developing stronger models, while scant attention has been paid to the properties of the data itself and the fact that as the number of data points increase, the ground truth distribution can be better uncovered. Estimating the prototype for a novel class using a single instance is fundamentally ill posed, resulting in poor one shot performance. [39] has shown that this can be alleviated by modeling the class conditional distribution as a Gaussian and sampling a large number of features from this distribution to train a classifier or estimate the prototype. They show that distributions of semantically similar classes in the base dataset have similar mean and variance to the distributions of the novel class. Therefore, the statistics of the class conditional distributions of novel classes are transferred from those of base classes which have been estimated with several examples (over 600) per class. This method assumes that the class conditional feature distributions are uni-modal Gaussian and that the transferable statistics are only global and not local to each base instance or its spatial locations.

© 2022. The copyright of this document resides with its authors.
It may be distributed unchanged freely in print or electronic forms.

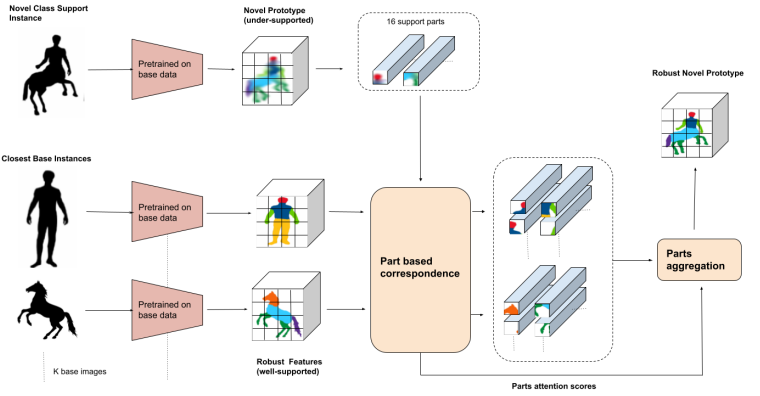

Figure 1: BaseTransformers construct robust novel class prototypes by attending to and aggregating semantically similar regions of the well supported base data feature space instead of using the noisy novel prototype as in Prototypical Networks [28].

We propose a novel method for estimating prototypes of unseen classes using the base dataset without making any assumptions on the distribution of the base data feature space or the transferability of the instance level or spatial level information. Our proposed method, BaseTransformers, is an end-to-end learnable cross attention mechanism that estimates a robust, base aligned prototype for novel categories by learning local part based correspondences between the support instance and semantically similar base instances. This is based on two key ideas: (i) the base dataset images are composed of semantically meaningful parts that could be reused during the classification of novel images; and (ii) since the base data features are estimated using many shots, the features corresponding to these parts are less noisy representations, closer to the ground truth distribution. The concept is illustrated in Fig 1, where a novel 'centaur' class has an undersupported prototype in the feature space of an encoder pretrained on base-data. However a robust prototype of a centaur can be constructed by taking the head, torso of a human and the body and legs of horse base classes which are individually well supported in the feature space.

We hypothesize that semantically similar parts of a well represented base data feature space can be used to estimate a novel prototype that is effectively a part based composition of the well estimated base data regions. To enable this BaseTransformers allow for: (i) spatial part based comparison between the support instance and similar base instances to select the semantically meaningful regions of the robust base data feature space; and (ii) aggregation of the semantically similar parts of the base instances to estimate a novel prototype that is a composition of robust meaningful base regions. Taking inspiration from [6] we instantiate a cross attention mechanism on the feature space of the pretrained encoder to enable this. We perform this adaptation of the support instance using the base instances in the feature space and not the original pixel space, as the feature space has lower dimensions and semantically meaningful structures that are more easily transferable between the base and novel domains.

For each episode, the BaseTransformer takes the 2D feature spaces of the support instance as query and the closest base instances as the key and value. The BaseTransformer is trained end-to-end using the meta learning paradigm to identify the most relevant regions in the base data feature space and use them to compose a more robust novel class prototype.

