Hypersphere-based Weight Imprinting for Few-shot Learning on Embedded Devices

Nikolaos Passalis, Alexandros Iosifidis, Moncef Gabbouj, and Anastasios Tefas

***Abstract*—Weight Imprinting (WI) was recently introduced as a way to perform gradient descent-free few-shot learning. Due to this, WI was almost immediately adopted for performing few-shot learning on embedded neural network accelerators that do not support back-propagation, e.g., Edge TPUs. However, WI suffers from many limitations, e.g., it cannot handle novel categories with multimodal distributions and special care should be given to avoid overfitting the learned embeddings on the training classes, since this can have a devastating effect on classification accuracy (for the novel categories). In this paper, we propose a novel hypersphere-based WI approach that is capable of training neural networks in a regularized, imprinting-aware way effectively overcoming the aforementioned limitations. The effectiveness of the proposed method is demonstrated using extensive experiments on three image datasets.**

***Index Terms*—Weight Imprinting, Few-shot Learning, Edge TPU, Embedded Deep Learning**

I. Introduction

Deep Learning (DL) has achieved remarkable results on a wide range of difficult problems [1], from image and video analysis to natural language processing and visual questioning answering. However, DL models are especially computationally intensive, both during the training and inference. Even though the use of Graphics Processing Units (GPUs) and Tensor Processing Units (TPUs) has led to significant performance improvements, both in terms of training/inference speed and energy consumption, they remain too bulky and energy-intensive to be used in most embedded applications, where significant energy and weight constraints exist [2]. This led to the development of embedded neural network accelerators, specifically designed to accelerate only the inference process, e.g., Edge TPUs. Even though these devices led to tremendous improvements in terms of operations/Watt, most of them suffer from the same limitation: they cannot be used to further train the network using back-propagation. This limits their usefulness under open-world settings, where the models must be able to continuously adapt to emerging categories that were not seen during the training, which is especially challenging for several robotic perception scenarios [3], as well as for a wide range of different multimedia applications [4], [5], [6], [7]. Therefore, the models should be able to draw connections and generalize their knowledge to new novel classes using only a few labeled examples, usually acquired during their interaction with the world. This problem is known as *low-shot learning* or *few-shot learning* [8], [9], [10], [11]. Note that zero-shot learning [12], [13], [14], is a related extreme case of the same problem, in which no labeled samples are available for each category.

It is worth noting that even though several methods have been recently proposed for few-shot learning, only a few of them are suitable for inference-only neural network accelerators. Among them, *Weight Imprinting* (WI) was recently proposed as a way for performing gradient descent-free few-shot learning [15]. WI allows for directly expanding the set of categories which a neural network can recognize by directly *imprinting* a new weight vector in the last layer of the network, without requiring back-propagating through the network. This is done by simply calculating the average embedding vector $\mathbf{w}_i$ of the new (i-th) category using a few training samples and then calculating its similarity with the embeddings $\phi(\mathbf{x})$, extracted through the penultimate layer of the network, where $\mathbf{x}$ is the input to the network. Therefore, the probability that the input $\mathbf{x}$ belongs to the class i can be readily calculated, without performing any additional learning, as:

$$p_i(\mathbf{x}) = \frac{\exp\left(c\mathbf{w}_i^T\phi(\mathbf{x})\right)}{\sum_l \exp\left(c\mathbf{w}_l^T\phi(\mathbf{x})\right)}, \tag{1}$$

where c is a trainable scaling factor (fixed during the inference process). Note that the embedding vectors are normalized to have unit l^2 norm, and, as a result, $\mathbf{w}_i^T\phi(\mathbf{x})$ equals to the cosine similarity between the two vectors. Also, using the scaling factor c ensures that the cosine similarity will range between $-c$ and c, allowing for effectively training the network without imposing a strict lower-bound on the cross-entropy loss [15]. This process found an immediate application on edge accelerators, e.g., Edge TPUs, since it can be readily applied to any neural network simply by extending the last fully connected layer.

WI, despite its immediate adoption, suffers from many limitations. First, it assumes that the distribution of the new categories will be unimodal. This assumption is mostly true for the distribution of classes presented to the network during the training process. However, this is not always the case for novel categories for which the network has not been optimized [15]. Furthermore, the process of imprinting can negatively affect the accuracy of the network for the existing categories, if there is a significant overlap between the class boundaries.

