Generalized and Incremental Few-Shot Learning by Explicit Learning and Calibration without Forgetting

Anna Kukleva[1] Hilde Kuehne[2,3] Bernt Schiele[1]

[1]MPI for Informatics, Saarland Informatics Campus [2]CVAILab, Goethe University Frankfurt [3] MIT-IBM Watson AI Lab, Cambridge

Abstract

Both generalized and incremental few-shot learning have to deal with three major challenges: learning novel classes from only few samples per class, preventing catastrophic forgetting of base classes, and classifier calibration across novel and base classes. In this work we propose a three-stage framework that allows to explicitly and effectively address these challenges. While the first phase learns base classes with many samples, the second phase learns a calibrated classifier for novel classes from few samples while also preventing catastrophic forgetting. In the final phase, calibration is achieved across all classes. We evaluate the proposed framework on four challenging benchmark datasets for image and video few-shot classification and obtain state-of-the-art results for both generalized and incremental few shot learning.

1. Introduction

In this paper we are interested in two practically important learning scenarios, namely generalized few-shot learning (GFSL) [13, 36, 32, 39] and incremental few-shot learning (IFSL) [43, 6]. In both scenarios it is possible to learn a performant classifier for a set of base classes for which many training samples exist. However, for the novel classes, only few training samples are available such that a *novel class learning* is challenging. Additionally, in generalized few-shot learning and in incremental learning it is important to prevent *catastrophic forgetting* of the base classes during novel class learning. Last, but not least, *classifier calibration* across classes has to be addressed, due to the imbalance in the amount of training samples. While previous work focuses on addressing a subset of these challenges [13, 36, 23, 43, 6], in this paper we aim to address all three.

To this end, we propose a three phase framework to explicitly address these challenges. The first phase is devoted to general representation learning as in previous work [13, 36, 47]. Here, we utilize a large base dataset for pretraining and obtain high performance for base classification. In the second phase we concentrate on learning novel classes. In contrast to the prior work, we pay special attention to training a calibrated classifier for the novel classes while simultaneously preventing catastrophic forgetting for the base classes. More specifically we propose base-normalized cross entropy that amplifies the softmax output of novel classes to overcome the bias towards the base classes, and simultaneously enforce the model to preserve previous knowledge via explicit weight constraints. In the third phase we address the problem of calibrating the overall model across base and novel classes. In Fig. 1 we show how the model develops during all three phases by plotting the test accuracy of base and novel classes in the

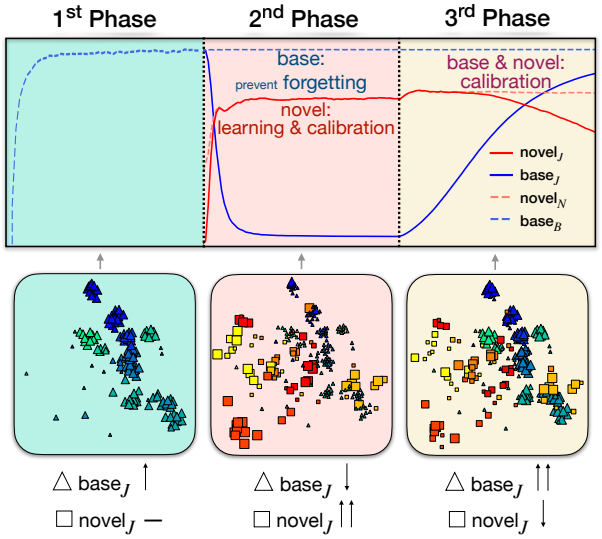

Figure 1: Overview of the performance of our framework during different phases. *J* indicates performance in the joint space, *B* and *N* denote performance in the base and novel spaces respectively. During the 1^{st} phase we train the model on the base classes. During the 2^{nd} phase we try to achieve a high performance on novel classes in the joint space and prevent forgetting of base classes. In the 3^{rd} phase we calibrate the two classifiers and achieve balanced performance between base and novel classes. T-SNE plots show the performance of the test samples at each phase. The symbol size shows the confidence of the model for the sample.

https://github.com/annusha/LCwoF

separate and joint spaces. The contributions of this work are as follows:

(1) A framework to explicitly address the problems of generalized few-shot-learning by balancing between learning novel classes, forgetting base classes and calibration across them in three phases;

(2) Base-normalized cross-entropy to overcome the bias learned by the model on the base classes in combination with weight constraints to mitigate the forgetting problem in the second phase;

(3) An extensive study to evaluate the proposed framework on images and videos showing state-of-the-art results for generalized and incremental few shot learning.