Our approach starts with a pretraining stage using cross entropy and contrastive losses on the base dataset to produce a robust encoder bypassing supervision collapse [6, 19]. This is followed by a meta training stage, in which the encoder and BaseTransformer are jointly trained to adapt the support instances using instances from the closest base classes. To identify the closest base classes we propose using the class label information of the support instances, and making queries on the base dataset based on semantic similarity. We show that the proposed method beats the current state-of-the-art in 3 different datasets (70.88%, 72.46%, 82.27% on mini-ImageNet, tiered-ImageNet and CUB respectively) in the inductive one shot setting.

Our novel contributions are: i) We identify that robust novel prototypes in one shot learning can be obtained by part based composition of semantically similar base features; ii) We design BaseTransformer that improves the 1-shot prototypes by learning to attend to the robust 2D feature space of base instances and aggregate these to compose the novel prototype; iii) We evaluate our method on two backbones and three benchmarks to show its effectiveness in the one shot inductive setting of few shot learning.

2 Related Work

Meta Learning aims to extract common useful knowledge for classifying novel classes by emulating few shot tasks during training time, and are usually optimization based or metric learning based. In optimization based methods, the objective is to meta-learn a good initialization of weights [8, 24, 26, 45] or the optimization process [16, 21, 25, 38] or a combination of both [2, 23]. In metric learning methods [28, 30, 33, 41] the objective is to develop an embedding space where similar instances are close to each other in some distance sense so that a simple nearest neighbour classifier can be used during meta test time. Our method is similar to metric learning, specifically prototypical networks, as we only have an extra transformer stage to adapt the support instances to form more robust prototypes.

Transfer Learning methods train a network to classify base classes, followed by finetuning the classifier on the novel instances whilst keeping the encoder fixed. [4, 34] has shown that this simple strategy performs surprisingly well, beating/matching several complex meta learning algorithms. We follow works such as [40] and have a pretraining stage in which the encoder is trained on a combination of cross entropy and self supervised loss. Other works [9, 17, 19, 29] have shown that addition of self supervision losses in the pretraining stage provides more robust features, resulting in improved few shot performance. We use the InfoNCE loss [3] as an auxiliary loss during the pretraining stage.

A *Base Dataset* has been used explicitly during meta test time in previous works, such as in [1, 39]. The approach of [39] models the feature space of each class as a Gaussian and transfers statistics from well estimated base class distributions to novel class distributions, and sample from this to train a classifier. In our approach, we do not assume that the class feature space follows a Gaussian distribution, but use a parametric function- a transformer to improve the prototype representation by means of attention over the feature space of base examples. The approach reported in [1] aligns the feature space of the novel instances to that of the closest base instances by reducing an adversarial alignment loss during the test time, while we do not tune any parameters of the transformer network during meta test time. Both methods make use of cosine similarity in the feature space to query the closest base classes. While this works well for us for shallow encoders, we find that making use of semantic information from the class labels results in semantically closer base classes.

Transformers have also been investigated in a similar context. Previous works like [11, 40] make use of transformer based adaptation on the feature space to improve few shot performance. The approach in [40] uses self attention over the prototypes to adapt them in a task specific manner, while the approach of [6] builds a classifier that aligns the prototypes and the queries spatially. Similarly [10, 12, 15, 36] use different forms of self-correlation and cross-correlation mechanisms to improve the relational comparison between the prototypes and the query instances. We differ from these methods, in that we explicitly attend over all spatial locations of a base data subset to improve the support instance features. To our knowledge, our work is the first to apply attention over the base data points for few shot learning.

3 Method

In this section we first introduce the setup of few shot classification in section 3.1 followed by description of our proposed method in sections 3.2 through 3.4.

3.1 Preliminaries

We follow the inductive setting for few shot learning. A few shot task is an N way M shot classification problem, with N classes sampled from novel classes C_n with M examples per class. $D_s = \{x_i, y_i\}_{i=1}^{M\times N}$ refers to the support set sampled from novel classes C_n. Test instances x_q are sampled from a a query set $D_q = \{x_i, y_i\}_{i=1}^{Q}$ and the goal is to find a function f that classifies x_q via $\hat{y} = f(x_q \mid D_s)$. In the few shot learning literature M is usually 1 or 5 referring to the 1-shot or 5-shot task.

