International Journal of Internet, Broadcasting and Communication Vol.15 No.3 66-72 (2023)
http://dx.doi.org/10.7236/IJIBC.2023.15.3.66

IJIBC 23-3-9

Triplet Class-Wise Difficulty-Based Loss for Long Tail Classification

Yaw Darkwah Jnr.[1], Dae-Ki Kang[2]

[1]*Master's Student, Department of Computer Engineering, Dongseo University*
darkwahyaw@hotmail.com

[2]*Professor, Department of Computer Engineering, Dongseo University*
dkkang@dongseo.ac.kr

Abstract

Little attention appears to have been paid to the relevance of learning a good representation function in solving long tail tasks. Therefore, we propose a new loss function to ensure a good representation is learnt while learning to classify. We call this loss function Triplet Class-Wise Difficulty-Based (TriCDB-CE) Loss. It is a combination of the Triplet Loss and Class-wise Difficulty-Based Cross-Entropy (CDB-CE) Loss. We prove its effectiveness empirically by performing experiments on three benchmark datasets. We find improvement in accuracy after comparing with some baseline methods. For instance, in the CIFAR-10-LT, 7 percentage points (pp) increase relative to the CDB-CE Loss was recorded. There is more room for improvement on Places-LT.

Keywords: *Long-tail classification, Class-wise difficulty, Imbalanced classification, Triplet Loss*

1. INTRODUCTION

Many popular image datasets like ImageNet [1] are balanced datasets. Models do not perform as good as when compared with pre-deployment results. One of the reasons is attributed to the distribution of the datasets used in creating these models. The problem is that data in the real world seldom have a balanced distribution. A more natural distribution is the Pareto or long tail distribution. Data is said to have a long tail when there are a few classes that greatly outnumber the vast majority of classes in terms of their individual frequencies.

To deal with class imbalance, various techniques have been proposed. An easy fix is to sample. Another is using cost-sensitive approaches. These methods make it relatively more costly for a model to predict tail class instances wrongly than for head class instances. These two approaches were found not to produce results that reflect their underlying assumption [2]—few-shot classes are underrepresented and many-shot classes are sufficiently diverse. [2] realized that the general trend of the accuracy did mirror the class-wise frequency. However, the trend was not perfect. There were cases of head classes recording lower accuracies compared with some tail classes. In response, they proposed a class difficulty-based weighting scheme [2].

We employ this weighting strategy, as will be seen in Section 3, in our proposed loss function. Approaches utilizing metric learning [3] and knowledge transfer [4] concepts have been used in handling long tail

Manuscript Received: May. 23, 2023 / Revised: May. 28, 2023 / Accepted: June. 2, 2023
Corresponding Author: dkkang@dongseo.ac.kr
Tel: +82-51-327-1724, Fax: +82-51-327-8955
Professor, Department of Computer Engineering, Dongseo University, Korea

recognition tasks lately. We add another objective in the form of Triplet Loss [5]. Although this may not directly impact the classification, we believe it will help the model learn the features better. This leads to better representation of data. Having data well represented and clearly delineated can enhance classification accuracy.

We introduce the datasets—CIFAR-10-LT, CIFAR-100-LT, and Places-LT [6]—in Section 4. In Section 5, we show the results. TriCDB-CE performed well on the CIFAR datasets but lagged behind in Places-LT.

2. RELATED WORK

There have been diverse ways of grouping approaches to long tail classification. However, these three have been prominently noted: data resampling, class-balanced losses, and metric learning and knowledge transfer.

The most popular approach for dealing with data imbalance is data resampling. This class of solutions strives to create a balance in an imbalance dataset prior to training the model. There is undersampling [7] where some of the samples belonging to the dominant classes are gotten rid of. Conversely, there is also oversampling [8] which increases the number of minority class samples in some fashion such that their frequencies become similar to those of the majority classes. Also, there is instance-balanced sampling where all instances have the same probability of being sampled. In [9], they utilized a sampling approach called progressively-balanced sampling. It starts off with instance-balanced sampling and gradually tilts toward a more class-balanced sampling strategy until it fully uses class-balanced sampling. The downside of using undersampling is that there is a probability of losing some relevant information pertaining to the abundant classes whiles with oversampling, there is a risk of overfitting in the case of the few-shot classes.