2. Related Work

Generalized Few-Shot Learning (GFSL) Many approaches for few-shot learning (FSL) rely on a meta-learning paradigm to quickly adapt a method to new underrepresented samples [40, 46, 11, 42, 29, 42]. Such models can be hard to extend to a generalized setup since they do not explicitly learn classification of base classes and do not consider extreme imbalance. Some recent work on few-shot learning additionally examine a generalized setup [51, 27] showing a significant drop in performance in the joint space.

First work [15, 47] on GFSL propose to hallucinate extra samples based on intra-class variations of base classes. Later, Gidaris *et al*. [13] propose an attention based weights generator for few-shot classes and promote cosine normalization to unify recognition of base and novel classes. Concurrently, Qi *et al*. [32] propose weight imprinting that is also based on the idea of cosine normalization. The technique is widely used to avoid the explicit calibration of magnitudes [18, 36, 39, 30, 14, 35]. In our work we exploit the bias in base classification weights to give the model impetus to learn novel classes in the joint space. Ren *et al*. [36] propose to use an attention attractor network [53] to regularize the learning of novel classes with respect to accumulated base attractor vectors. The above works are based on meta-learning frameworks, consequently they can be dependent on the number of novel classes. In contrast, Shi *et al*. [39] propose a graph-based framework to model the relationship between all classes that can be trained end-to-end. GFSL receives attention in the video domain as well. Previous work propose to enlarge the training data for few novel classes by means of a generative adversarial network [23] or to retrieve similar data from a large-scale external dataset [49].

Incremental Few-Shot Learning (IFSL) Class incremental learning (CIL) [25, 26, 5, 18, 35] addresses the problem of a continuously growing classification space, where each set of novel classes extends previously observed classes. The major problem of incremental learning, catastrophic forgetting [28], is caused by limiting the access to already seen data while each novel class is provided with a large train set. Due to the ample number of training samples, on the one hand, authors [30, 1, 54] propose to separate softmax classification into several subspaces to balance learning. On the other hand, some work addresses it with bias removal techniques [48, 3, 4] by training batch of additional hyperparameters [48], using dual memory [3] or with post-processing [4]. On the contrary, we benefit from the joint space to overcome the deficiency of data and to learn a stronger classifier for novel classes without introducing any additional parameters.

A new task that combines FSL and CIL is incremental few-shot learning (IFSL) [6, 43, 2]. GFSL can be seen as a subproblem of IFSL with only one incremental learning set of novel classes. Tao *et al*. [43] propose to preserve the topology of the feature manifold via a neural gas network. Chen *et al*. [6] use a non-parametric method based on learning vector quantization in deep embedding space to avoid imbalanced weights in a parametric classifier. In this work we approach the problem from the perspective of classic parametric classification models [16, 22] that recently are shown to be effective in FSL [44] and CIL [31].

3. LCwoF-Framework

In the following we introduce the general setting and the motivation of our method based on four separate performance measures introduced below. We then discuss the second phase training as the most crucial part to achieve strong performance for both novel and base classes, followed by our third phase training. Finally, we discuss how to generalize our method to incremental few shot learning.

Setting: In both generalized and incremental few-shot learning we have a set of base classes C_B with many training samples. Additionally, we have one or several sets of novel classes C_N with only few training samples. In generalized few-shot learning we have just one set of novel classes, while in incremental few-shot learning we have a sequence of such sets. In the following, to keep the notation simple, we discuss our approach based on a single set of novel classes, whereas in incremental learning the approach is applied to the sequence of such sets of novel classes. Note that incremental few-shot learning in our work is the same as the few-shot class incremental learning in [6, 43].