Finding f from the very few examples in the support set is very difficult, so a base dataset is provided consisting of base classes C_b such that $C_b \cap C_n = \emptyset$. In the meta-learning paradigm, f is learnt by sampling several N-way M-shot tasks D_s^b and corresponding query sets D_q^b from the base dataset to emulate the test time scenario. In each sampled task, f is learnt to minimize the average error on D_q^b:

$$f^* = \arg\min_f \sum_{(x_q^b, y_q^b)\in D_q^b} \ell\left(f(x_q^b \mid D_s^b), y_q^b\right), \tag{1}$$

where ℓ can be any loss that measures the discrepancy between prediction and true label.

During meta test time the optimal f^* is applied on tasks sampled from C_n. The performance of the model is evaluated on multiple tasks sampled from the novel classes C_n. For example, in prototypical networks, f consists of an embedding network E and a nearest neighbour classifier:

$$\phi_x = E(x) \in \mathbb{R}^d, \qquad \hat{y}_q = f(\phi_{x_q}; \{\phi_{x_s}^c\}), \tag{2}$$

where $\{\phi_{x_s}^c\}$ is the set of prototypes. Here, each prototype is given by:

$$\phi_{x_s}^c = \sum_{y_i \in c} E(x_i), \qquad (x_i, y_i) \in D_s. \tag{3}$$

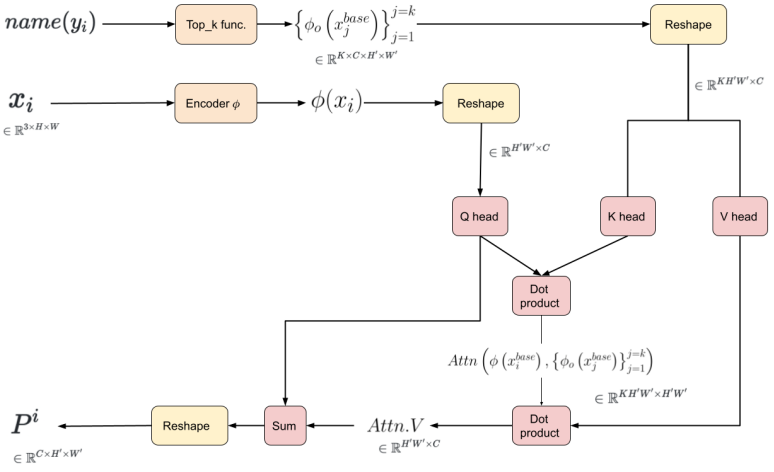

Figure 2: Support instance feature $\phi(x_i)$ is reshaped and projected by query head Q to obtain queries q_m^i where m corresponds to spatial locations in the support instance. q_m^i is then compared with the keys k_n^j from all spatial locations n of base instances to get attention scores $attn_{mjn}^i$, which are used to aggregate the values v_n^j and summed with original support feature $\phi(x_i)$ to obtain the base adapted prototype.

3.2 BaseTransformer

Given a support instance x_i and its closest base instances $\{x_i^{base}\}_{i=1}^k$ the BaseTransformer aims to learn a representation that enables part-based adaptation of x_i by attending over all the spatial locations of all base instances in $\{x_i^{base}\}_{i=1}^k$.

First, an image representation of support instance $\phi(x_i)$ is obtained using the encoder ϕ, while the class name corresponding to the support instance is used to get the k closest instances in the base dataset. The top k function is described in detail in Section 3.3. The features of the closest base instances are passed through a fixed encoder ϕ_0 whose weights are the weights obtained after the pre-training stage on the base dataset. These representations are then used by the Transformer to establish correspondences between support instances and base instances to produce the adapted prototype. Finally, similar to prototypical networks, the Euclidean distance is used to classify the query feature $\phi(x_{test})$ by making use of adapted prototypes $\{P_i\}_{i=1}^N$. Prototypical networks use 1D feature embedding while, BaseTransformers use 2D embeddings as input to allow the model to make part based soft correspondences between support and base instances, and use these to weigh the most relevant regions of base instances to estimate the prototype of a support instance as a composition of robust base parts.

