Ortho-Shot: Low Displacement Rank Regularization with Data Augmentation for Few-Shot Learning

Uche Osahor
West Virginia University
uo0002@mix.wvu.edu

Nasser M. Nasrabadi
West Virginia University
nasser.nasrabadi@mail.wvu.edu

Abstract

In few-shot classification, the primary goal is to learn representations from a few samples that generalize well for novel classes. In this paper, we propose an efficient low displacement rank (LDR) regularization strategy termed Ortho-Shot; a technique that imposes orthogonal regularization on the convolutional layers of a few-shot classifier, which is based on the doubly-block toeplitz (DBT) matrix structure. The regularized convolutional layers of the few-shot classifier enhances model generalization and intra-class feature embeddings that are crucial for few-shot learning. Overfitting is a typical issue for few-shot models, the lack of data diversity inhibits proper model inference which weakens the classification accuracy of few-shot learners to novel classes. In this regard, we broke down the pipeline of the few-shot classifier and established that the support, query and task data augmentation collectively alleviates overfitting in networks. With compelling results, we demonstrated that combining a DBT-based low-rank orthogonal regularizer with data augmentation strategies, significantly boosts the performance of a few-shot classifier. We perform our experiments on the miniImagenet, CIFAR-FS and Stanford datasets with performance values of about 5% when compared to state-of-the-art.

1. Introduction

The performance of convolutional neural network (CNN) models largely depend on training a network with a lot of labelled instances and a spectrum of visual variations which are mostly in thousands per class [29]. The cost of labelling these data manually by human annotation as well as the scarcity of data that captures the complete diversity in a specific class significantly limits the potential of current vision models. However, the human visual system (HVS) can identify new classes with fewer labelled examples [27, 40], this unique trait of the HVS reveals the need to dive into new paradigms that would learn to generalize new classes with a limited amount of labelled data for each novel class. Recently, significant progress has been made towards better solutions using ideas of meta-learning [42, 48, 69, 32, 33, 38]. Empirically, it has been observed that the convolutional filters learned in deeper layers are highly correlated and redundant [64], thereby resulting in unstable training performance and vanishing gradients. These shortcomings of convolutional neural networks are also more damaging in few-shot classification due to the small data size. The potential pitfalls of such convolutional layers could result in under-utilization of model capacity, overfitting, vanishing and exploding gradients [16, 7], growth in saddle points [13] and shifts in feature statistics [24], which collectively affect model generalization.

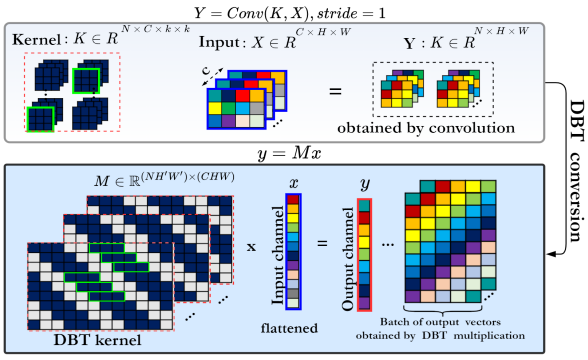

Figure 1: The convolution expression; Conv(K, X) is converted into a faster DBT vector representation; $y = Mx$.

The doubly block-toeplitz (DBT) matrix [18] is part of a class of low displacement rank (LDR) matrix constructions [72] that guarantee model reduction and computational complexity reduction in neural networks which is achieved by regularizing the weight matrices of network layers. The storage requirement of such a DBT-regularized network is reduced from $O(n^2)$ to $O(n)$ and the computational complexity can be reduced from $O(n^2)$ to $O(nrlogn)$, due to the fast matrix-vector multiplication property of LDR structured matrices as shown in Figure 2. It is also well established [34, 58] that when filters are learned to be as orthogonal as possible, model capacity is better utilized which in-turn improves feature expressiveness and intra-class feature representation [1, 62, 59, **?**].

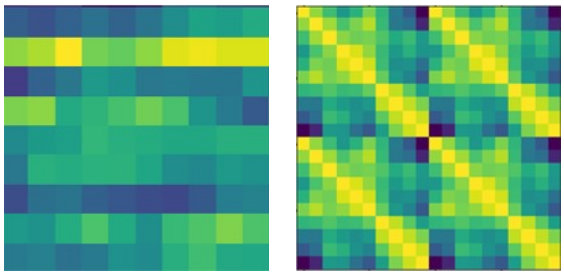

Figure 2: Toeplitz covariance matrices from features samples. This requires learning $O(n)$ parameters, in contrast to $O(n)^2$ for generic covariance matrices.

Our goal is to present an effective baseline model that harnesses good learned representations for few-shot classification kinds of tasks which perform better or at par with current few-shot algorithms [66, 63, 61, 53, 55, 42, 48, 38]. In a nutshell, we tackled the few-shot learning limitations by imposing orthogonal regularization on the model baseline which is a simpler yet effective approach compared to techniques used previously in [65, 15, 44]. We also incorporated data augmentation strategies that significantly improved data diversity and overall model performance.

1.1. Contributions:

- We adopted an efficient orthogonal regularization technique on convolutional layers of the few-shot classifier that enhances model generalization and intra-class feature embedding, using the doubly block toeplitz (DBT) matrix structure.
- We broke down the pipeline of a few-shot learner, and based on our findings, we established three augmentations strategies namely: support augmentation, query augmentation and task augmentation that aid in minimizing overfitting.
- We show with compelling results that combining a DBT-based regularizer with a robust augmentation strategy improves few-shot learning performance at an average of 5%.

2. Related works

Orthogonal regularization. In convolutional networks, orthogonal weights have being used to stabilize layer-wise distributions and to make optimization as efficient as possible. In [4, 37] the authors introduced orthogonal weight initialization driven by the norm preserving property of an orthogonal matrix. However, it was shown that the orthogonality and isometry property does not necessarily sustain throughout training [4] if the convolutional layers are not properly regularized. In other works, [25, 43, 23] considered Stiefel manifold-based hard constraints of weights [**?**], but their performance reported on VGG networks [52] were not as promising. These aforementioned methods [25, 43, 23] are associated with hard orthogonality constraints and in most cases, they have to repeat singular value decomposition (SVD) during training which is computationally expensive on the GPUs. A recent work adopted soft orthogonality [2, 3, 4, 67], where the Gram matrix of the weight matrix K is required to be close to identity, given as $\lambda \left\| K^T K - I \right\|_F^2$, where λ is the Frobenius norm-based regularization coefficient. It's a more efficient approach than the hard orthogonality assumption [25, 43, 23, 20, 68] and can be viewed as a different weight decay term limiting the set of parameters close to a Stiefel manifold [**?**]. Their approach constrained orthogonality among filters in one layer, leading to smaller correlations among learned features and implicitly reducing the filter redundancy. However, there are special cases where the Gram matrix cannot be close to identity which implies that matrix K is overcomplete [58]. Similarly, other works explored orthogonal weight initialization [55], mutual coherence with the isometry property [4], penalizing off-diagonal elements [9] towards improving kernel orthogonality.

In general, the orthogonality of K alone is not sufficient to make the linear convolutional layer orthogonal among its filters. Due to these shortcomings, we apply the improved regularization technique used in [64, 31]. We adopt the DBT matrix denoted as M with a filter K, while we keep the reshaped input x and output y intact. The matrix multiplication; $y = Mx$ enforces the orthogonality of M as shown in Figure 1 and Figure 3.

Augmentation. Data augmentation has become a well established technique for most image classifiers and deep networks, as it provides an efficient strategy that significantly mitigates the models' vulnerability to overfitting. In contrast, data augmentation still has room for expansion and adaptation in few-shot classification or other derivatives of meta-learning in general. Existing works [57, 26, 56], apply basic data augmentation strategies like random crops, horizontal flips and color jitter as the staple method for most meta-learning applications. However, these aforementioned techniques have plateaued in performance with little room for significant improvement [47, 41]. Other works have added random noise to labels to alleviate overfitting [46], some techniques rotate all the images in a class and consider the newly rotated class as distinct from its parent class. Recent works [12, 51, 41, 15, 45] are recording better performance values when augmentation strategies are injected within the meta-learning pipeline.