Nikolaos Passalis and Moncef Gabbouj are with the Faculty of Information Technology and Communication, Tampere University, Finland. Anastasios Tefas is with the Department of Informatics, Aristotle University of Thessaloniki, Greece. Alexandros Iosifidis is with the Department of Engineering, Aarhus University, Denmark. E-mail: nikolaos.passalis@tuni.fi, alexandros.iosifidis@eng.au.dk, moncef.gabbouj@tuni.fi, tefas@csd.auth.gr.
This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 871449 (OpenDR). This publication reflects the authors' views only. The European Commission is not responsible for any use that may be made of the information it contains.

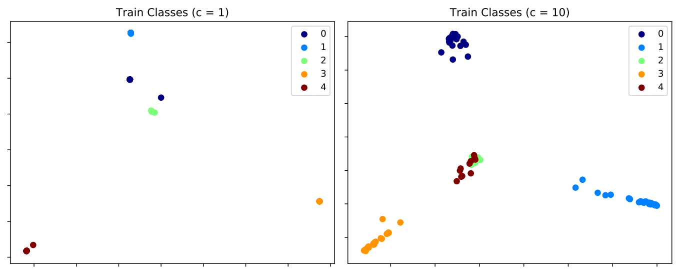

Fig. 1. Weight Imprinting: Visualizing the embeddings for two different initial values of c using the MNIST dataset and the setup described in Section III. The data were visualized by projecting them into 2 dimensions using PCA.

TABLE I
EFFECT OF PARAMETER c ON WEIGHT IMPRINTING

c	1-shot	2-shot	5-shot
1	43.19 ± 3.0	43.26 ± 3.2	43.02 ± 0.6
10	$\mathbf{57.51 \pm 5.8}$	$\mathbf{58.35 \pm 6.4}$	$\mathbf{64.49 \pm 3.2}$
20	54.59 ± 4.0	57.10 ± 2.9	62.30 ± 2.9

Results reported on the MNIST dataset using the setup described in Section III. Classification accuracy (%) on all the classes (novel and training) is reported.

WI does not provide any efficient mechanism for detecting if adding a new category will negatively impact the existing ones. Finally, the impact of the scaling factor c was not thoroughly discussed in [15]. We experimentally found out that the initial value of c can significantly affect the behavior of the model in some cases. For smaller initial values of c the embeddings tend to gather closely around the class prototypes, while for larger initial values of c the embedding vectors are spread around each class prototype $\mathbf{w}_i$. This behavior is illustrated in Fig. 1 (please refer to Section III for more details on the setup used for this experiment). This does not only affect how the embeddings are distributed through the space, but it has also a significant impact on the classification accuracy, as shown in Table I. Using larger initial values of c (up to a point) leads to better classification performance. Therefore, we observe that the embeddings that maintained larger variance around the prototypes allowed for performing better weight imprinting later on.

These observations hint to a direct connection between maintaining the variance of the embeddings around the prototypes and the generalization abilities of a representation/model on unknown classes. This is not a surprising result, since it is well known that overfitted representations almost always lead to worse generalization (after a certain point) [16]. This naturally leads us to the following question: Is it possible to design a representation in which the variance around the prototypes will be deliberately controlled to achieve the perfect balance between overfitting and underfitting instead of relying on early-stopping, implicit regularization or other heuristics to maintain enough variance? Also note that maintaining the variance will allow more information about the in-class similarities/dissimilarities to be encoded in the resulting representation.

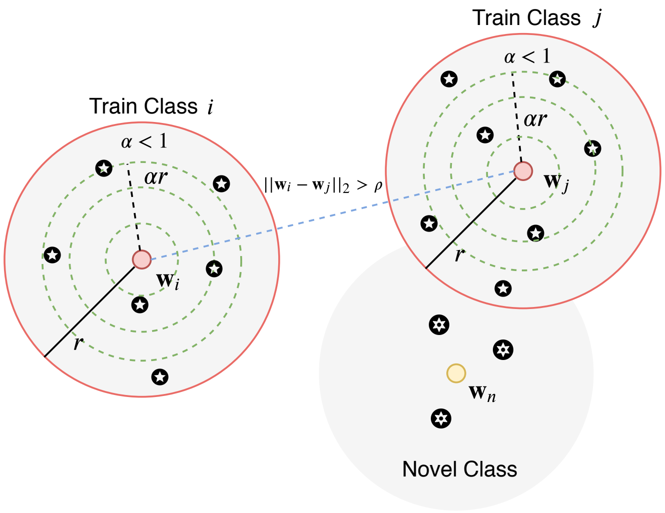

Fig. 2. Hypersphere-based Weight Imprinting: The representation space is constructed in a way that natively supports imprinting, spreads the embeddings in hyperspheres with radius r (the in-class variance is better preserved) and allow for detecting when the imprinting process cannot be performed safely (e.g., potential overlap between the novel class and the j-th class).