Class-balanced losses tackle the imbalance problem by making adjustments to the loss function. It does this by assigning weights. Typically, tail classes get to have larger weights while the head classes get lesser weights. The weighting is often based on the number of instances per class. Class-Balanced (CB) Loss [10] rather ties the weighting to the number of samples deemed effective. Similarly, [2] defined the weights of the classes to be proportional to the classes' difficulties. Aside classes, weights can be tied to the difficulty of individual samples, as is the case for Focal Loss [11]. [12] approached the problem of class imbalance from the perspective of domain adaptation thus, proposed a weighting strategy to that effect. This produces a larger margin for the tail classes while the head classes get a smaller margin. A fundamental assumption of many of this class of solutions is that the minority class lacks diversity, which may not be true always.

Metric learning and transfer learning provide another means of resolving class imbalance. The deficit in tail classes means a higher probability of absence of critical features. Transfer learning thus, transfers knowledge acquired from the many-shot classes to the few-shot classes [13]. With the aid of a Generative Adversarial Network (GAN) [14] translation of head-to-tail class samples in order to create balance has been achieved. Knowledge distillation [15], under knowledge transfer, has also been used for long tail classification. Metric learning is about learning a representation function that represents data such that instances that belong together are close while those that are different are placed further from the group in representation space. The Range Loss [3] is a loss function that seeks to increase the dissimilarity of two class centers in a mini batch, while making classes more consistent by lessening the largest intra-class distances.

There are other approaches which may not fall under any of the above categories. A typical example is [9]. They decoupled model training and made it a two-stage process: representation learning and then classifier retraining. Others focus on designing special classifiers like the τ-normalized classifier [9]. The Balancing GAN [16] generates more examples to restore balance.

The proposed loss function draws from the benefits of cost-sensitive learning and metric learning.

3. THEORY

3.1 CLASS-WISE DIFFICULTY-BASED WEIGHTING

To measure a class' difficulty, the model is run against a balanced validation dataset. The class-wise accuracy is then used to gauge how difficult each class is and how biased the model is towards the majority classes. For a class c, its difficulty at epoch t is found by subtracting its accuracy $A_{c,t}$ on a validation dataset from one. This is denoted by $d_{c,t}$ (1). We then raise the difficulty $d_{c,t}$ to the power of a hyperparameter τ.

$$d_{c,t} = 1 - A_{c,t} \tag{1}$$

The value of τ depends on the imbalance ratio μ and the level of class difficulty. Since class difficulty changes as a model trains, determining an appropriate τ value prior to the start of training may be difficult. In order to calculate the value of τ on the fly, a bias term is computed.

This bias term gauges the level of imbalance regarding the performance of the model. If the model is doing well on all classes, the value of the bias reduces, and has a floor value of close to zero. Conversely, if the model is not performing evenly across all k classes, the bias increases. The bias at an epoch t, represented by b_t as shown in (2), is the ratio of the maximum accuracy among the classes to the minimum accuracy among the classes. We add 0.0001 to the denominator to prevent division by zero. b_t is then passed to the Sigmoid function to get τ. Finally, the value of the weights for each class can be calculated. The weight w is given by the difficulty $d_{c,t}$ with the exponent of τ. (3) shows the formula for calculating the weight for a class c.

$$b_t = \frac{max_{1,2,\ldots k} A_{c,t}}{min_{1,2,\ldots k'} A_{c',t} + 0.0001} - 1 \tag{2}$$

$$w_{c,t} = {d_{c,t}}^{\tau} \tag{3}$$

3.2 CLASS-WISE DIFFICULTY-BASED CROSS-ENTROPY LOSS

Assume a model is being trained on data where x_i is ith the sample and y_i is the ith label. Passing a mini batch from the dataset through the network at epoch t yields an output $z = \{z_{1,t}, z_{2,t}, \ldots, z_{k,t}\}$, denoting the prediction of the model. The Softmax function converts z into a probability distribution $\{p_{1,t}, p_{2,t}, \ldots, p_{k,t}\}$ over k classes. Since the weighting scheme used for this work is the Class-Wise Difficulty-Based weighting from (3), the Class-Wise Difficulty-Based Cross-Entropy (CDB-CE) Loss for a class c is given in (4).