In our work, we explored the benefits of including augmentation strategies along the pipeline of a DBT regularized few-shot classifier. We identified how different augmentation approaches could affect a few-shot classifier when placed strategically along the classifier pipeline. At the core of our findings, we observed that the classifier is more sensitive to query data than support data.

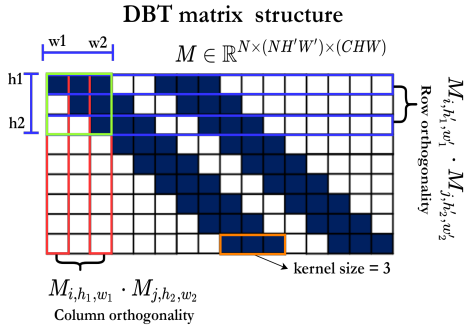

Figure 3: A doubly block-Toeplitz (DBT) matrix $M \in \mathbb{R}^{(NH'W')\times(CHW)}$ derived from the kernel tensor $K \in \mathbb{R}^{N\times C\times k\times k}$.

Toeplitz matrix applications. Kimitei et al. [28] used toeplitz matrices with Tikhonov regularization [39] as a mathematical approach to restoring blurred images. They explored their techniques on image restoration, enhancement, compression and recognition. In [19], the authors presented modern computational methods for treating linear deconvolution problems, they showed how to exploit the toeplitz structure to derive efficient numerical deconvolution algorithms. In compressive sensing applications [54], toeplitz-like matrices allow the entire signal to be efficiently acquired and reconstructed from relatively few measurements, compared to previous compressive sensing frameworks where a random measurement matrix is employed.

3. Background

We consider a meta learning scenario for an N-shot, K-way classification problem where the training and testing task datasets can be represented as $\mathcal{T} = \{\mathcal{D}_i^{train}, \mathcal{D}_i^{test}\}_{i=1}^{I}$. Such a meta-training task is divided into $\mathcal{D}_i^{train} = \{(x_t, y_t)\}_{t=1}^{T}$ and $\mathcal{D}_i^{test} = \{(x_q, y_q)\}_{q=1}^{Q}$, called a meta-training set [42, 48, 69, 32, 33, 38]. The set of D^{train} and D^{test} represent a small number of samples from the same distribution. We implement a DBT-based learner $\mathcal{B}_{dbt}(\cdot)$ to train the model for a given input feature denoted as $y = f_\theta(\mathbf{x}_*)$, where (*) denotes implementations for train and test sets. We then map train and test examples into a DBT structured embedding space $\Psi_* = f_\theta(\mathbf{x}_*)$.

The objective of our model becomes:

$$\begin{aligned}\theta &= \mathcal{B}_{dbt}(\mathcal{D}_i^{train}; \phi) \\ &= \arg\min_{\theta} \mathcal{L}^{base}(\mathcal{D}_i^{train}; \theta, \phi) + \mathcal{R}(\theta),\end{aligned} \tag{1}$$

where ϕ represents the parameters of the embedding model, $\mathcal{L}^{base}$ is the loss function and $\mathcal{R}$ is the regularization as described in Section 4.2. At the end of meta-training, the performance of the model is evaluated on a set of tasks $\mathcal{S} = \{(D_i^{train}, D^{test})\}_{i=1}^{I}$ called the meta-testing set. The final evaluation representation over the test set is:

$$E_S[L^{meta}(D^{test}; \theta, \phi)]. \tag{2}$$

The goal of meta learning is to learn a transferable efficient embedding model f_θ that generalizes to new tasks. As described in section 4, we deviated from popular techniques [63, 53, 55, 14] that train classifiers with convolutional blocks with some form of hard orthogonality constraint [41]. Our strategy, imposes a better low displacement rank DBT-based soft orthogonality constraint on the classifier network to produce more efficient embeddings for the base learner. The final embedding model is given as:

$$\phi = \arg\min_{\phi} \mathcal{L}^{ce}(\mathcal{D}^{new}; \phi), \tag{3}$$

where $\mathcal{D}_i^{new}$ is the task from $\mathcal{T}$ and $\mathcal{L}^{ce}$ denotes the cross-entropy loss between predictions and ground truth labels.

3.1. Doubly-block toeplitz (DBT) regularization

The feature interaction between two weights vectors v and w, within the layers of a few-shot classifier involves a convolution operation which can simply be represented as $v * w = \sum_{i=0}^{k} v(i)w(k-i)$, such that if v has length m and w has length n then $v * w$ has length $m + (n-1)$. Unfortunately, this computation involves $O(nm)$ operations which is not suitable for fast linear algebraic computations and intra-class parameter sharing which is critical for few shot learning. For such computations, if we consider a single convolution layer with input tensor $X \in \mathbb{R}^{C\times H\times W}$ and kernel $K \in \mathbb{R}^{N\times C\times k\times k}$, the convolution's output tensor is expressed as $Y = Conv(K, X)$, where $Y \in \mathbb{R}^{N\times H'\times W'}$, we replaced the convolution operator $(*)$ with $Conv(.)$ for simplicity. N, H, W and C are the number of kernels, height, width and channel of the input tensor, respectively. While k represents the kernel size and H', W' are the height and width for the output tensor, respectively.

Inline with our goal to improve the computational complexity and enhance better feature representation, we adapted a DBT matrix construction by utilizing the linear property of the convolution operation. The convolution expression; Conv(K, X) is converted into a faster DBT matrix-vector representation given as:

$$Y = Conv(K, X) \Leftrightarrow y = Mx. \tag{4}$$

This simple rearrangement establishes the foundation for adapting the DBT regularizer in our few-shot classifier network. Where M is the DBT matrix, x and y represent flattened input and output tensors, respectively. M is structured and is of rank $r << min(m, n)$ [58], this representation minimizes the storage requirements to $(mr + nr)$ parameters and accelerates the matrix-vector multiplication time to $O(mr + nr)$. Section 1: Figure 1 in the supplementary material shows the hierarchy for storage cost and operation count for matrix-vector multiplications. This DBT formulation stabilizes the spectrum of the newly derived DBT-based matrix M. In section 1.1 and 1.4 of the supplementary material, we reflect the overall benefit of the DBT model.

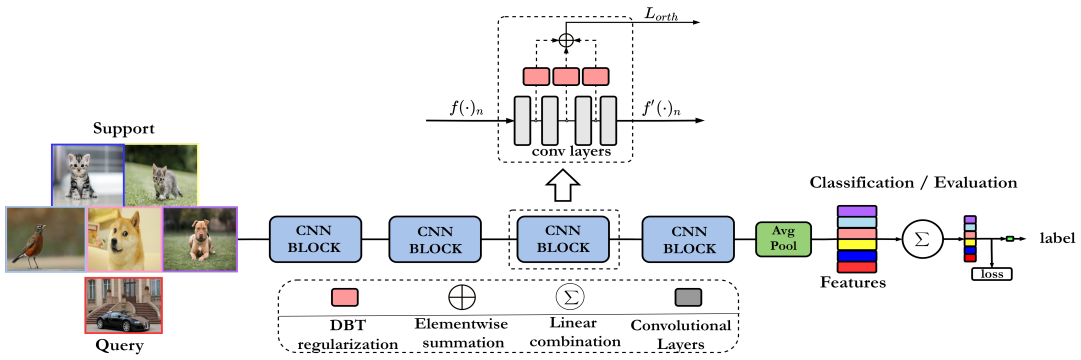

Figure 4: The network depicts a DBT-regularized few shot learner. The network embeddings $B_{dbt}(\cdot)$ are regularized based on the DBT structured matrix. The dotted box of the CNN block illustrates the inner translation between convolution layer embeddings $f(\cdot)$ and the more efficient DBT-based embeddings denoted as $f'(\cdot)$, see above. The Algorithm 1 gives a logical representation of the training process.