Performance measures and approach: In few-shot learning we are interested to achieve best performance for both base and novel classes simultaneously. Therefore, in order to monitor performance for both sets of classes, we are considering four different measures (see Fig. 3) First, we denote $B_{/B}$ the classification performance of *base* samples in the space of only *base* classes C_B, and $N_{/N}$ the classification performance of *novel* samples in the space of only *novel* classes C_N. More importantly, we are interested in the performance in the *joint* (J) space where both base and novel classes are accounted for simultaneously $C_B \cup C_N$. For

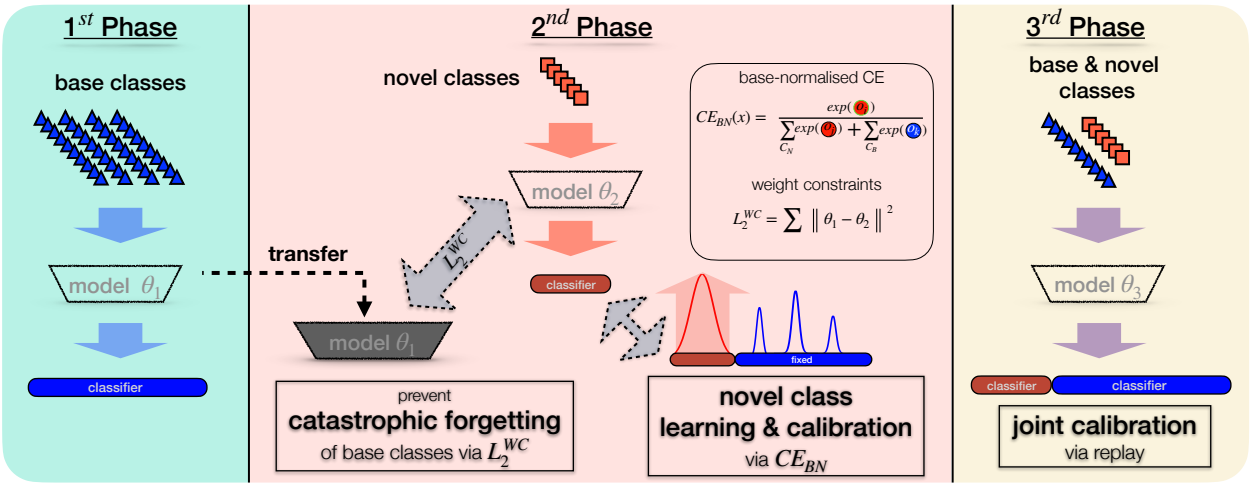

Figure 2: Overview of our framework. To achieve balanced performance on base and novel classes we deal with three problems *learning novel classes*, *catastrophic forgetting*, and *calibration* that we address in different phases of our framework. In the 1^{st} phase we pretrain model on base classes with abundant data. During the 2^{nd} phase we employ L_2^{WC} weight constraints to preserve knowledge and base-normalized cross entropy (CE_{BN}) to calibrate learning of novel classes in the joint space with base classes. In the 3^{rd} phase we calibrate the performance with the balanced replay of novel and base samples.

this we consider the performance of base and novel samples separately in the *joint* space, that is $N_{/J}$ and $B_{/J}$. We prefer these two measures rather than using only the joint performance in joint space due to the imbalance of the number of classes [18, 3, 48] with $|C_B| \gg |C_N|$ (e.g. 64 base vs. 5 novel).