$$L_{CDB-CE} = -w_{c,t} \log p_{c,t} \tag{4}$$

We modify the objective by complementing the CDB-CE Loss by the inclusion of the Triplet Loss.

3.3 TRIPLET LOSS

The objective of the Triplet Loss [5] is to learn appropriate representations of data. It adjusts the weights of the model based on the produced embeddings. The embeddings are the output of the backbone as shown in Figure 1. Triplet Loss seeks to cluster instances of one class around the same region while simultaneously increasing the distance between different classes. The distance between a pair of samples x_i and x_j with feature embeddings f^i and f^j respectively is the squared L2 distance between them i.e., $\left\| f^i - f^j \right\|_2^2$.

To compute the Triplet Loss value for a mini batch of data, sets of triplets must be generated from the batch. Then (5) is used to compute the value. Each triplet has an anchor x_a and a positive sample x_p, both of the

$$L_{Tri} = \max(\|f^a - f^p\|_2^2 - \|f^a - f^n\|_2^2 + \alpha, 0) \tag{5}$$

same class, and a negative sample x_n of another class. This function makes use of a margin constraint α. This ensures x_n is far from the anchor class by a distance with floor value α while x_p gets closer.

3.3 TRIPLET CLASS-WISE DIFFICULTY-BASED CROSS-ENTROPY LOSS

We present this new loss function, Triplet Class-Wise Difficulty-Based Cross-Entropy (TriCDB-CE) Loss to improve classification performance, especially for long tail datasets. This loss functions combines the Triplet Loss and CDB-CE Loss functions to define a new objective for models to train with. This is given as:

$$L_{TriCDB-CE} = \lambda L_{Tri} + L_{CDB-CE} \tag{6}$$

where λ is a hyperparameter to regulate trade-off between the two losses. As a combination of two loss functions, as shown in Figure 1, we expect the two objectives working together to yield better results. While L_{Tri} causes the model to learn better embeddings, L_{CDB-CE} will focus on the classifier. The model weights will not only be tuned for classifying data correctly, they will also be adjusted such that the model will be able to separate the samples into their respective classes thereby improving classification accuracy.

Not all triplets in a batch have their loss value computed. In our experiments, triplets that violate the margin constraint are the ones for which a loss value was calculated and used to update the weights. We do this because there is no need for learning from triplets with well-positioned samples.

4. EXPERIMENTS

We used these datasets: CIFAR-10, CIFAR-100, and Places-LT, as proof of the loss function's performance.

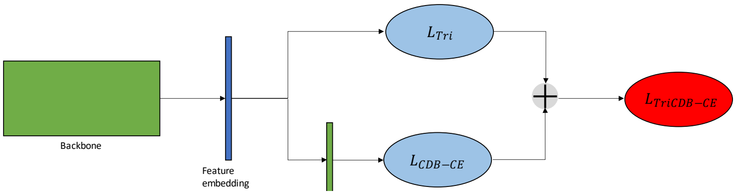

Figure 1. An overview of the proposed framework

CIFAR [17] datasets are popular benchmark datasets. CIFAR-10 has ten classes and CIFAR-100 has100 classes. We created CIFAR-10-LT and CIFAR-100-LT out of the CIFAR-10 and CIFAR-100 datasets respectively based on [2]. The model we used was ResNet-32 [18]. The training regime also followed [2].

The Places-LT [6] is an imbalanced dataset created out of the Places-2 [19] dataset. It has 365 categories or classes. We use an ImageNet-pretrained ResNet-152 [18] as the training model. We followed a training strategy close to that of [6]. However, we used 90 epochs and a batch size of 64. The model was optimized using SGD with an initial learning rate of 0.1. Also, we decayed the learning rate by 0.1 after every 30 epochs.