Algorithm 1 Ortho-shot algorithm

Require: $\mathcal{D} \leftarrow \{\mathcal{X}, \mathcal{Y}\}_{i=1}^{N}$
procedure ORTHOGONAL-REGULARIZER
  $I_{ro} \leftarrow t(I)$ ▷ toeplitz matrix
  $M \leftarrow Conv(K, K)$
  $y \leftarrow \lambda(\|M - I_{ro}\|)$
  $\psi(\cdot) \leftarrow Mx$ ▷ DBT based output
Require: $\mathcal{D}^{train}, \mathcal{D}^{test} \leftarrow \{\mathcal{X}^t, \mathcal{Y}^t\,;\, \mathcal{X}^q, \mathcal{Y}^q\}_{i=1}^{K}$
procedure FEW SHOT LEARNING
  if Train **then**
    for $i = 1$ **do**, T
      $B_{dbt} \leftarrow S_{\psi}$ ▷ DBT model
      $\mathcal{L}^{dce} + \mathcal{L}^{orth} \leftarrow loss(B_{dbt}(\mathcal{X}^t), \mathcal{Y}^t)$
      $\mathcal{L}_{dbt}^{total} \leftarrow \mathcal{L}^{dce} + \lambda\mathcal{L}^{orth}$
      ▷ (orthogonal regularization)
  if Test **then**
    for $i = 1$ **do**, Q
      $\mathcal{L}^{test} \leftarrow loss(B_{dbt}(\mathcal{X}^q), Y^q)$

4. The proposed method

We present an efficient low displacement rank (LDR) regularization strategy termed Ortho-Shot that imposes orthogonal regularization on the convolutional layers of a few-shot classifier which is based on the doubly-block toeplitz (DBT) matrix structure [64, 23]. Our technique, as reflected in section 4.1 deviates from popular methods that train classifiers with convolutional blocks with some form of hard orthogonality constraint. We also adapted a set of augmentation strategies based on the support, query and task datasets to boost overall model performance. In general, our approach enhances model generalization, intra-class feature embeddings and also minimizes overfitting for a few-shot classifier. To further describe our approach, we consider a single convolutional layer case. We extract feature embeddings $X \in \mathbb{R}^{C\times H\times W}$ from the intermediate convolutional layers of the few-shot classifier and then flatten it to a vector $x \in \mathbb{R}^{1\times(H\times W)}$. The weight tensor; K of our model is also converted to a doubly block-Toeplitz (DBT) matrix $M \in \mathbb{R}^{(NH'W')\times(CHW)}$ derived from kernel tensor $K \in \mathbb{R}^{N\times C\times k\times k}$ as shown in Figure 3. With the aforementioned matrix structure, we are able to apply a better orthogonality constraint as described by the Lemma in 4.2. In Figure 4, we show a fully regularized setup for a single CNN block. The network embeddings $B_{dbt}(\cdot)$ are regularized based on the DBT structure and the entire losses from each respective layer is summed up to L_{orth}. We show promising results for our technique as described by the CAM plots in Figure 5.

4.1. Convolutional orthogonality

A DBT kernel matrix M can be applied on both a rectangular or square case, where kernel $M \in \mathbb{R}^{(NH'W')\times(CHW)}$ dimensions can be rectangular $(NH'W') \leq (CHW)$ or square, $(NH'W') > (CHW)$. In the rectangular case, the uniform spectrum applies row orthogonal convolution while the square case requires column orthogonal convolution. In theory, the DBT kernel M is highly structured and sparse [31] as a result, an equivalent representation is required to regularize the spectrum of M to be uniform [64, 23]. We give the cases for both row and column orthogonality and we also propose an equivalent representation in this section.

Row orthogonality case. The row of matrix M corresponds to a filter at a particular spatial location flattened to a vector, denoted as $M_{i,h'w'} \in \mathbb{R}^{CHW}$. The row orthogonality condition is given as:

$$\langle M_{i,h'_1,w'_1} \cdot M_{jh'_2,w'_2} \rangle = \begin{cases} 1, & i_{h'_1,w'_1} = j_{h'_2,w'_2} \\ 0, & otherwise. \end{cases} \quad (5)$$

This results to an equivalent of Equation 4 as the following self-convolution:

$$Y = Conv(K, K, padding = P, stride = S) = I_{r0}, \quad (6)$$

where $I_{r0} \in \mathbb{R}^{N\times N\times(2P/S+1)\times(2P/S+1)}$ is a tensor with an identity matrix at the centre and zeros entries elsewhere.

Column orthogonality case. If $X_{i,hw} \in \mathbb{R}^{C\times H\times W}$ denotes an input tensor, which has all zero except an entry at the i^{th} input channel at spatial location (h, w). Then we can denote the flattened vector as $x_{ihw} \in \mathbb{R}^{C\times H\times W}$ derived from $X_{i,hw}$. A column vector $M_{i,hw}$ of M is obtained by multiplying M and column vector $x_{i,hw}$. Similar to the row orthogonality,

$$Y = Conv(K^T, K^T, padding = k-1, stride = 1) = I_{c0}, \tag{7}$$

where K^T is the input-output transposed K, i.e., $K^T \in \mathbb{R}^{C\times N\times k\times k}$, $I_{c0} \in \mathbb{R}^{C\times C\times(2k-1)\times(2k-1)}$ has all zeros except for the center $C \times C$ entries as an identity matrix. Figure 2 illustrates the DBT matrix M structure of our model.

4.2. Row-column orthogonality equivalence

To develop an equivalent representation for row and column orthogonality, we build on the equation described by **lemma 1**, which states that the minimizing of the column orthogonality and row orthogonality costs are equivalent [31] due to the property of the Frobenius norm.

Lemma 1: The row orthogonality cost $\lambda \left\| KK^T - I_{r0} \right\|_F^2$ is equivalent to the column orthogonality cost $\lambda \left\| K^T K - I_{c0} \right\|_F^2 + U$ where U is a constant. This implies that convolution orthogonality independent of the shape of M (square or rectangular) can be regularized, given as:

$$L_{orth} = \lambda \left\| K^T K - I_{r0} \right\|_F^2, \tag{8}$$

where L_{orth} is the DBT-based orthogonal regularization term that depends only on Equation 6 and replaces the $\mathcal{R}(\cdot)$ term in Equation 1.

5. Experimental setup and analysis

Our experiments were conducted on the miniImagNet, CIFAR-FS, Stanford Dogs and Stanford Cars datasets, respectively. We used the R2-D2 base leaner [8], the "ResNet-12" and "64-64-64-64" backbone for different few-shot learning modes used in our work. Data augmentation strategies were also analysed to determine the best combination for a DBT-regularized model. The complete details for of the entire setup is expressed in section 1.2 of the supplementary material.

5.1. Data augmentation strategy

Motivated by the impact of applying a diverse augmentation strategy on meta-learners, we established three unique augmentation approaches; support, query and task augmentation that contribute to the overall classifier performance, aimed at minimising overfitting. Our empirical analysis confirm that support augmentation increases the number of fine tuning data while the query data improves evaluation performance while training the classifier. Similarly, task augmentation is used to increase the number of classes per task while training. We adapted a couple of augmentation techniques such as CutMix [70], where image patches are cut and pasted among training images and the ground truth labels are also mixed proportionally within the area of the patches. Mixup [50], a technique that generates convex combinations of pairs of examples and their labels, which proved to be effective for support and query augmentation strategies. As well as Self-Mix [71] in which an image is substituted into other regions in the same image. This dropout effect improves few-shot learning generalization overall. In addition, we implemented standard data augmentation techniques by randomly erasing patches from the images (Random Erase), horizontally flipping the images (Horizontal Flip), rotating the images at different specified angles (Rotation) and Color Jitter, where we randomly change the brightness, contrast and saturation of the images. To boost the performance of our augmentation strategy, we combine different augmentation techniques using the MaxUp augmentation approach proposed in [17]. The rationale behind MaxUp augmentation is to minimize training loss by performing parameter updates on the task that maximizes loss in a min-max optimization manner, the MaxUp expression is given as:

$$\min_{\theta} E_{\mathcal{T}}[\max_{M\in\mathcal{S}} \mathcal{L}(\mathcal{B}_{\theta'}, M(\mathcal{T}^q))], \tag{9}$$

where θ' represents the model parameters, $\mathcal{B}$ is the base model, $\mathcal{L}$ is the loss function and $\mathcal{T}$ is a task for both support and query data; $\mathcal{T}^s$ and $\mathcal{T}^q$, respectively.