More concretely, we consider a CNN without the final fully connected or pooling layers, such that $\phi(x_i) \in \mathbb{R}^{C\times H'\times W'}$. Top k function uses the pre-trained encoder phi_0 to provide the closest base instances features set $\{\phi_0(x_i^{base})\}_{i=1}^k$ where $\phi_0(x_i^{base}) \in \mathbb{R}^{C\times H'\times W'}$. During meta training care is taken so as to exclude the class of the support feature itself from this set of base features so as to force the BaseTransformer to learn to compose novel prototypes using only instances from different classes. These features are reshaped such that the attention

would be between spatial locations of $\phi(x_i)$ and spatial locations of the $\phi_0\left(x_i^{base}\right)$. Key-value pairs of base instance features $K\phi_0\left(x_i^{base}\right)$, $V\phi_0\left(x_i^{base}\right)$ are obtained using two independent linear layers K, V while the transformer's queries $Q\phi(x_i)$ are obtained by using linear mapping Q on the support instance features. Here, we distinguish between a query (or test) set sample and the query of the transformer by explicitly referring to the latter as transformer's query. The dot product between transformer's query and key features results in an attention map between support features and base features. This is followed by a softmax over all spatial locations and k base instances. The computed attention is then used to aggregate the values and a residual connection from the transformer's query features is added to obtain the adapted prototype. Figure 2 illustrates this process.

We follow the mathematical notation outlined in [6]. Let $q_m^i = Q\phi(x_{im})$ be the transformer queries i.e., the support features projected by Q, where i is the index of the support instance and m is the spatial location and $k_n^j = K\phi_0(x_{jn}^{base})$ are the key features, i.e., the base features projected by K where j is the index of the base instance and n is the index of the spatial location. An attention map $\widetilde{\text{attn}}$ between support features and base features is calculated as:

$$\widetilde{\text{attn}}^i_{mjn} = \frac{\exp(\text{attn}^i_{mjn})}{\sum_{mjn}\exp(\text{attn}^i_{mjn})}, \quad \text{where} \quad \text{attn}^i_{mjn} = \langle k_n^j, q_m^i \rangle. \tag{4}$$

Next the base adapted prototype P_m^i at spatial location m is obtained as follows:

$$P_m^i = q_m^i + \sum_{jn} \langle \widetilde{\text{attn}}^i_{mjn}, v_n^j \rangle. \tag{5}$$

For a test instance x_{tm}^{test}, logits are obtained by calculating the similarity and averaging over the spatial and channel locations as,

$$\text{sim}(\phi(x_t^{test}), p^i) = -\frac{1}{H'W'}\sum_m \left\| \phi(x_{tm}^{test}) - P_m^i \right\|_2^2. \tag{6}$$

Here we do not update the features of the base instances during training so as to not corrupt the base data features that have been learnt using several examples per class. The features of a random subset of base instances are computed using the pretrained encoder f_0 and stored in a memory bank, which is then queried by the top-k querying function described in Section 3.3.

3.3 Querying function

We use a semantic similarity based querying function, which uses the label name of the support instance and finds the 5 closest base classes in a semantic space that varies according the dataset. Then base instances are sampled randomly from these classes such that they sum up to k. For mini-Imagenet dataset the semantic similarity is equal to the LCH-similarity[13] of the labels in the WordNet graph[20]. LCH similarity between class labels do not work well for tiered-ImageNet because the class splits were made using higher up nodes in the WordNet hierarchy resulting in very similar LCH similarity scores between a test class label and many base class labels. Hence, we use BERT[8] embeddings of the word labels concatenated with their hypernyms from WordNet to find more semantically similar base classes. For CUB, category-level attributes describing the visual features of each bird species are already available. Similar to [27], we use the cosine similarity between normalized category attribute vectors to query the closest base classes.