Motivated by the aforementioned observations, in this paper we propose a novel weight imprinting method for few-shot learning that can overcome the limitations of WI by: a) learning a regularized representation that maintains the variance around the prototypes in a structured way, while natively supporting weight imprinting and few-shot learning, b) providing a way to directly handle novel categories with multimodal distributions, and c) allowing for detecting beforehand if imprinting a new category will significantly affect the performance of the model for the rest of the categories. An open-source implementation of the proposed method, along with code that can be used to reproduce the conducted experiments, will be available online at [17].

To the best of our knowledge, this is the first work that provides a structured way to perform imprinting-aware neural network training, while at the same time proposing a simple, yet efficient classification scheme for few-shot learning. Compared to the weight imprinting method proposed in [15], the proposed method models each class using a hypersphere, instead of just using a softmax-based classification formulation. This allows for learning a more regularized feature space that leads to better generalization on unknown classes by maintaining the variance around each class prototype, as well as being able to handle multimodal classes by using multiple prototypes per class.

The rest of this paper is structured as follows. First, the proposed method is derived and discussed in Section II. Then, the proposed method is evaluated and compared to regular WI under different scenarios in Section III. Finally, conclusions are drawn in Section IV.

II. PROPOSED METHOD

The proposed method, called *Hypersphere-based Weight Imprinting* (HWI), learns a carefully designed feature space to more effectively support weight imprinting. To this end, it employs a centroid-based loss which uniformly distributes the embedding vectors within a radius r around each prototype (centroid). Furthermore, to ensure that the prototypes are discriminative enough it is required that the minimum distance between two prototypes is at least $\rho > 2r$. This process is illustrated in Fig. 2. After learning a representation that fulfills the aforementioned requirements we can directly classify a new sample, perform gradient descent-free few-shot learning, detect and handle multimodal novel classes and detect intrusion to the existing classes that can lower the performance of the model.

Neural network training: First, we will describe the proposed imprinting-aware training process. Let $\phi(\mathbf{x}) \in \mathbb{R}^m$ be the output of a neural network, where m is the dimensionality of the embeddings extracted from the network, when presented an input sample $\mathbf{x}$. Also, let $\mathbf{w}_i$ be the prototype vector for the i-th class used during the training process and $\mathcal{X}_i$ be the set of samples that belong to the class i. Then, to ensure that the embeddings will be uniformly distributed around each class prototype $\mathbf{w}_i \in \mathbb{R}^m$ we define the appropriate class-induced loss as:

$$\mathcal{L}_c = \frac{1}{N} \sum_{l=1}^{N_C} \sum_{\mathbf{x} \in \mathcal{X}_l} \left(||\phi(\mathbf{x}) - \mathbf{w}_l||_2 - \alpha r \right)^2, \tag{2}$$

where N is the total number of training samples, N_C is the total number of training classes, $||\cdot||_2$ denotes the l^2 norm of a vector and $\alpha \in [0, 1]$ is a number drawn uniformly from the range $[0 \ldots 1]$. During the optimization a different random value is drawn for α for each sample and iteration, leading to a uniform distribution of the embeddings within a radius r from each $\mathbf{w}_i$. Even though this process does not ensure that the full space around $\mathbf{w}_i$ will be occupied, it ensures that the embeddings will be sampled uniformly at various radiuses around the corresponding center, significantly improving the generalization abilities of the representation, as we will also demonstrate later in Section III. Note that by setting $r = 0$, the loss $\mathcal{L}_{class}$ degenerates to the regular center loss [18]. Furthermore, to further model the uncertainty regarding the class prototypes, we can use Gaussian noise to corrupt the prototypes as $\tilde{w}_i = w_i + \mathcal{N}(0, \sigma)$. The effect of varying the radius r is demonstrated in the toy example of Figure 3. For radius equal to $r = 0$ the model compresses the embeddings around the prototypes (whether a collapse will happen or not depends on the ability of the neural network to overfit the data), while for larger values the embeddings are spread around the prototypes as desired, ensuring that the in-class variance is maintained.