5. RESULTS AND DISCUSSION

To determine an appropriate value for the Triplet Loss multiplier, we experimented on CIFAR-100-LT (μ = 100). We chose 0.25, 0.50, 0.75, and 1.00. We reported the scores on accuracy, precision, recall, and F1 score for each of the multiplier values as can be seen in Table 1. The accuracies for 0.25 and 1.00 recorded the highest value of 0.41. The scores for recall, 0.41, and F1 score, 0.38, for 1.00 were the highest. The trend observed in the F1 scores suggests a positive correlation with the multiplier. This we attribute to the contribution of the Triplet Loss. Thus, for the remainder of the experiments, λ was set to 1.00. We chose the optimal numeric value of τ for each imbalance ratio as found in [2]. Aside CIFAR-100-LT, we resorted to using the dynamic way for determining a value for τ.

Table 1. Accuracy, Precision, Recall & F1 Score for Different λ Values on CIFAR-100-LT

λ	Accuracy	Precision	Recall	F1 Score
0.25	0.40 ± 0.01	0.44 ± 0.01	0.40 ± 0.01	0.36 ± 0.01
0.50	**0.41** ± **0.01**	**0.45** ± **0.01**	0.40 ± 0.01	0.37 ± 0.01
0.75	0.40 ± 0.01	0.44 ± 0.01	0.40 ± 0.01	0.37 ± 0.01
1.00	**0.41** ± **0.01**	**0.45** ± **0.02**	**0.41** ± **0.01**	**0.38** ± **0.01**

We also experimented with different imbalance ratios. The imbalance ratios used were: 200, 100, 50, and 10. As mentioned earlier, the higher the imbalance ratio the more extreme the imbalance is. This can be observed in the results presented in Table 2. All the metrics suggest an inverse relationship with the imbalance ratio. The imbalance ratio of 10 recorded the highest value across all metrics. This ratio is the most balanced so the model can easily classify the samples, as compared to the more extreme ones like 200.

Table 2. Accuracy, Precision, Recall & F1 Score for Different μ Values on CIFAR-10-LT

μ	Accuracy	Precision	Recall	F1 Score
200	0.66 ± 0.01	0.71 ± 0.01	0.66 ± 0.01	0.65 ± 0.01
100	0.71 ± 0.01	0.75 ± 0.01	0.71 ± 0.01	0.71 ± 0.02
50	0.75 ± 0.01	0.78 ± 0.01	0.75 ± 0.01	0.75 ± 0.01
10	0.84 ± 0.00	0.84 ± 0.01	0.84 ± 0.01	0.84 ± 0.01

In Figure 2, we compare our proposed loss, TriCDB-CE with the CDB-CE. After three runs, on average, TriCDB-CE has an edge over CDB-CE across all metrics excluding precision. TriCDB-CE had the following scores for accuracy, recall, and F1 score: 0.41, 0.41, and 0.38 respectively. The average precision core for both was 0.45. The difference in values across the metrics may seem marginal. This proves the positive impact of the modification of the objective.

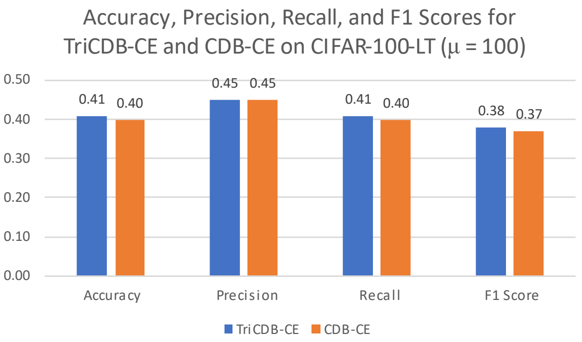

Figure 2. Accuracy, Precision, Recall, and F1 Score for TriCDB-CE and CDB-CE on CIFAR-100-LT (μ = 100)

We compare TriCDB-CE with Focal Loss [11], CFS [20], CB Loss [10], LDAM [21], CDB-CE [2] in Table 3. These experiments were performed on CIFAR-10-LT and CIFAR-100-LT. We considered their accuracies.