3.4 Training

Following [6, 17, 19] we note that we require base embeddings that contain more information than just information regarding base classes to be effective for adapting novel classes. To restrict supervision collapse, we train our encoder with an auxiliary contrastive loss in the pretraining stage. We follow a version of InfoNCE loss from [4], where the distance measure is Euclidean instead of cosine distance.

$$l(i,j) = -\log\left(\frac{\exp(s_{i,j})}{\sum_{k=1}^{2N} 1_{k!=i}\exp(s_{i,k}))}\right), \tag{7}$$

$$L_{\text{infoNCE}} = \frac{1}{2N}\sum_{k=1}^{N}\left[l(2k-1,2k) + l(2k,2k-1)\right], \tag{8}$$

where $s_{i,j} = -\left\|f_i - f_j\right\|_2^2$ and f_i, f_j are features of SimCLR [3] style augmented images in a minibatch. Concretely, the pretraining is a N_b way classification task where N_b is the number of classes in the base dataset. It is evaluated on a 16-way 1-shot classification task on the validation set. The complete pretraining objective is:

$$L_{\text{pretraining}} = L_{\text{classification}} + b \times L_{\text{InfoNCE}}, \tag{9}$$

where b is a hyperparameter balancing the auxiliary loss and $L_{\text{classification}}$ is a N_b way cross-entropy loss.

After pretraining, we train the transformer and the encoder end to end in a meta-learning fashion similar to [40]. Because the feature encoder is pretrained on base dataset, a lower learning rate(factor of 10) is used for the feature encoder to ensure convergence. Similar to the pretraining stage we use unsupervised InfoNCE loss as an auxiliary loss along with the cross entropy loss during meta training stage to restrict supervision collapse.

4 Experiments

We evaluate our method on three different datasets, namely mini-Imagenet, tiered-Imagenet and CUB [35]. Mini-Imagenet and tiered-Imagenet are subsets of the Imagenet dataset designed specifically for few shot learning. Mini-Imagenet dataset consists of 60,000 images across 100 classes of which train, validation, and test have 64, 16, and 20 classes respectively. We follow the split specified in [25] with 64 classes in the base dataset. Tiered-Imagenet is a larger dataset consisting of 351, 97, and 160 categories for model training, validation, and evaluation, respectively. We follow the split specified in [40]. In addition to this, we also look at a more fine grained few shot classification task using the CUB dataset that consists of images of various species of birds. CUB dataset contains 11,788 images split into 100, 50, and 50 classes for train, validation, and test. For all images in CUB dataset, we use the provided bounding box to crop all the images as a preprocessing step [31]. We follow the split specified in [40]. Similar to [26, 40], we use 10,000 randomly sampled few shot tasks for testing as well as report the average accuracy and 95% confidence intervals.

5 Implementation details

We test our method with two networks popularly used in the few shot learning literature, namely Conv4-64 – a 4 layer convolution network [28, 31, 33, 40] and ResNet-12 – a 12-

Table 1: 5-way 1-shot and 5-way 5-shot classification accuracy (%) on miniImageNet dataset using ResNet-12 and Conv4-64 backbones. 95% confidence intervals reported. The numbers in bold are the best performing methods for the corresponding setting.

Setups	1-shot		5-shot	
Backbone	Conv4-64	Res12	Conv4-64	Res12
ProtoNets[28]	49.42±0.78	60.37±0.83	68.20±0.66	78.02±0.57
SimpleShot[34]	49.69±0.19	62.85±0.20	66.92±0.17	80.02±0.14
CAN[11]	-	63.85±0.48	-	79.44±0.34
FEAT[40]	55.15±0.20	66.78±0.20	71.61±0.16	82.05±0.14
DeepEMD[42]	-	65.91±0.82		82.41±0.56
IEPT[43]	56.26±0.45	67.05±0.44	**73.91**±0.34	82.90±0.30
MELR[2]	55.35±0.43	67.40±0.43	72.27±0.35	83.40±0.28
InfoPatch[17]	-	67.67±0.45	-	82.44±0.31
DMF[37]	-	67.76±0.46	-	82.71±0.31
META-QDA[44]	56.41±0.80	65.12±0.66	72.64±0.62	80.98±0.75
PAL[18]	-	69.37±0.64	-	**84.40**±0.44
BaseTransformer	**59.37**±0.19	**70.88**±0.17	73.40±0.18	82.37±0.19

layer residual network [14, 40]. As mentioned above we have an additional pretraining stage over the base dataset before the meta training stage. We use images resized to input resolution of 84×84 for both networks.