At the same time, each prototype $\mathbf{w}_i$ is required to be at a distance of at least ρ from each other prototype (to ensure that there is no overlapping between the hyperspheres that enclose the embeddings of each class). To this end, we also define the prototype loss as:

$$\mathcal{L}_p = \frac{1}{N_C(N_C - 1)} \sum_{i=1}^{N_C} \sum_{j=1, i \neq j}^{N_C} \max(0, \rho - ||\mathbf{w}_i - \mathbf{w}_j||_2). \tag{3}$$

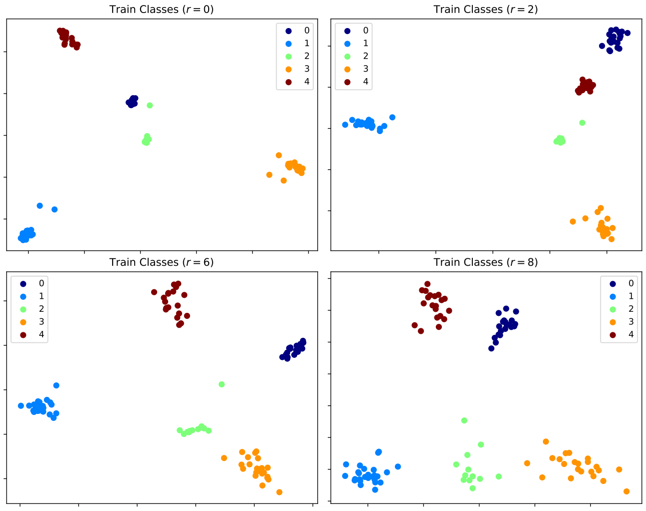

Fig. 3. Visualizing the embeddings learned for different radiuses r using the MNIST dataset and the setup described in Section III. The data were visualized by projecting them into 2 dimensions using PCA.

Let $\mathbf{W}$ denote the parameters of the model $\phi(\mathbf{x})$, along with the prototype vectors $\mathbf{w}_i$. The proposed method employs gradient descent to optimize these parameters in order to minimize both the class-induced loss and the prototype loss: $\mathcal{L} = \mathcal{L}_c + \gamma \mathcal{L}_p$, where γ is a hyper-parameter that alters the weight of the prototype loss (set to 1 for all the experiments conducted in this paper). Therefore, at each training iteration, a batch of data is sampled and the parameters of the model are updated as:

$$\Delta \mathbf{W} = -\frac{\partial \mathcal{L}}{\partial \mathbf{W}.} \tag{4}$$

Classifying a sample: To classify an input sample we can directly choose the class that corresponds to the prototype with the smaller distance to the extracted embedding $\phi(\mathbf{x})$. The network can be used in a similar fashion as a one trained using the softmax activation simply by using a final classification layer that calculates the membership value for each prototype/class probabilities as:

$$p_i(\mathbf{x}) = \frac{1}{1 + ||\phi(\mathbf{x}) - \mathbf{w}_i||_2}. \tag{5}$$

Performing few-shot learning: To perform few-shot learning we can simply augment the final classification layer with an additional prototype vector $\mathbf{w}_n$ calculated as:

$$\mathbf{w}_n = \frac{1}{|\mathcal{X}_n|} \sum_{\mathbf{x} \in \mathcal{X}_n} \phi(\mathbf{x}), \tag{6}$$

where $\mathcal{X}_n$ is the set that contains the training samples for a novel category. Note that similarly to regular WI, no gradient descent-based optimization is required for extending the classifier to support novel classes. However, as we further demonstrate in Section III, the regularized nature of the learned feature space leads to significantly better performance compared to regular WI.

Detecting and handling multimodal novel categories: There are several ways to detect if the distribution of a novel class is indeed multimodal, including, but not limited to, the bandwidth test [19], and the runt test [20]. Any of these approaches can be combined with the proposed method to allow for detecting whether the distribution of a class is multimodal. Furthermore, the proposed hypersphere-based formulation also provides a straightforward way to discover multimodal classes: the embedding vectors extracted for a novel category are clustered and the distance between the cluster centroids is measured. If we detect centroids that are at distance greater than r from each other, then a hypersphere with radius of r cannot enclose the embeddings of the novel class. To address this, we can simply add one or more prototypes (according to the number of centers that are at distance greater than r) to model the distribution of the novel class. In this way, one class can be represented using more than one prototype. On the other hand, if the centers of the clusters are within a radius of r, then we assume that proposed classification scheme can directly handle the distribution of the novel class (even though there is no guarantee that the distribution is not multimodal). The proposed way of handling multimodal classes is straightforward to implement when the proposed hypersphere imprinting approach is used and allows for improving the accuracy of the proposed method, as further demonstrated in Section III.