These measures are directly related to the three challenges mentioned above: novel class learning is measured by $N_{/N}$ and $N_{/J}$, catastrophic forgetting by $B_{/B}$ and $B_{/J}$, while calibration is related to $B_{/J}$ and $N_{/J}$. While ideally we would like to address all the measures simultaneously, we found this to be difficult in practice. Instead, during the first phase of our framework, we optimize for $B_{/B}$. In the second phase, during novel class learning, we are aiming for a calibrated classifier for the novel classes and thus optimize for both $N_{/N}$ and $N_{/J}$, instead of only $N_{/N}$ as in standard few-shot learning. Simultaneously, we aim to prevent catastrophic forgetting by an additional weight regularization that keeps $B_{/B}$ high (see Fig. 1). In the third and last phase we aim to calibrate across novel and base classes and thus optimize for both $B_{/J}$ and $N_{/J}$.

Model parameters: We denote the backbone parameters as θ_1, θ_2, and θ_3 for the first, the second, and the third phase respectively. As classifier we use a linear classification layer without bias that we train on the top of the backbone. Practically, during the second phase we introduce a classification layer for novel classes. To evaluate performance in the joint space we concatenate the output of the two classifiers before the normalization. o_i denotes the output logit of the model for the classification into class i. We train θ_1 on a large dataset of base classes to obtain a good representation. For the second and the third phase we initialize θ_{phase} with the parameters $\theta_{phase-1}$ and fine-tune on the corresponding to phase set.

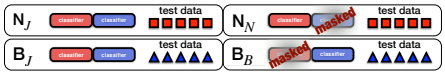

Figure 3: Classification layers for different evaluation protocols.

3.1. Second Phase - Novel Class Training

Base-Normalized Cross Entropy (CE_{BN}) Recently, Tian *et al.* [44] showed that in few-shot learning competitive classification on novel classes can be achieved given good representations using standard cross entropy without meta-learning and prototypes. We follow this idea and train θ_2 using a pretrained model θ_1 from the first phase and fine-tune it with a new classification layer for novel classes. The standard way to fine-tune the model on the training set that includes C_N classes is

$$CE(x) = \sum_{C_N} y_i ln\Big(\frac{exp(o_i)}{\sum_{C_N} exp(o_j)}\Big), \qquad (1)$$

where o_i is the logit of the corresponding class $i \in C_N$, and y_i equals to one if x belongs to class i, otherwise to zero.

One problem here is that even if the model is capable of learning information about the new classes C_N well, there is no guarantee that this performance is replicated in the joint space $N_{/J}$. By training two disjoint classifiers we learn classification weights that satisfy the classification problem either on novel classes $N_{/N}$ or on base classes $B_{/B}$, but so far the model does not learn any correlation between base classification weights and novel classification weights.

To this end, we propose to provide the model with information about the base class distribution in the joint space

using readily available information. Specifically, for each novel training sample we compute logits not only for the novel C_N classes, but also for the base C_B classes (note, in the second phase the base classes classifier is kept fixed). We use these logits to compute classification scores with the softmax function, thus the normalization of each score includes the base class logits that initially prevail in the sum, as follows:

$$CE_{BN}(x) = \sum_{C_N} y_i ln\Big(\frac{exp(o_i)}{\sum_{C_N} exp(o_j) + \sum_{C_B} exp(o_k)}\Big). \quad (2)$$

With this normalization, the novel model learns output probabilities for the novel classes directly in the joint space, and specifically increasing magnitude of novel class logits with respect to base class logits. This allows to have a good classification accuracy for novel classes $N_{/N}$ in the second phase and at the same time helps to match this accuracy in the joint space $N_{/J}$. Note that we do not use any base class training samples during this learning phase and we keep the weights of the base classifier fix.

Knowledge Preservation After the first phase the model performs well for the base classes and we aim to keep this capability. In FSL [8, 33] and IL [25, 35] multiple works show that adaptive representations can be beneficial for learning novel classes, specifically to fine-tune the parameters of the representation. In IL, the typical way to preserve knowledge from base classes [5, 24, 48, 9, 25, 35] is knowledge distillation (KD) [17] that is applied in the form of KL-divergence between logits of base classes from adapted and old models.