In pretraining stage, we use SGD with momentum with an initial learning rate of 0.1 which is decayed by 0.1 using a custom schedule for both networks, similar to [40]. For weighing the auxiliary contrastive loss, we use balance $b = 0.1$.

In the meta learning stage, we use SGD with momentum with an initial learning rate of 0.002 and $\gamma = 20$ for Conv4-64 and an initial learning rate of 0.0002 and $\gamma = 40$ for ResNet-12. We follow the standard implementation of multi-headed self attention as presented in [32]. In meta training stage, the temperature hyperparameter used for softening the logits is critical for convergence to a good solution. We set the temperature as 0.1 for both networks. The optimal value for k is set to 30 after a hyperparameter search.

The memory bank consists of features of 200 randomly sampled instances per base class computed using the trained encoder f_0. The value of k was fixed to be 20 after trying out values of k from 2 to 30 and choosing the best performing value on 1-shot classification on mini-ImageNet.

5.1 Results

We report the results of BaseTransformer and other methods for mini-ImageNet in Table 1 and tiered-ImageNet and CUB in Table 2 and 4 respectively. We can see that one shot performance of BaseTransformers is better than all competing methods. For fairness, we have excluded comparisons with works that use larger encoders or extra image data [39]. We make the following observations: 1) BaseTransformers are effective in improving 1 shot performance on all considered backbones and benchmarks; 2) In comparison to other works [11, 40] that use transformers for prototype adaptation, we show improvements of 4.1%, 1.66%, and 3.28% on mini-ImageNet, tiered-ImageNet, and CUB dataset in the 1-shot setting; 3) We do not see the strong improvements in 1-shot reflected in the 5-shot setting. We hypothesize that this could be because the prototypes in 5-shot setting are already a good estimate of the true prototype. We investigate this phenomenon in 5.3. Results with the oracle

Table 2: 5-way 1-shot and 5-way 5-shot classification accuracy (%) on tieredImageNet dataset for ResNet-12. The numbers in bold are the best performing methods for the corresponding setting.

Setups	1-shot	5-shot
ProtoNets[28]	65.65	83.40
SimpleShot[34]	69.75	85.31
FEAT[40]	70.80	84.79
CAN[11]	69.89	84.23
DeepEMD[42]	71.16	86.03
IEPT[43]	72.24	86.73
MELR[7]	72.14	**87.01**
InfoPatch[17]	71.51	85.44
DMF[37]	71.89	85.96
META-QDA[24]	69.97	85.51
PAL[18]	72.25	86.95
BaseTransformer	**72.46**	84.96

Table 3: Test accuracy over number of shots for BaseTransformer and SupportTransformer

shot	1	2	3	4	5
BT	70.8	74.61	78.1	80.23	82.37
ST	66.34	73.12	77.33	79.8	82.01

Table 4: 5-way 1-shot and 5 way 5-shot classification accuracy (%) on CUB dataset. The numbers in bold are the best performing methods for the corresponding setting.

Setups	1-shot		5-shot	
Backbone	Conv4-64	Res12	Conv4-64	Res12
ProtoNets[28]	64.42	-	81.82	-
FEAT[40]	68.87	-	82.90	-
DeepEMD[42]	-	75.65	-	88.69
IEPT[43]	69.97	-	84.33	-
MELR[7]	70.26	-	**85.01**	-
BaseTransformer	**72.15**	**82.27**	82.12	**90.64**

top-k querying function are reported in 5.4. See supplementary for comparison with other baselines that use semantic knowledge and detailed results with 95% confidence intervals.