Detecting intrusions: We can directly detect if a novel prototype $\mathbf{w}_n$ will intrude an existing class simply by measuring the distance between the prototype and each other prototype as: $d_i = ||\mathbf{w}_n - \mathbf{w}_i||_2$. If $\min_i(d_i) < r$, then at least one hypersphere of an existing class will be intruded by the novel class. This can be regarded as a hint that imprinting will fail and cannot be used to support the specific novel class. In this case, the model should be probably re-trained off-line using another few-shot learning method.

III. Experimental Evaluation

The proposed method was evaluated using three different image datasets, the MNIST dataset [21], the Caltech-UCSD Birds 200-2001 (CUB-200-2011) dataset [22], and the Animals with Attributes (AwA2) dataset [23]. For the MNIST dataset the first 5 classes were used to train the model, while the remaining 5 classes were used for few-shot learning and evaluation. For the CUB-200-2011 dataset, we followed the evaluation setup proposed in [15]. For the AwA2 dataset [23] the first 40 classes were used for training the model, and the rest of the classes were employed for evaluating the performance of the proposed method. The performance evaluation was repeated 5 times using different training samples for the novel classes (except for the hyper-parameter evaluation experiments) and the mean and standard deviation is reported. For the MNIST dataset the employed neural network was composed of a 3×3 convolutional layer with 32 filters, followed by a 2×2 max pooling layer, another 3×3 convolutional layer with 64 filters, an additional 2×2 max pooling layer and a fully connected layer with 256 neurons. An InceptionV1 architecture [24], pretrained on the Imagenet dataset was used for the CUB-200-2011 dataset. The last fully connected layer of the InceptionV1 architecture was discarded and replaced with a fully connected layer with 256 hidden neurons, following the approach used in [15]. For the AwA2 dataset we used a pretrained ResNet101 to extract feature vectors (as described in [23]) that were then fed to two fully connected layers with 2048 and 512 neurons respectively. The *relu* activation function was used for all the layers. The models were trained using the Adam optimizer with a learning rate of 10^{-3} for 20 training epochs for the MNIST dataset and of 10^{-4} for 20 training epochs for the AwA2 dataset. For the CUB-200-2011 dataset, the added fully connected layers were pre-trained for 10 epochs using a learning rate of 10^{-3}, followed by 10 additional full training epochs using a learning rate of 10^{-4}.

TABLE II
Evaluating the effect of the hyper-parameters ρ on the 2-shot learning accuracy of the proposed imprinting method

Min. Distance ρ	**Novel Split**	**Combined Split**
1	56.28	77.79
2	56.24	77.42
5	58.53	78.66
10	**61.02**	**80.06**
20	53.82	76.51

TABLE III
Evaluating the effect of the hyper-parameters r on the 2-shot learning accuracy of the proposed imprinting method

Radius r	**Novel Split**	**Combined Split**
0	61.02	80.06
4	64.51	80.59
5	**69.57**	**81.55**
6	66.86	78.26

First, we evaluated the effect of altering the minimum distance ρ between the different class prototypes $\mathbf{w}_i$, while keeping the radius fixed to $r = 0$. The results are presented in Table II. Increasing the minimum distance between the class prototypes seems to have a positive effect on the classification accuracy, both for the novel split (denoted by "Novel") and the combined split of novel and training classes (denoted by "All"). This was expected since the learned representation is not capable of perfectly collapsing the embeddings to the corresponding prototypes, even though the radius was set to $r = 0$. Therefore, keeping a quite large margin between the different prototypes helps to reduce the risk of wrongly classifying a sample.

Next, we also evaluated the effect of altering the radius r on the learned representation. In Section II it was conjectured that spreading the training embeddings in a hypersphere of radius r will have a positive regularization effect on the learned representation by allowing the model to capture and better model the in-class variations. Indeed, as demonstrated in Table III, the classification accuracy increases by more than 14% for the novel split, and by 1.8% for the combined split. This confirms our hypothesis that collapsing the embeddings to the class centers, without any form of regularization, can significantly reduce the classification accuracy, especially when dealing with classes that were not seen during the training process and using powerful models that can overfit the training data.

TABLE IV
MNIST: EVALUATING THE ACCURACY OF IMPRINTING METHODS ON THE COMBINED NOVEL AND TRAINING CATEGORIES.