For CIFAR-100-LT, it can be seen that TriCDB-CE was at par with LDAM-DRW as both had 0.42. They outperformed the other methods. However, for CIFAR-10-LT, TriCDB-CE significantly outperformed the other methods. It had an accuracy of 0.72. Even though CDB-CE [2] introduced the novel weighting scheme based on class difficulty, adding the Triplet Loss led to an improvement for both datasets.

Table 3. Accuracies on CIFAR-10-LT and CIFAR-100-LT (μ = 100)

Method	CIFAR-10-LT	CIFAR-100-LT
Focal Loss [11]†	0.52	0.38
Class Frequency-Based Sampling [20]†	0.57	0.39
Class-Balanced Loss [10]†	0.58	0.40
LDAM-DRW [21]†	0.59	**0.42**
CDB-CE Loss [2]	0.65	0.40
TriCDB-CE Loss (Ours)	**0.72**	**0.42**

†Results were reported from [2]

In Table 4, we show the results for Places-LT. This table shows accuracy for: many-shot classes, medium-shot classes, few-shot classes, and the entire test set. They compared their results with those of an ImageNet-pretrained ResNet-152 model, Focal Loss [11], Range Loss [3], and FSLwF [22]. The ResNet-152 model was the best-performing method for the many-shot classes with an accuracy of 0.459. OLTR [6] topped the medium-shot metric with 0.370. The outperforming method for the few-shot classes is FSLwF [22], with accuracy close to 30%. Overall, OLTR [6] had the best accuracy of 0.359. Tri-CDB-CE Loss exhibited poor performance in all metrics. We suspect the smaller batch size affected performance.

Table 4. Accuracies on Places-LT

Method	Many-Shot	Medium-Shot	Few-shot	All
Plain Model [18]†	**0.459**	0.224	0.004	0.272
Focal Loss [11]†	0.411	0.348	0.224	0.346
Range Loss [3]†	0.411	0.354	0.232	0.351
FSLwF [22]†	0.439	0.299	**0.295**	0.349
OLTR [6]†	0.447	**0.370**	0.253	**0.359**
CDB-CE [2]	0.395	0.195	0.026	0.234
TriCDB-CE Loss (Ours)	0.390	0.192	0.028	0.231

†Reported from [6]

6. CONCLUSION

We introduced a new loss function suitable for handling long tail datasets. This loss function combines Triplet Loss and Class-Wise Difficulty-Based Cross-Entropy Loss. The Triplet Loss is responsible for the model to learn features to enable good representation learning. The CDB-CE Loss focuses on getting the classification right by assigning weights to the classes based on the perceived difficulty of the class—the more difficult a class is the greater its weight. We performed experiments on CIFAR-10-LT, CIFAR-100-LT, and Places-LT. The results showed that TriCDB-CE outperformed on CIFAR datasets but did not do well on Places-LT. We expect better performance to be possible with further enhancement on the Triplet Loss.

Acknowledgement

This work was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science and ICT (NRF-2022R1A2C2012243) in 2023.

References

[1] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “ImageNet: A large-scale hierarchical image database,” in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, Miami, FL: IEEE, Jun., pp. 248–255.

[2] S. Sinha, H. Ohashi, and K. Nakamura, “Class-Wise Difficulty-Balanced Loss for Solving Class-Imbalance,” in *Proceedings of the Asian Conference on Computer Vision*, Springer International Publishing, 2020, pp. 549–565.

[3] X. Zhang, Z. Fang, Y. Wen, Z. Li, and Y. Qiao, “Range Loss for Deep Face Recognition with Long-Tailed Training Data,” in *2017 IEEE International Conference on Computer Vision (ICCV)*, Venice: IEEE, Oct., pp. 5419–5428.