As an alternative to keep the network to remember about the previous knowledge, we propose to utilise explicit weight constraints (WC) of the model with respect to the old model from the first phase. We formulate it in form of a L_2 penalization over adaptive parameters of the representation [25, 21, 10]:

$$L_2^{WC} = \sum \|\theta_1 - \theta_2\|^2, \quad (3)$$

where θ_2 denotes adaptive parameters of the backbone excluding classification parameters during the second phase and θ_1 are the parameters of the model after base pretraining. The above constraint forces the model to keep the representation learned on base samples, but still allows the model to adjust the weights of the representation to better fit novel classes while not diverging a lot from the old model. The overall loss for the second phase is thus:

$$L = CE_{BN} + \lambda L_2^{WC}, \quad (4)$$

where λ controls the strength of knowledge preservation.

3.2. Third Phase - Joint Calibration

Balanced Replay The first and the second phase account for the performance on the base classes ($B_{/B}$) and for the novel class learning in both spaces ($N_{/N}$ and $N_{/J}$). For the third phase, due to the difference of number in training samples for base and novel classes and preservation of $B_{/B}$ during the second phase, the model is able to obtain good performance in the joint space as well. Empirically we found that during the second phase $B_{/J}$ performance can drop drastically, but due to keeping the base class performance $B_{/B}$ we can achieve good $B_{/J}$ performance in the third phase.

To achieve a balanced performance in the joint space of base and novel classes we apply the replay technique that is common in incremental learning [35, 26, 34, 24]. Specifically, we randomly draw only once base training samples, one per class, and join these samples with the novel training data. Moreover, in our case we require the least possible memory [35] to store exemplars of base classes, an essential component for incremental learning.

We continue training the model on the balanced dataset in the joint space. Due to the initial strong bias towards base classes from the first phase ($B_{/B}$), the model can improve its performance for base classes in the joint space ($B_{/J}$) quickly while at the same time overwriting the novel class performance at least partially ($N_{/J}$).

3.3. From Generalized to Incremental Learning

The main difference between GFSL [36, 39] and IFSL [43, 6] is the number of few-shot tasks. So far we considered the case of GFSL and it can be regarded as the first two tasks in terms of incremental setting: the training of the base classes refers to task one, the training of novel classes to task two in the incremental setup. As the current framework addresses the joint generalized problem in three phases, base classes, novel classes, and joint classes, we can easily extend the architecture to more tasks by repeating the novel class training. Specifically, for each new few-shot task we apply the second phase to learn a good joint classification for the current classes.

To evaluate the performance of the current joint space we finalize the training with the last phase of recuperation. To this end, we keep exemplars from base and novel classes from different tasks and perform training with the base-normalized cross-entropy loss and L_2^{WC} weights constraints as before. So, each time when we need to evaluate the joint performance on all classes, we apply the third phase.

For example, to report accuracy after five tasks, we learn representation parameters from base classes in the first task, as next stage we then apply the second phase sequentially for the second, third, fourth, and fifth tasks respectively, each time enlarging the classifier by the number of new classes in the task. After the fifth task we apply the third phase, balanced replay, where for each few-shot task we use all available data, and one sample per class for the data from the first task. During test the performance of the model is evaluated on a set that contains all previously seen classes.