Method	Split	1-shot	2-shot	5-shot
WI	All	57.51 ± 5.8	58.35 ± 6.4	64.49 ± 3.2
HWI-	All	73.27 ± 4.7	$\mathbf{79.46 \pm 1.5}$	82.88 ± 1.8
HWI	All	$\mathbf{73.66 \pm 3.9}$	79.09 ± 6.0	$\mathbf{84.23 \pm 1.3}$

TABLE V
MNIST MULTIMODAL: EVALUATING THE ACCURACY OF IMPRINTING METHODS ON THE NOVEL CATEGORIES.

Method	Thres.	2-shot	4-shot	10-shot
WI	-	55.81 ± 11.7	65.17 ± 12.3	78.89 ± 2.6
HWI-	-	42.07 ± 6.5	47.40 ± 9.6	55.47 ± 2.4
HWI	-	60.95 ± 4.0	71.03 ± 5.5	75.69 ± 1.0
HWI-M	5	62.36 ± 3.4	70.00 ± 5.3	75.69 ± 1.0
HWI-M	4	66.47 ± 5.0	71.52 ± 2.3	75.11 ± 2.8
HWI-M	3	$\mathbf{66.78 \pm 4.9}$	$\mathbf{73.99 \pm 5.6}$	$\mathbf{82.69 \pm 1.5}$

Next, we evaluated the proposed method using a 1-shot, 2-shot and 5-shot evaluation protocol on the MNIST dataset. The results are reported in Table IV. Two variants of the proposed method were evaluated: HWI-, where $r = 0$ and $\sigma = 0$ were used, and HWI, where $r = 5$ and $\sigma = 0.05$ were used. The proposed method was also compared to the plain Weight Imprinting (WI) approach [15], using an initial scaling value of $c = 10$ (following the results of Table I). Again, it was confirmed that using the proposed variance preserving approach improves the performance over simply using a center-based loss, allowing for outperforming the regular WI method. It is worth noting that the accuracy for all the evaluated methods is relatively low compared to the state-of-the-art, since neither WI or the proposed method perform any kind of optimization according to a discriminative objective.

The proposed approach for handling multimodal novel classes, abbreviated as "HWI-M", was also evaluated using an additional multimodal split of the MNIST dataset. This split was compiled by merging two succeeding classes into one, e.g., "0" and "1" were merged into a new class, "2" and "3" into another, and so on. Then, the three first classes (digits 0 to 5) were used for training and the remaining two of them (6 to 9) for evaluating the few-shot learning performance. The evaluation results are reported in Table V. The employed threshold was used to detect whether a class distribution is multimodal (by clustering the training data into two clusters and measuring the distance between the resulting centroids). If the distance of the resulting centers was greater than the specified threshold, then two prototypes were used per novel class. Again, note that the proposed variance-preserving variant of HWI greatly outperforms the HWI- variant. Using the multimodal variant HWI-M allows for further improving the accuracy of the proposed method, outperforming all the other evaluated approaches.

TABLE VI
CUB-200-2011: EVALUATING THE ACCURACY OF VARIOUS METHODS ON THE NOVEL CATEGORIES.

Method	1-shot	2-shot	5-shot
Generator [11]	18.56	19.07	20.00
Match. Nets [25]	13.45	14.75	16.65
WI [15]	21.26	28.69	39.52
WI + FT [15]	18.67	30.17	46.08
WI	24.88 ± 1.14	33.92 ± 1.66	45.28 ± 1.40
HWI-	27.21 ± 1.63	35.49 ± 1.05	45.45 ± 1.11
HWI	$\mathbf{27.61 \pm 1.01}$	$\mathbf{36.33 \pm 1.25}$	$\mathbf{46.63 \pm 1.10}$

Results for Generator + Classifier approach ("Generator") and Matching Networks ("Match. Nets") are as reported in [15]. "WI + FT" refers to results obtained in [15] using the proposed combined weight imprinting and fine-tuning approach.

TABLE VII
CUB-200-2011 MULTIMODAL: EVALUATING THE ACCURACY OF VARIOUS IMPRINTING METHODS ON THE MULTIMODAL SPLIT OF THE CUB-200-11 DATASET (NOVEL CATEGORIES).

Method	1-shot*	2-shot*	5-shot*
WI	26.05 ± 1.30	26.37 ± 0.92	27.64 ± 0.69
HWI-	21.39 ± 0.68	22.29 ± 0.35	23.77 ± 0.54
HWI	26.93 ± 1.26	28.25 ± 1.34	29.66 ± 0.57
HWI-M	$\mathbf{27.67 \pm 1.63}$	$\mathbf{29.69 \pm 0.44}$	$\mathbf{33.56 \pm 0.62}$

*n-shot refers to the number of samples n used for each modality.