[4] C. Wei, K. Sohn, C. Mellina, A. Yuille, and F. Yang, “CReST: A Class-Rebalancing Self-Training Framework for Imbalanced Semi-Supervised Learning,” in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Nashville, TN, USA: IEEE, Jun. 2021, pp. 10852–10861.

[5] E. Hoffer and N. Ailon, “Deep Metric Learning Using Triplet Network,” in *Similarity-Based Pattern Recognition*, A. Feragen, M. Pelillo, and M. Loog, Eds., in Lecture Notes in Computer Science, Vol. 9370. Cham: Springer International Publishing, 2015, pp. 84–92.

[6] Z. Liu, Z. Miao, X. Zhan, J. Wang, B. Gong, and S. X. Yu, “Large-Scale Long-Tailed Recognition in an Open World,” in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Jun. 2019.

[7] X.-Y. Liu, J. Wu, and Z.-H. Zhou, “Exploratory Undersampling for Class-Imbalance Learning,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, Vol. 39, No. 2, pp. 539–550, Apr. 2009.

[8] K. E. Bennin, J. Keung, P. Phannachitta, A. Monden, and S. Mensah, “MAHAKIL: Diversity Based Oversampling Approach to Alleviate the Class Imbalance Issue in Software Defect Prediction,” *IEEE Transactions on Software Engineering*, Vol. 44, No. 6, pp. 534–550, Jun. 2018.

[9] B. Kang *et al.*, “Decoupling Representation and Classifier for Long-Tailed Recognition,” in 8th International Conference on Learning Representations, 2020.

[10] Y. Cui, M. Jia, T.-Y. Lin, Y. Song, and S. Belongie, “Class-Balanced Loss Based on Effective Number of Samples,” in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Long Beach, CA, USA: IEEE, Jun. 2019, pp. 9260–9269.

[11] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, “Focal Loss for Dense Object Detection,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 42, No. 2, pp. 318–327, Feb. 2020.

[12] M. A. Jamal, M. Brown, M.-H. Yang, L. Wang, and B. Gong, “Rethinking Class-Balanced Methods for Long-Tailed Visual Recognition From a Domain Adaptation Perspective,” in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Jun. 2020.

[13] X. Yin, X. Yu, K. Sohn, X. Liu, and M. Chandraker, “Feature Transfer Learning for Face Recognition With Under-Represented Data,” in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Jun. 2019.

[14] A. Sahoo, A. Singh, R. Panda, R. Feris, and A. Das, “Mitigating Dataset Imbalance via Joint Generation and Classification,” in *Computer Vision – ECCV 2020 Workshops*, Springer International Publishing, 2020, pp. 177–193.

[15] L. Xiang, G. Ding, and J. Han, “Learning From Multiple Experts: Self-paced Knowledge Distillation for Long-Tailed Classification,” in *Computer Vision – ECCV 2020*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds., in Lecture Notes in Computer Science, Vol. 12350. Cham: Springer International Publishing, 2020, pp. 247–263.

[16] G. Mariani, F. Scheidegger, R. Istrate, C. Bekas, and C. Malossi, “BAGAN: Data Augmentation with Balancing GAN,” 2018*, arXiv:1803.09655*.

[17] A. Krizhevsky and G. Hinton, “Learning multiple layers of features from tiny images,” University of Toronto, Toronto, Ontario, Technical Report, 2009.

[18] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, NV, USA: IEEE, Jun. 2016, pp. 770–778.

[19] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, “Places: A 10 million Image Database for Scene Recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[20] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, “Distributed Representations of Words and Phrases and their Compositionality,” in *Advances in Neural Information Processing Systems*, C. J. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, Eds., Curran Associates, Inc., 2013.

[21] K. Cao, C. Wei, A. Gaidon, N. Arechiga, and T. Ma, “Learning Imbalanced Datasets with Label-Distribution-Aware Margin Loss,” in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché-Buc, E. Fox, and R. Garnett, Eds., Curran Associates, Inc., 2019.

[22] S. Gidaris and N. Komodakis, “Dynamic Few-Shot Visual Learning Without Forgetting,” in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA: IEEE, Jun. 2018, pp. 4367–4375.