5.2. Augmentation performance

In this section, we investigate the performance of a few-shot classifier for different augmentation strategies. We investigated three test cases that check the training performance when data is sampled from the support, query and task data, respectively. Our approach is similar to techniques adapted by [41, 30, 50] that examine the impact of augmentation on a diverse set of data combinations.

Case 1: We trained the model at an equal number of support and query data as indicated in Table 1, so as to establish a baseline performance of the model. We use this strategy to compare the impact of any of the data pools (support or query) when any of the augmented pairs is reduced.

Case 2: We initiated training of the classifier by randomly sampling from 5 and 10 unique samples per class of the support data while using the entire query data pool. Using this approach, we reduced the influence of support data in order to examine the impact of the diverse pool of query data on the classifier. Our findings reflected in Table 1 show accuracy values at $\pm$ 2%. This is a clear indication that

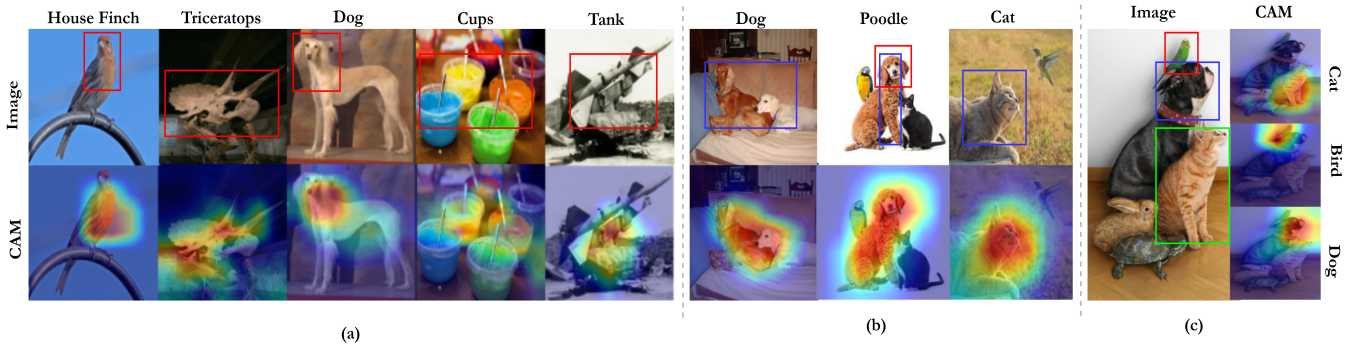

Figure 5: Illustration of CAM plots. (a) The second row represents CAM plots for single classes. The red squares highlight regions of interest clearly highlighted by the model.(b) Shows more complex scenarios where multi classes are involved. Classes are clearly separated from non-classes of interest. In (c), all the objects separated by bounding boxes are clearly localised as indicated by the CAM plot.

Table 1: Few-shot classification accuracy (%) using R2-D2 base leaner with a ResNet-12 backbone on CIFAR-FS dataset. Support, Query and Task columns represent the number of samples per class for support, query data and the total number of tasks available.

Support	Query	Task	1-shot	5-shot
500	500	full	71.41± 0.21	86.01± 0.08
100	500	full	70.11± 0.01	83.00± 0.03
10	500	full	70.72± 0.01	81.41± 0.32
500	300	full	69.41± 0.11	72.41± 0.08
500	100	full	59.00± 0.21	70.41± 0.08
5 (random)	500	full	61.01± 0.11	80.41± 0.08
10 (random)	500	full	63.01± 0.30	81.24± 0.02

augmentation of query data plays a more significant role in the overall model performance. In contrast, we reduced the number of query data while maintaining the initially set cap for support data and recorded a decline in accuracy.

Case 3: To evaluate the impact of task augmentation, we used the CIFAR-FS data to initially allocate 10 distinct 5-way classification tasks (252 combinations) before training, while the support and query datasets are maintained equally at 500, respectively. We observed a decline in performance. However, as we increased the amount of task data, significant improvement is recorded, which confirms that task augmentation is crucial in few-shot learning.

In summary, we broke down the few-shot learning process to determine the influence of support, query and task augmentation, respectively. Our findings confirm that our baseline learner is most sensitive to query data [41]. In addition, task augmentation provided significant value (about 2%) that cannot be overlooked by the classifier.

5.3. Augmentation modes

This section builds on the findings of section 5.1, where we established three core data augmentation cases; support, query and task data augmentation. Similar to [41, 17, 12],

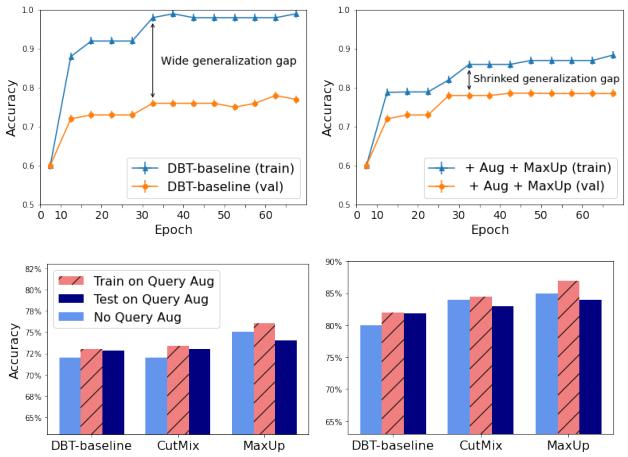

Figure 6: Accuracy results for training and validation on R2-D2 base-learner [8] with a DBT-regularized ResNet-12 backbone on the CIFAR-FS dataset. (Top Left) Baseline model and (Top right) Augmentation "Aug" and MaxUp. The MaxUp augmentation strategy narrows down the generalization gap and reduces overfitting. (Bottom left) 1-shot classification and (Bottom Right) 5-shot classification for Query data augmentation.

we used the CutMix, SelfMix, MixUp, Random Crop and Horizontal Flip augmentation methods on the support, query and task datasets, respectively. We identified the best augmentation combinations that suit a few-shot learner and with our findings, we picked the best strategy to determine which mode of augmentation suits a DBT-regularized few-shot learner. To start with, we used the R2-D2 base learner [8] and the CIFAR-FS database to evaluate the augmentation performance on support, query and task augmentations as shown in Table 1. Our findings show that the pair of CutMix and SelfMix augmentation produces the best results with over 2.5% in accuracy improvement [41]. Other approaches lag behind in performance at about ± 3% for

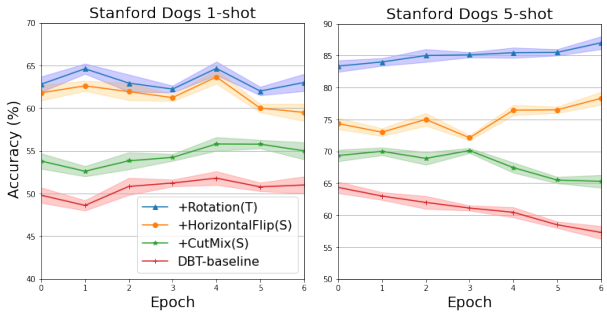

Figure 7: Accuracy plots for different datasets compared to the baseline model. Augmentations techniques were applied on Task "T" and Support "S" datasets. Overall, accuracy for 5-shot is maintained at 85-88% while for 1-shot, a range of 65-68% is recorded.

Table 2: Few-shot classification accuracy (%) using R2-D2 base leaner with a ResNet-12 backbone on the CIFAR-FS dataset. Support(S), Query(Q) and Task(T) data are used on different augmentation strategies.