5.2 Ablation studies

Table 6 provides detailed ablation study of the various parts of our method for the Conv4-64 encoder. We can see that performance without BaseTransformer and SimCLR-pretraining is similar to that of Prototypical Networks. Including just InfoNCE as the auxiliary loss in the pretraining stage improves performance by 1.3%. Applying BaseTransformers with visual querying on Prototypical Networks further improves one shot accuracy to 54.46%. Using SimCLR in the pretraining stage with BaseTransformers improves accuracy further to 57.38%. This shows that the SimCLR loss in the pretraining stage is necessary to prevent supervision collapse and provide the BaseTransformer with robust base features. Finally, applying semantic querying gives a further improvement of $\sim 2\%$.

5.3 5-shot results

We believe that the performance improvements from using base dataset is only significant in the 1-shot to 3-shot domain. We ran experiments comparing BaseTransformer (BT) with semantic querying to SupportTransformer (ST), a variant of BT where the $Q = \sum_{y_i \in c} \phi(x_i)$ and $K = V = \{\phi(x_i)\}$ where $y_i \in c$, keeping all other hyperparameters same. Here Q is the prototype of class c and $K = V$ are the support instances of class c. Test accuracy of ST approaches that of BT as the number of shots approaches 5, showing that the prototypes from 5 different support instances of the novel class become as good as the prototype computed using base instances queried via a semantic query (Table 3).

Table 5: 1 shot results using oracle querying function

Querying Setup	mini-ImageNet	tiered-ImageNet
Visual	67.40±0.20	71.05±0.18
Semantic	70.88±0.17	72.46±0.19
Oracle	72.38±0.18	76.55±0.17

Table 6: Ablation study of BaseTransformer

SimCLR-pre	Querying	BT	1-shot
No	NA	No	51.65
Yes	NA	No	52.68
No	Visual	Yes	54.46
Yes	Visual	Yes	57.38
Yes	Semantic	Yes	59.37

5.4 Oracle querying

Table 5 reports 1-shot classification results using visual, semantic, and oracle querying for the mini-ImageNet and tiered-ImageNet datasets. Oracle querying uses the ResNet-12 encoder trained on both seen and unseen classes in the dataset. Then the closest base classes are found by the Euclidian similarity between the class prototypes estimated using all the instances in the class. We see that by improving the querying function, BaseTransformers can improve 1-shot accuracy by a significant margin, especially for the tiered-ImageNet dataset where the classes are distributed into seen and unseen classes with limited semantic overlap [22]. This shows that the 1-shot performance of the BaseTransformer architecture is limited by the querying function. We leave the search for an optimal querying function for the future.

5.5 Visualization of learnt attention over base datapoints

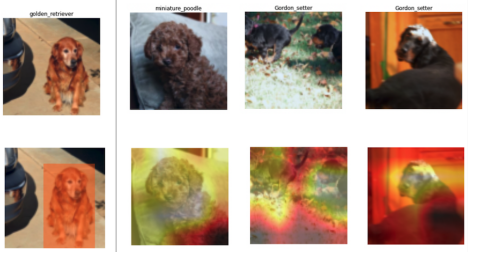

Figure 3: Left: support instance; right: the three closest base instances (top) and attention maps overlaid over the closest base instances (bottom).

We visualize the attention maps learnt by the BaseTransformer in Fig. 3. These are obtained by overlaying the resized attention map over the corresponding image of base instance selected by the querying function. We can see that for each support image, BaseTransformer has learnt to attend to semantically similar regions of base instances. BaseTransformer is successful in identifying multiple visually similar features in base instance images when there are multiple instances of the class in one image. For example for golden retriever, the BaseTransformer attends to two instances of gordon setter without being explicitly trained to identify multiple gordon setters – see supplementary material for more examples.

6 Conclusion

In this paper we propose that the one shot performance of metric learning based few shot approaches is hindered by the bias in estimation of the prototype. We show that prototype estimation can be improved by using well supported base instance features. Our proposed method, BaseTransformers, adapts the prototype by making use of learnt correspondences between the support instance and closest base class instances. Extensive experiments on three benchmarks and two encoders show the effectiveness of our method.

Acknowledgement

This publication has emanated from research conducted with the financial support of Science Foundation Ireland under Grant number 18/CRT/6183. For the purpose of Open Access, the author has applied a CC BY public copyright licence to any Author Accepted Manuscript version arising from this submission.