The proposed method was also evaluated using the CUB-200-2011 dataset, which allows for a direct comparison with the original weight imprinting approach, as presented in [15]. The experimental results are reported in Table VI. The proposed method ($\rho = 30, r = 15$) was also compared to three other baseline few-shot learning approaches, the Generator + Classifier approach, as proposed in [11], the Matching Networks [25], as well as the Weight Imprinting + Fine-tuning approach (which requires further optimization of the network for the novel classes compared to plain weight imprinting), as proposed in [15]. Note that the proposed method outperforms the rest of the evaluated approaches. Note that despite using the same network architecture as the one proposed in [15], i.e., an InceptionV1 model [24], our implementation also leads to slightly better accuracy for the original WI approach, possibly due to the different initialization of the network. However, the proposed HWI still outperforms the plain WI, regardless the used setup. Similar results are also obtained when the multimodal split of the CUB-200-2011 dataset is used, as reported in Table VII. This split was compiled by merging each set of 10 successive classes of the original dataset, leading to 10 classes that are used for training the models and 10 classes for evaluating the imprinting performance. The proposed multimodal-aware imprinting approach again leads to higher accuracy over all the evaluated methods, confirming its ability to handle multimodal novel classes.

Finally, we also evaluated the performance of the proposed method using the AwA2 dataset. The results are reported in Table VIII. As before, the proposed method leads to significant performance improvements over the plain WI method, while it still outperforms the HWI- methods. The smaller differences between HWI- and HWI can be possibly attributed the to the smaller learning capacity of the employed network (the risk of overfitting the representation is higher when more powerful networks are employed). Note that slightly different parameters were used for the HWI method in this experiment:

TABLE VIII
AwA2: Evaluating the accuracy of imprinting methods on the combined novel and training categories split.

Method	1-shot	2-shot	5-shot
WI	51.03 ± 3.71	61.33 ± 2.73	75.23 ± 1.85
HWI-	54.55 ± 3.31	68.44 ± 3.17	76.95 ± 2.12
HWI	$\mathbf{56.14 \pm 2.70}$	$\mathbf{70.16 \pm 2.63}$	$\mathbf{77.85 \pm 1.84}$

TABLE IX
Class intrusion robustness evaluation on the AwA2 dataset

δ	1-shot	2-shot	5-shot
0.1	55.60 ± 3.06	69.69 ± 3.33	77.02 ± 2.81
0.5	55.73 ± 3.02	70.02 ± 2.85	77.73 ± 1.95
0.8	56.02 ± 2.78	70.06 ± 2.79	77.75 ± 1.93
1	56.10 ± 2.73	70.16 ± 2.64	77.83 ± 1.86
No distractors	56.10 ± 2.73	70.16 ± 2.64	77.83 ± 1.86

$\rho = 20$, $r = 10$, and $\sigma = 0$.

To demonstrate that the proposed method is also robust to novel prototypes/distractors that are at a distance of at least r from the existing prototypes, we also performed one additional experiment. We randomly generated 10 additional prototypes, each one having a distance between δr and $\delta r + 1$ from the closest prototype, where δ is a hyper-parameter that controls the intrusion (for $\delta > 1$ there will be no class intrusion). The results are in Table IX. The goal of this experiment was to demonstrate that the existing prototypes can be effectively "protected" from distractors, if they exist in a distance greater than r from the prototypes. Indeed, for $\delta = 1$ the accuracy of the model has not been reduced, confirming the robustness of the proposed method to distractors (given that the appropriate distance from the existing prototypes is maintained to avoid intrusions). For smaller values of δ, even though the existing hyperspheres are intruded by the distractors, the effect on the classification accuracy is minimal.

IV. Conclusions

In this paper we proposed a novel hypersphere-based weight imprinting approach that maintains all the advantages of regular WI [15], i.e., it is able to readily extend a pretrained neural network to classify samples from novel categories simply by adding new weight vectors in the final classification layer without requiring to perform any form of back-propagation to this end. At the same time, the proposed method was capable to overcome significant limitations of WI by being able to learn regularized representations that provide better generalization for classes which were not seen during the training and provides a straightforward way to directly handle novel categories with multimodal distributions. The proposed method was extensively evaluated on three image datasets, outperforming the regular WI approach.