Augmentation	**1-shot**	**5-shot**
CutMix(Q)	76.01± 0.21	87.14± 0.08
+ CutMix(S)	75.11± 0.31	85.30± 0.14
+ Horizontal Flip(S)	76.32± 0.11	87.01± 0.23
+ Rotation(T)	75.33± 0.25	**87.68± 0.03**
SelfMix(Q)	76.04± 0.21	86.81± 0.08
+ CutMix(S)	76.19± 0.29	86.35± 0.16
+ Horizontal Flip(S)	75.27± 0.32	86.88± 0.03
+ Rotation(T)	75.61± 0.22	**87.40± 0.18**
MixUp(Q)	72.14± 0.01	82.81± 0.08
+ CutMix(S)	71.03± 0.29	85.15± 0.11
+ Horizontal Flip(S)	72.27± 0.10	83.08± 0.01
+ Rotation(T)	74.10± 0.11	85.10± 0.22

both 1-shot and 5-shot cases. Secondly, since the CutMix and SelfMix methods stand out as the best augmentation approach for our setup, we used them as bases to combine augmentations on the three data cases; support, query and task, respectively as shown in Table 2. Model performance significantly improved with the best case occurring when CutMix (query) is combined with SelfMix (support).

5.4. DBT-regularization with data augmentation

As discussed in section 1, DBT-based regularization improves model generalization and intra-class feature expressiveness. Data augmentation on the other hand creates sufficient data diversity which helps to mitigate overfitting. In this section, we highlight the collective benefits of combing a DBT-based regularizer with augmentation strategies for few-shot learning, using different datasets.

Accuracy results with different datasets: In this section, we setup a testing scheme where we evaluate our method

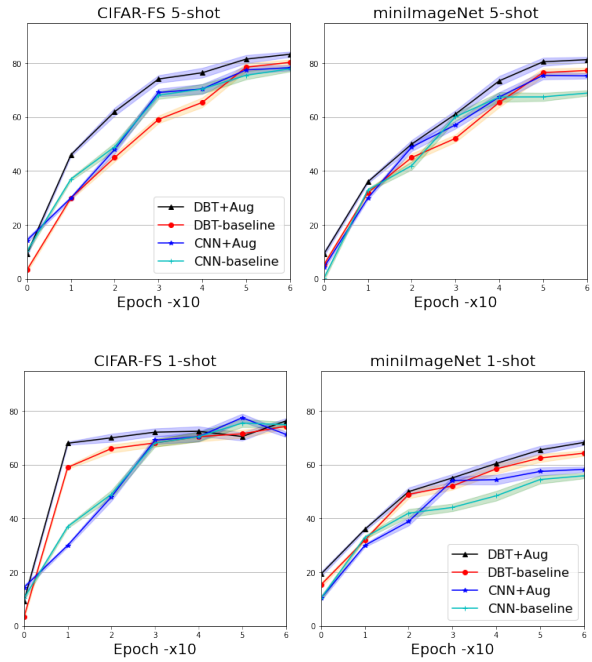

Figure 8: Model accuracy plots for CIFAR-FS and miniImageNet datasets on CNN and DBT model baselines with augmentation "Aug" and without Augmentation for 5-shot and 1-shot cases.

over four runs which is quite similar to techniques applied in [60]. We computed the mean accuracy as the accuracy for every run, the experiments are conducted on the Stanford Dogs, Stanford Cars, miniImageNet and CIFAR-FS datasets, respectively as shown in Table 4 and Table 3. Our accuracy results for 5-shot were maintained at 80-88% while for 1-shot at a range of 65-68% as shown in Figure 7 and Figure 8. Our baseline integrated with the DBT-based regularizer "DBT-baseline" model performs at about 2% better than state-of-the art without data augmentation. Applying the CutMix and SelfMix augmentation on the query "Q" and support "S" datasets, show significant improvement. Rotation "R" and Horizontal Flip "HF" are integrated into the CutMix and SelfMix data augmentation modes, respectively as indicated in Table 3.

Improvement with MaxUp augmentation: In this section, we evaluate the performance of our model with the Max-Up approach for both 1-shot and 5-shot classification. We use a similar experimentation setting described in [41] at different augmentation pool sizes. Figure 6 and Table 4 depict the impact of Max-Up with the augmentation strategies; denote generally as "Aug" for both train and validation data. We also show results for the baseline model without augmentation (DBT-baseline), with CutMix and MaxUp augmentation for different Query data schemes. We observe from Figure 6 (Top right) that the generalization gap shrinks considerably and by implication, overfitting is minimized

Table 3: Comparison to prior work on miniImageNet and CIFAR-FS. Few-shot classification accuracy (%) using R2-D2 base leaner a "ResNet-12" and "64-64-64-64" backbone on CIFAR-FS and miniImageNet datasets, respectively. We applied Rotation (R) to the CutMix and Horizontal Flip (HF) to the SelfMix augmentation modes. "Q" denotes query data, "S" represents support data and "M" dentotes MaxUp.

		CIFAR-FS 5-way		miniImageNet 5-way	
DBT-model	**Backbone**	**1-shot**	**5-shot**	**1-shot**	**5-shot**
Basline(No Aug)	ResNet-12	70.26 ± 0.61	83.12 ± 0.53	55.03 ± 0.40	74.06 ± 0.24
CutMix(Q)	ResNet-12	71.46 ± 0.24	84.32 ± 0.73	57.36 ± 0.24	74.46 ± 0.11
CutMix(Q) + M	ResNet-12	$\mathbf{72.00 \pm 0.01}$	$\mathbf{86.20 \pm 0.61}$	$\mathbf{58.13 \pm 0.25}$	$\mathbf{75.69 \pm 0.74}$
SelfMix(S) + R	ResNet-12	62.56 ± 0.54	79.82 ± 0.33	50.38 ± 0.63	71.44 ± 0.08
SelfMix(S) + M	ResNet-12	$\mathbf{63.51 \pm 0.78}$	$\mathbf{80.20 \pm 0.66}$	$\mathbf{57.31 \pm 0.89}$	$\mathbf{72.69 \pm 0.70}$
CutMix(S) + HF	64-64-64-64	60.56 ± 0.29	85.32 ± 0.73	62.26 ± 0.63	79.28 ± 0.63
CutMix(S) + M	64-64-64-64	$\mathbf{63.42 \pm 0.17}$	$\mathbf{86.33 \pm 0.66}$	$\mathbf{63.31 \pm 0.89}$	$\mathbf{80.69 \pm 0.54}$
SelfMix(Q) + HF	64-64-64-64	75.56 ± 0.84	84.32 ± 0.73	66.31 ± 0.89	82.69 ± 0.74
SelfMix(Q) + M	64-64-64-64	$\mathbf{76.42 \pm 0.38}$	$\mathbf{86.10 \pm 0.36}$	$\mathbf{67.39 \pm 0.34}$	$\mathbf{83.44 \pm 0.24}$

Table 4: Experimental results that compare prior work on the Stanford Dogs, Stanford Cars and CIFAR-FS dataset. Average few-shot classification accuracy with 95 % confidence intervals. The second column shows which kind of embedding is employed, we used a 4-layer convolutional network with their respective filters in each layer.