References

[1] Arman Afrasiyabi, Jean-François Lalonde, and Christian Gagné. Associative alignment for few-shot image classification. In *European Conference on Computer Vision*, pages 18–35. Springer, 2020.

[2] Sungyong Baik, Myungsub Choi, Janghoon Choi, Heewon Kim, and Kyoung Mu Lee. Meta-learning with adaptive hyperparameters. *Advances in Neural Information Processing Systems*, 33:20755–20765, 2020.

[3] 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.

[4] Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. A closer look at few-shot classification. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=HkxLXnAcFQ.

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423.

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

[7] 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.

[8] 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.

[9] 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.

[10] Fusheng Hao, Fengxiang He, Jun Cheng, Lei Wang, Jianzhong Cao, and Dacheng Tao. Collect and select: Semantic alignment metric learning for few-shot learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8460–8469, 2019.

[11] 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.

[12] 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*, pages 8822–8833, 2021.

[13] Claudia Leacock and Martin Chodorow. Combining local context and wordnet similarity for word sense identification. *WordNet: An electronic lexical database*, 49(2): 265–283, 1998.

[14] 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.

[15] Wenbin Li, Lei Wang, Jinglin Xu, Jing Huo, Yang Gao, and Jiebo Luo. Revisiting local descriptor based image-to-class measure for few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7260–7268, 2019.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[21] Tsendsuren Munkhdalai and Hong Yu. Meta networks. In *International Conference on Machine Learning*, pages 2554–2563. PMLR, 2017.

[22] 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.

[23] Eunbyung Park and Junier B Oliva. Meta-curvature. *Advances in Neural Information Processing Systems*, 32, 2019.

[24] Aravind Rajeswaran, Chelsea Finn, Sham M Kakade, and Sergey Levine. Meta-learning with implicit gradients. *Advances in neural information processing systems*, 32, 2019.

[25] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=rJY0-Kcll.

[26] 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*, 2019. URL https://openreview.net/forum?id=BJgklhAcK7.

[27] Xiahan Shi, Leonard Salewski, Martin Schiegg, Zeynep Akata, and Max Welling. Relational generalized few-shot learning. *British Machine Vision Conference*, 2020.

[28] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30, 2017.

[29] 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.

[30] 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.

[31] Eleni Triantafillou, Richard Zemel, and Raquel Urtasun. Few-shot learning through an information retrieval lens. *Advances in neural information processing systems*, 30, 2017.

[32] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[33] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. *Advances in neural information processing systems*, 29, 2016.

[34] Yan Wang, Wei-Lun Chao, Kilian Q Weinberger, and Laurens van der Maaten. Simpleshot: Revisiting nearest-neighbor classification for few-shot learning. *arXiv preprint arXiv:1911.04623*, 2019.

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

[36] Ziyang Wu, Yuwei Li, Lihua Guo, and Kui Jia. Parn: Position-aware relation networks for few-shot learning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6659–6667, 2019.

[37] 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.

[38] Jin Xu, Jean-Francois Ton, Hyunjik Kim, Adam Kosiorek, and Yee Whye Teh. Metafun: Meta-learning with iterative functional updates. In *International Conference on Machine Learning*, pages 10617–10627. PMLR, 2020.

[39] Shuo Yang, Lu Liu, and Min Xu. Free lunch for few-shot learning: Distribution calibration. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=JWOiYxMG92s.

[40] Han-Jia Ye, Hexiang Hu, De-Chuan Zhan, and Fei Sha. Few-shot learning via embedding adaptation with set-to-set functions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8808–8817, 2020.

[41] Sung Whan Yoon, Jun Seo, and Jaekyun Moon. Tapnet: Neural network augmented with task-adaptive projection for few-shot learning. In *International Conference on Machine Learning*, pages 7115–7123. PMLR, 2019.

[42] Chi Zhang, Yujun Cai, Guosheng Lin, and Chunhua Shen. Deepemd: Few-shot image classification with differentiable earth mover's distance and structured classifiers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12203–12213, 2020.

[43] 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.

[44] 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.

[45] 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.