References

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, p. 436, 2015.

[2] A. Dundar, J. Jin, B. Martini, and E. Culurciello, "Embedded streaming deep neural networks accelerator with applications," *IEEE Trans. on Neural Networks and Learning Systems*, vol. 28, no. 7, pp. 1572–1583, 2016.

[3] S. Murata, Y. Yamashita, H. Arie, T. Ogata, S. Sugano, and J. Tani, "Learning to perceive the world as probabilistic or deterministic via interaction with others: A neuro-robotics experiment," *IEEE Trans. on Neural Networks and Learning Systems*, vol. 28, no. 4, pp. 830–848, 2015.

[4] C. Yan, Y. Zhang, J. Xu, F. Dai, J. Zhang, Q. Dai, and F. Wu, "Efficient parallel framework for hevc motion estimation on many-core processors," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 24, no. 12, pp. 2077–2089, 2014.

[5] C. Yan, H. Xie, J. Chen, Z. Zha, X. Hao, Y. Zhang, and Q. Dai, "A fast uyghur text detector for complex background images," *IEEE Trans. on Multimedia*, vol. 20, no. 12, pp. 3389–3398, 2018.

[6] C. Yan, L. Li, C. Zhang, B. Liu, Y. Zhang, and Q. Dai, "Cross-modality bridging and knowledge transferring for image understanding," *IEEE Trans. on Multimedia*, 2019.

[7] C. Yan, Y. Tu, X. Wang, Y. Zhang, X. Hao, Y. Zhang, and Q. Dai, "Stat: spatial-temporal attention mechanism for video captioning," *IEEE Trans. on Multimedia*, 2019.

[8] S. Gidaris and N. Komodakis, "Dynamic few-shot visual learning without forgetting," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2018, pp. 4367–4375.

[9] S. Qiao, C. Liu, W. Shen, and A. L. Yuille, "Few-shot image recognition by predicting parameters from activations," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2018, pp. 7229–7238.

[10] Q. Cai, Y. Pan, T. Yao, C. Yan, and T. Mei, "Memory matching networks for one-shot image recognition," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2018, pp. 4080–4088.

[11] B. Hariharan and R. Girshick, "Low-shot visual recognition by shrinking and hallucinating features," in *Proc. IEEE Int. Conf. on Computer Vision*, 2017, pp. 3018–3027.

[12] R. Socher, M. Ganjoo, C. D. Manning, and A. Ng, "Zero-shot learning through cross-modal transfer," in *Proc. Advances in Neural Information Processing Systems*, 2013, pp. 935–943.

[13] Y. Yu, Z. Ji, J. Guo, and Y. Pang, "Transductive zero-shot learning with adaptive structural embedding," *IEEE Trans. on Neural Networks and Learning Systems*, vol. 29, no. 9, pp. 4116–4127, 2017.

[14] Z. Ji, Y. Sun, Y. Yu, Y. Pang, and J. Han, "Attribute-guided network for cross-modal zero-shot hashing," *IEEE Trans. on Neural Networks and Learning Systems*, 2019.

[15] H. Qi, M. Brown, and D. G. Lowe, "Low-shot learning with imprinted weights," in *Proc. Conf. on Computer Vision and Pattern Recognition*, 2018, pp. 5822–5830.

[16] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[17] "Pytorch-based impleemnetaiton of hypersphere-based weight imprinting," https://github.com/passalis/hypersphere_imprinting.

[18] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, "A discriminative feature learning approach for deep face recognition," in *Proc. European Conf. on Computer Vision*, 2016, pp. 499–515.

[19] B. W. Silverman, "Using kernel density estimates to investigate multimodality," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 43, no. 1, pp. 97–99, 1981.

[20] J. A. Hartigan and S. Mohanty, "The runt test for multimodality," *Journal of Classification*, vol. 9, no. 1, pp. 63–70, 1992.

[21] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner *et al.*, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[22] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The caltech-ucsd birds-200-2011 dataset," 2011.

[23] Y. Xian, C. H. Lampert, B. Schiele, and Z. Akata, "Zero-shot learning-a comprehensive evaluation of the good, the bad and the ugly," *IEEE Trans. on Pattern Analysis & Machine Intelligence*, 2018.

[24] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2015, pp. 1–9.

[25] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra *et al.*, "Matching networks for one shot learning," in *Proc. Advances in Neural Information Processing Systems*, 2016, pp. 3630–3638.