	Stanford Dogs 5-way		Stanford Cars 5-way		CIFAR-FS 5-way	
Model	**1-shot**	**5-shot**	**1-shot**	**5-shot**	**1-shot**	**5-shot**
Matching Networks [63]	35.80 ± 0.99	47.50 ± 1.03	34.80 ± 0.98	44.70 ± 1.03	61.16 ± 0.89	72.86 ± 0.70
MAML [14]	44.81 ± 0.34	58.68 ± 0.31	47.22 ± 0.39	61.21 ± 0.28	55.92 ± 0.95	72.09 ± 0.76
Relation Nets [55]	43.33 ± 0.42	55.23 ± 0.41	47.67 ± 0.47	60.59 ± 0.40	62.45 ± 0.98	76.11 ± 0.69
Prototypical Networks [53]	37.59 ± 1.00	48.19 ± 1.03	40.90 ± 1.01	52.93 ± 1.03	51.31 ± 0.91	70.77 ± 0.69
DN4 [35]	45.41 ± 0.76	63.51 ± 0.62	59.84 ± 0.80	88.65 ± 0.44	52.79 ± 0.86	81.45 ± 0.70
PABN [22]	45.65 ± 0.71	61.24 ± 0.62	54.44 ± 0.71	67.36 ± 0.61	63.56 ± 0.79	75.35 ± 0.58
MATANet [10]	55.63 ± 0.88	70.29 ± 0.62	73.15 ± 0.88	91.89 ± 0.45	67.33 ± 0.84	83.92 ± 0.63
GNN [49]	46.38 ± 0.78	62.27 ± 0.95	55.85 ± 0.97	71.25 ± 0.89	51.83 ± 0.48	63.69 ± 0.94
Rfs [60]	55.64 ± 0.28	62.02 ± 0.63	79.64 ± 0.44	69.74 ± 0.72	83.41 ± 0.55	83.50 ± 0.11
Rfs-distill [60]	56.01 ± 0.48	64.82 ± 0.60	$\mathbf{82.14 \pm 0.43}$	71.52 ± 0.69	$\mathbf{86.03 \pm 0.49}$	84.10 ± 0.28
DBT-baseline	56.06 ± 0.03	71.00 ± 0.25	73.49 ± 0.01	92.02 ± 0.33	74.41 ± 0.50	84.21 ± 0.65
+ CutMix(Q) + R	56.36 ± 0.64	71.39 ± 0.04	73.69 ± 0.51	93.00 ± 0.15	74.81 ± 0.37	86.01 ± 0.67
+ SelfMix(Q) + HF	56.86 ± 0.64	72.19 ± 0.78	74.21 ± 0.01	93.30 ± 0.35	75.01 ± 0.15	87.01 ± 0.74
+ MaxUp	$\mathbf{57.06 \pm 0.63}$	$\mathbf{73.15 \pm 0.22}$	75.34 ± 0.41	$\mathbf{94.38 \pm 0.25}$	76.41 ± 0.25	$\mathbf{87.68 \pm 0.24}$

when the MaxUp strategy is adapted. MaxUp also adds the extra boost with an average of about 2.3% in performance.

Comparison with different methods: We compared our results against different methods [49, 10, 22, 35] as shown in Table 4. We observed that [60] is closest to ours but we outperform their approach significantly for the 5-shot cases by over 6% on the average. We recorded a better performance than GNN [49] and MATANet [11] using both the 5-way 1-shot and 5-way 5-shot few-shot learning settings, we saw an improvement of about 3.3% 4.2% and 3.16% on Stanford Dogs, Stan-ford Cars, and CIFAR-FS, respectively for 5-way 1-shot task while for the 5-way5-shot task, our method achieved about 4.7%, 2.1%, and 4.9% overall. Clearly, the MaxUp boost is significant in almost all cases.

6. Conclusion

We proposed a structured doubly block-toeplitz (DBT) matrix based model that imposes orthogonal regularization on the filters of the convolutional layers termed Ortho-Shot. Our approach was aimed at maintaining the stability of activations, preserving gradient norms, and enhancing feature transferability of deep networks. We also broke down the pipeline of a few-shot learner and based on our findings, we established three augmentations strategies that aid in minimizing overfitting and increasing data diversity. Our findings and empirical results confirm that a DBT regularized model is beneficial to few-shot classification and meta-learning in general.

References

[1] A. Araujo, B. Négrevergne, Y. Chevaleyre, and J. Atif. On lipschitz regularization of convolutional layers using Toeplitz matrix theory. *arXiv: Learning*, 2020. 1

[2] R. Balestriero and R. Baraniuk. Mad Max: Affine spline insights into deep learning. *arXiv: Machine Learning*, 2018. 2

[3] R. Balestriero and R. Baraniuk. A spline theory of deep networks. In *ICML*, 2018. 2

[4] N. Bansal, X. Chen, and Z. Wang. Can we gain more from orthogonality regularizations in training deep cnns? *ArXiv*, abs/1810.09102, 2018. 2

[5] P. Bartlett and W. Maass. Vapnik-Chervonenkis dimension of neural nets. 2003. 2

[6] P. Bartlett, V. Maiorov, and R. Meir. Almost linear VC-dimension bounds for piecewise polynomial networks. *Neural Computation*, 10:2159–2173, 1998. 1, 2

[7] Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5 2:157–66, 1994. 1

[8] L. Bertinetto, J. F. Henriques, P. H. S. Torr, and A. Vedaldi. Meta-learning with differentiable closed-form solvers. *ArXiv*, abs/1805.08136, 2019. 5, 6

[9] A. Brock, J. Donahue, and K. Simonyan. Large scale gan training for high fidelity natural image synthesis. *ArXiv*, abs/1809.11096, 2019. 2

[10] H. Chen, H. Li, Y. Li, and C. Chen. Multi-scale adaptive task attention network for few-shot learning. *ArXiv*, abs/2011.14479, 2020. 8

[11] Y. Chen, X. Wang, Z. Liu, H. Xu, and T. Darrell. A new meta-baseline for few-shot learning. *ArXiv*, abs/2003.04390, 2020. 8

[12] A. Dabouei, S. Soleymani, F. Taherkhani, and N. Nasrabadi. Supermix: Supervising the mixing data augmentation. *ArXiv*, abs/2003.05034, 2020. 2, 6

[13] Y. Dauphin, R. Pascanu, Çaglar Gülçehre, K. Cho, S. Ganguli, and Y. Bengio. Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. In *NIPS*, 2014. 1

[14] C. Finn, P. Abbeel, and S. Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, 2017. 3, 8

[15] S. Gidaris and N. Komodakis. Dynamic few-shot visual learning without forgetting. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4367–4375, 2018. 2

[16] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feed forward neural networks. In *AISTATS*, 2010. 1

[17] C. Gong, T. Ren, M. Ye, and Q. Liu. Maxup: A simple way to improve generalization of neural network training. *ArXiv*, abs/2002.09024, 2020. 5, 6

[18] R. Gray. Toeplitz and circulant matrices: A review. *Found. Trends Commun. Inf. Theory*, 2, 2005. 1

[19] P. Hansen. Deconvolution and regularization with Toeplitz matrices. *Numerical Algorithms*, 29:323–378, 2004. 3

[20] M. Harandi and B. Fernando. Generalized backpropagation, étude de cas: Orthogonality. *ArXiv*, abs/1611.05927, 2016. 2

[21] N. Harvey, C. Liaw, and A. Mehrabian. Nearly-tight VC-dimension bounds for piecewise linear neural networks. *ArXiv*, abs/1703.02930, 2017. 1, 2

[22] H. Huang, J. Zhang, J. yu Zhang, J. Xu, and Q. Wu. Low-rank pairwise alignment bilinear network for few-shot fine-grained image classification. *ArXiv*, abs/1908.01313, 2019. 8

[23] L. Huang, X. Liu, B. Lang, A. W. Yu, and B. Li. Orthogonal weight normalization: Solution to optimization over multiple dependent Stiefel manifolds in deep neural networks. *ArXiv*, abs/1709.06079, 2018. 2, 4

[24] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *ArXiv*, abs/1502.03167, 2015. 1

[25] K. Jia, D. Tao, S. Gao, and X. Xu. Improving training of deep neural networks via singular value bounding. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3994–4002, 2017. 2

[26] G. Kang, X. Dong, L. Zheng, and Y. Yang. Patchshuffle regularization. *ArXiv*, abs/1707.07103, 2017. 2

[27] T. Kietzmann, C. J. Spoerer, L. K. A. Sörensen, R. M. Cichy, O. Hauk, and N. Kriegeskorte. Recurrence is required to capture the representational dynamics of the human visual system. *Proceedings of the National Academy of Sciences of the United States of America*, 116:21854 – 21863, 2019. 1

[28] S. K. Kimitei. Algorithms for Toeplitz matrices with applications to image deblurring. Master's thesis, Georgia State University Georgia State University, Department of Mathematics., USA, 2011. 3

[29] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60:84 – 90, 2012. 1

[30] S. M. Kye, H. Lee, H. Kim, and S. J. Hwang. Transductive few-shot learning with meta-learned confidence. *ArXiv*, abs/2002.12017, 2020. 5

[31] Q. V. Le, A. Karpenko, J. Ngiam, and A. Ng. ICA with reconstruction cost for efficient overcomplete feature learning. In *NIPS*, 2011. 2, 4, 5

[32] K. Lee, S. Maji, A. Ravichandran, and S. Soatto. Meta-learning with differentiable convex optimization. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10649–10657, 2019. 1, 3, 2

[33] H. Li, D. Eigen, S. F. Dodge, M. D. Zeiler, and X. Wang. Finding task-relevant features for few-shot learning by category traversal. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–10, 2019. 1, 3

[34] J. Li, Y. Wu, J. Zhao, and K. Lu. Low-rank discriminant embedding for multiview learning. *IEEE Transactions on Cybernetics*, 47:3516–3529, 2017. 1, 2

[35] W. Li, L. Wang, J. Xu, J. Huo, Y. Gao, and J. Luo. Revisiting local descriptor based image-to-class measure for few-shot learning. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7253–7260, 2019. 8

[36] J. Liu, F. Chao, and C.-M. Lin. Task augmentation by rotating for meta-learning. *ArXiv*, abs/2003.00804, 2020.

[37] D. Mishkin and J. Matas. All you need is a good init. *CoRR*, abs/1511.06422, 2016. 2

[38] S. Motiian, Q. Jones, S. M. Iranmanesh, and G. Doretto. Few-shot adversarial domain adaptation. In *NIPS*, 2017. 1, 2, 3

[39] F. Natterer. Error bounds for Tikhonov regularization in Hilbert scales. *Applicable Analysis*, 18:29–37, 1984. 3

[40] A. Nayebi, D. Bear, J. Kubilius, K. Kar, S. Ganguli, D. Sussillo, J. DiCarlo, and D. Yamins. Task-driven convolutional recurrent models of the visual system. In *NeurIPS*, 2018. 1

[41] R. Ni, M. Goldblum, A. Sharaf, K. Kong, and T. Goldstein. Data augmentation for meta-learning. In *ICML*, 2021. 2, 3, 5, 6, 7

[42] B. N. Oreshkin, P. R. López, and A. Lacoste. Tadam: Task dependent adaptive metric for improved few-shot learning. In *NeurIPS*, 2018. 1, 2, 3

[43] M. Ozay and T. Okatani. Optimization on submanifolds of convolution kernels in CNNs. *ArXiv*, abs/1610.07008, 2016. 2

[44] H. Qi, M. Brown, and D. Lowe. Low-shot learning with imprinted weights. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5822–5830, 2018. 2

[45] S. Qiao, C. Liu, W. Shen, and A. Yuille. Few-shot image recognition by predicting parameters from activations. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7229–7238, 2018. 2

[46] J. Rajendran, A. Irpan, and E. Jang. Meta-learning requires meta-augmentation. *ArXiv*, abs/2007.05549, 2020. 2

[47] M. Ren, E. Triantafillou, S. Ravi, J. Snell, K. Swersky, J. Tenenbaum, H. Larochelle, and R. Zemel. Meta-learning for semi-supervised few-shot classification. *ArXiv*, abs/1803.00676, 2018. 2

[48] A. A. Rusu, D. Rao, J. Sygnowski, O. Vinyals, R. Pascanu, S. Osindero, and R. Hadsell. Meta-learning with latent embedding optimization. *ArXiv*, abs/1807.05960, 2019. 1, 2, 3

[49] V. G. Satorras and J. Bruna. Few-shot learning with graph neural networks. *ArXiv*, abs/1711.04043, 2018. 8

[50] J. Seo, H. G. Jung, and S.-W. Lee. Self-augmentation: Generalizing deep networks to unseen classes for few-shot learning. *Neural networks : the official journal of the International Neural Network Society*, 138:140–149, 2021. 5

[51] C. Shorten and T. Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6:1–48, 2019. 2

[52] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2015. 2

[53] J. Snell, K. Swersky, and R. Zemel. Prototypical networks for few-shot learning. In *NIPS*, 2017. 2, 3, 8

[54] X. Su, H. Yin, Y. Chai, Y. Xiong, and X. Tan. An improved Toeplitz measurement matrix for compressive sensing. *International Journal of Distributed Sensor Networks*, 10, 2014. 3

[55] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. Torr, and T. M. Hospedales. Learning to compare: Relation network for few-shot learning. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1199–1208, 2018. 2, 3, 8

[56] R. Takahashi, T. Matsubara, and K. Uehara. Data augmentation using random image cropping and patching for deep cnns. *IEEE Transactions on Circuits and Systems for Video Technology*, 30:2917–2931, 2020. 2

[57] L. Taylor and G. Nitschke. Improving deep learning with generic data augmentation. *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1542–1547, 2018. 2

[58] A. Thomas, A. Gu, T. Dao, A. Rudra, and C. Ré. Learning compressed transforms with low displacement rank. *Advances in neural information processing systems*, 2018:9052–9060, 2018. 1, 2, 3

[59] A. T. Thomas, A. Gu, T. Dao, A. Rudra, and C. Ré. Learning compressed transforms with low displacement rank, 2019. 1, 2

[60] Y. Tian, Y. Wang, D. Krishnan, J. Tenenbaum, and P. Isola. Rethinking few-shot image classification: a good embedding is all you need? *ArXiv*, abs/2003.11539, 2020. 7, 8, 2

[61] E. Triantafillou, R. Zemel, and R. Urtasun. Few-shot learning through an information retrieval lens. In *NIPS*, 2017. 2

[62] V. N. Vapnik. The nature of statistical learning theory. In *Statistics for Engineering and Information Science*, 2000. 1, 2

[63] O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, and D. Wierstra. Matching networks for one-shot learning. In *NIPS*, 2016. 2, 3, 8

[64] J. Wang, Y. Chen, R. Chakraborty, and S. X. Yu. Orthogonal convolutional neural networks. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11502–11512, 2020. 1, 2, 4

[65] Y.-X. Wang, R. B. Girshick, M. Hebert, and B. Hariharan. Low-shot learning from imaginary data. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7278–7286, 2018. 2

[66] Y.-X. Wang and M. Hebert. Learning to learn: Model regression networks for easy small sample learning. In *ECCV*, 2016. 2

[67] D. Xie, J. Xiong, and S. Pu. All you need is beyond a good init: Exploring better solution for training extremely deep convolutional neural networks with orthonormality and modulation. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5075–5084, 2017. 2

[68] H. Xu, Z. Wang, H. Yang, D. Liu, and J. Liu. Learning simple thresholded features with sparse support recovery. *IEEE Transactions on Circuits and Systems for Video Technology*, 30:970–982, 2020. 2

[69] H.-J. Ye, H. Hu, D. Zhan, and F. Sha. Learning embedding adaptation for few-shot learning. *ArXiv*, abs/1812.03664, 2018. 1, 3

[70] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, and Y. Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6022–6031, 2019. 5

[71] H. Zhang, M. Cissé, Y. Dauphin, and D. Lopez-Paz. mixup: Beyond empirical risk minimization. *ArXiv*, abs/1710.09412, 2018. 5

[72] L. Zhao, S. Liao, Y. Wang, J. Tang, and B. Yuan. Theoretical properties for neural networks with weight matrices of low displacement rank. *ArXiv*, abs/1703.00144, 2017. 1

Supplementary Materials

7. VC dimension and sample complexity

The Vapnik–Chervonenkis (VC) dimension is a measure of the capacity (complexity, expressive power, richness, or flexibility) of a set of functions. In our setting we focus on neural networks where all the weights are of low discriminant rank (LDR) such as the Toeplitz-like, Hankel-like, Vandermonde-like, and Cauchy-like matrices.

7.1. Bounding VC dimension

Theorem 1 For input $x \in \mathcal{X}$ and parameter $\theta \in \mathbb{R}^W$, let $f(x, \theta)$ denote the output of the network. let $\mathcal{F}$ be the class of functions $\{x \to f(x, \theta) : \theta \in \mathbb{R}^W\}$. Let $\mathbf{W}_l$ be the number of parameters up to layer l i.e the total number of parameters in layer (1,2,...,l). we define the depth effective path as:

$$\bar{L} := \frac{1}{W} \sum_{l=1}^{L} W_l, \tag{10}$$

Then the total number of computations units is given as :

$$\mathbf{U} := \sum_{l=0}^{L} n_l \tag{11}$$

Inline with works of [6, 21, **?**, **?**].If k=1, corresponding to piece-wise linear networks, it can be shown that:

$$\mathbf{V}Cdim(sign\ \mathcal{F}) = O(\bar{L}W log(pU)) = O(LW logW). \tag{12}$$

Lemma 1. Let $p_q, ..., p_m$ be polynomials of degree at most d in $n \leq m$ variables, then we define:

$$K := |\{(sign(p_1(x)), ..., (sign(p_m(x))) : x \in \mathbb{R}^n\}|, \tag{13}$$

i.e., if K is the number of possible sign vectors given by the polynomials, then $K \leq 2(2emd/n)^n$. To partition the parameter space $\mathbb{R}^W$ for a fixed input x_j, the output $f(x, \theta)$ on each region in the partition implies $\mathcal{S} = P_1, ..., P_N$ of the parameter $\mathbb{R}^{\mathbb{W}}$.

Hence, we have:

$$K \leq \sum_{j=1}^{N} |\{(signf(x_1, \theta)), ..., signf(x_m, \theta)) : \theta \in P_j\}|, \tag{14}$$

Hence from Lemma 1, we can show that by recursive construction, $\mathcal{S}_{L-1}$ is a partition of $\mathbb{R}^W$ such that for $\mathcal{S} \in \mathcal{S}_{L-1}$ [59]. The network output for input x_j is a fixed polynomial of $\theta \in \mathcal{S}$ which collectively gives:

$$\begin{aligned} K &\leq \sum_{j=1}^{N} |\{(signf(x_1, \theta)), ..., signf(x_m, \theta)) : \theta \in P_j\}| \\ &\leq 2(\frac{2emkd_L}{W_L})^{W_L}, \end{aligned} \tag{15}$$

with the size of $\mathcal{S}_{L-1}$ and equation (6) we get:

$$K \leq \prod_{l=1}^{L} 2(\frac{2emkd_L}{W_L})^{W_L} \tag{16}$$

We can take logarithm and apply Jensen's inequality, with $\bar{W} := \sum_{l=1}^{L} W_l$:

$$\begin{aligned} K &\leq \prod_{l=1}^{L} 2(\frac{2emn_l pd_l}{W_L})^{W_L} \\ log_2 K &\leq L + \sum_{l=1}^{L} W_l log_2 2(\frac{2emn_l pd_l}{W_L}) \\ &= L + \bar{W} \sum_{l=1}^{L} \frac{W_l}{\bar{W}} log_2 2(\frac{2emn_l pd_l}{W_L}) \\ &\leq L + \bar{W} log_2 \left(\sum_{l=1}^{L} \frac{W_l}{\bar{W}} \frac{2emn_l pd_l}{W_L} \right) \\ &= L + \bar{W} log_2 \frac{2emnp \sum_{l=1}^{L} n_l d_l}{\bar{W}} \end{aligned} \tag{17}$$

We bound $\sum_{l=1}^{L} n_l d_l$ using the bound on d_l; since the degree of an LDR matrix d_l is at most:

$$\begin{aligned} \sum_{j=0}^{l-1} n_l d_l &\leq c_1 k^{l-1} \sum_{j=0}^{l-1} n_j^{c_2} \\ &\leq LU c_1 k^{L-1} U^{c_2} \leq U^{c_2 + 2\frac{L}{k}} \end{aligned} \tag{18}$$

where c is a constant $L \leq U$ thus

$$log_2 K \leq L + \bar{W} log_2 \left(\frac{2c_1 empU^{2+c_2 k^L}}{\bar{W}} \right) \tag{19}$$

To bound the VC-dimension, if VCdim(sign $\mathcal{F}$) = m there exists m data points $x_1, ..., x_m$ such that the output of the model can have 2^n sign patterns[62]. The bound on log_2K then implies:

$$\begin{aligned} &VCdim(sign\, \mathcal{F}) \leq L \\ &+ \bar{W} log_2 \left(\frac{2c_1 epU^{2+c_2 k^L} VCdim(sign\, \mathcal{F}}{\bar{W}} \right) \\ &= O(\bar{L}W log(pU) + \bar{L}LW logk) \end{aligned} \tag{20}$$

Hence completing the proof. Since the number of parameters W is around the square root of the number of parameters of a network (e.g doubly block toeplitz based network) with unstructured layers, the sample complexity of an LDR network is much smaller than that of unstructured networks (e.g CNN) which is beneficial for deep networks.

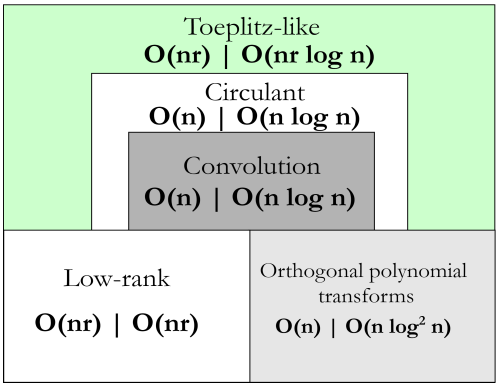

Figure 9: Captions for each class shows the storage cost and operation count for matrix-vector multiplication. Our proposed Toeplitz-like is of lowest rank. compared to circulant, standard convolutional filters, and orthogonal polynomial transforms.

7.2. Space and time complexity.

The proposed DBT-model has a time complexity of $O(nrlogn)$ and the small number of parameters also makes the network perform better with limited amount of training data which is crucial for few-shot learning [34]. We also ran tests on the model backbone with two NVIDIA GeForce GTX 1080 Ti GPU and a batch size of 64. Table 1 reflects the accuracy performance and we see an overall model performance of about 4%. Similar to [**?**], the network used in our test consists of 4 convolutional layers, 1 fully-connected layer and one softmax layer. Rectified linear units (ReLU) are used as the activation units. Images were cropped to 24x24 and augmented with horizontal flips, rotation, and scaling trans-formations. We use an initial learning rate of 0.0001 and train for 800-400-100 epochs with their respective default weight decay. Our efficient DBT-based approach obtains a test error of 6.61%, compared to 5.26% obtained by the conventional CNN model. At the same time, the DBT-based network is 4x more space efficient and 1.2x more time efficient than the conventional CNN-based model.

7.3. Optimization setup.

An SGD optimizer with a momentum of 0.9 and a weight decay of $2\epsilon^{-5}$ was used for our setup. We used a learning rate initialized at 0.0001 with a decay factor of 0.1 applied for all datasets. We trained over 100 epochs for miniImageNet and 200 epochs for both CIFAR-FS and 150 epochs for Stanford Dogs and Stanford Cats, respectively.

7.4. Architecture

The works of [42, 32, 60] used a ResNet12 as backbone for their model, we used a similar structure but replace the convolutional layer with a doubly-block toeplitz matrix the network consists of 4 residual blocks and 3 x 3 kernels. A 2x2 max-pooling layer is applied after each of the first 3 blocks; and a global average-pooling layer is on top of the fourth block to generate feature embeddings. Similar to [58], we used spectral regularization and changed the number of filters from (64, 128, 256, 512) to (64, 160, 320, 640).

Usefulness of DBT regularization The DBT matrix represents a class of structured matrices whose layers interact multiplicatively (A^i, B^i) at $O(nrlogn)$ time as compared to convolutional layers that are linear and unstructured and are implemented in about $O(n^2)$ time [34]. The generic term structured matrix refers to an nxm matrix that can be described in fewer than nm parameters and is capable of fast operation with at most double the displacement rank, which is far simpler for computations [34]. Hence, if $\mathcal{F}$ denotes a class of neural networks comprising of L DBT layers, W total parameters and piece-wise linear activations, we can measure the complexity, expressive power, richness, or flexibility of $\mathcal{F}$ via a measure referred to as the Vapnik–Chervonenkis (VC) dimension [62, 5].

For a simple classification problem of the form: $\{x \rightarrow sign\ f(x) : f \in \mathcal{F}\}$, the VC dimension (VC_{dim}) of the class is expressed as:

$$VC_{dim}(sign\ \mathcal{F}) = O(LWlogW). \tag{21}$$

$\mathcal{V}C_{dim}(\cdot)$ matches the standard bound for unconstrained weight matrices [6, 21, 59].