method	mini-ImageNet 5w1s						mini-ImageNet 5w5s					
	$N_{/N}$ ($5_{/5}$)	$B_{/B}$ ($64_{/64}$)	$N_{/J}$ ($5_{/69}$)	$B_{/J}$ ($64_{/69}$)	$hm_{/J}$	$am_{/J}$	$N_{/N}$ ($5_{/5}$)	$B_{/B}$ ($64_{/64}$)	$N_{/J}$ ($5_{/69}$)	$B_{/J}$ ($64_{/69}$)	$hm_{/J}$	$am_{/J}$
CONV4												
PN [40]°	53.88	54.02	0.02	54.02	0.04	27.02	70.84	60.42	2.99	60.41	5.70	31.70
DFSL [13]°	55.80	69.93	40.30	58.54	47.74	49.42	72.24	70.24	58.26	59.89	59.06	59.07
RGFSL [39]	55.08	65.14	39.86	54.65	46.10	47.25	72.32	67.79	56.32	59.30	57.71	57.81
LCwoF (ours) *lim*	58.32	72.75	47.16	55.07	**50.81**	**51.12**	73.63	71.82	62.23	59.94	**61.06**	**61.09**
ResNet												
PN [40]⋆	-	-	-	-	-	42.73	-	-	-	-	-	57.05
IW [32](i)	47.17	61.78	31.25	47.72	37.77	39.49	67.56	69.07	46.96	58.92	52.26	52.94
DFSL [13] (c)	56.83	70.15	41.32	58.04	48.27	49.68	72.82	70.03	59.27	58.68	58.97	58.98
AAN [36](c)	56.14	77.58	45.61	63.92	53.24	54.76	69.72	77.58	60.82	64.14	62.43	62.48
AAN [36](orig)	-	-	-	-	-	54.95	-	-	-	-	-	63.04
LCwoF (ours) *lim*	60.78	79.89	53.78	62.89	**57.39**	**57.84**	77.65	79.96	68.58	64.53	**66.49**	**66.55**
LCwoF (ours) *unlim*	61.15	80.10	53.33	62.99	**57.75**	**58.16**	77.88	80.09	67.17	66.59	**66.88**	**66.88**

Table 1: Comparison to state-of-the-art on mini-ImageNet 5w1s (left) and 5w5s (right) with backbones CONV4 and ResNet. *lim* denotes limited access to base train samples during the third phase, for *unlim* we do not apply such restrictions. ° indicates results copied from RGFSL [39], ⋆ indicates results from AAN [36], (c) denotes that we run available code on the corresponding data, (i) states for our re-implementation of the respective method, (orig) indicates original numbers from the respective paper.

method	tiered-ImageNet 5w1s			tiered-ImageNet 5w5s		
	$N_{/J}$	$B_{/J}$	$hm_{/J}$	$N_{/J}$	$B_{/J}$	$hm_{/J}$
IW [32](i)	44.95	62.53	52.30	71.85	56.11	63.01
DFSL [13](c)	47.32	36.10	40.96	67.94	39.08	49.61
AAN [36](c)	54.39	55.85	55.11	57.76	64.13	64.93
LCwoF *lim*	57.13	60.39	**58.71**	69.05	63.44	**66.12**
LCwoF *unlim*	59.79	60.86	**58.75**	70.20	63.01	**66.41**

Table 2: Comparison to state-of-the-art on tiered-ImageNet 5w1s (left) and 5w5s (right) with ResNet backbone. $N_{/J}$ equal to $5_{/205}$, $B_{/J}$ to $200_{/205}$. *lim*, *unlim*, (i) and (c) see in Table 1.

4. Experimental Results

This section validates our proposed LCwoF-framework. First, we compare our method to the previous state-of-the-art work on both GFSL and IFSL in Section 4.1. Then we analyze each phase and the components separately to show the importance and connections of each to the improved performance in Section 4.2.

Datasets: *mini-ImageNet* [46] is a 100-class subset of ImageNet [7]. For FSL we follow [36] and use a subsets 64-12-24 classes that corresponds to base-val-novel classes, and for IFSL we follow [6, 43] with a subset of 60 and 40 classes for base training and incremental few-shot testing with 5 classes per each novel set. *tiered-ImageNet* [37] is a larger subset of Imagenet [7] with categorical splits for for base, validation, and novel classes. Here, each high-level category (e.g. dog that includes different breeds) belongs only to one of the splits. *mini-Kinetics* [49] is a 100-class subset of the Kinetics video classification dataset [20]. We use the splits from [49]. *UCF101* [41] is a video dataset with 101 classes in total. We follow [23] with a splitting of 50-51 for base and novel classes for FSL on videos. We additionally introduce and evaluate more challenging division of the dataset.

Implementation details For the FSL experiments on mini-ImageNet and tiered-ImageNet we employ the same ResNet12 with DropBlock [12] as backbone and pretrain it on base classes for 500 epochs with SGD optimizer with momentum with the learning rate (lr) of 1e-3 that is decayed by 0.1 at 75, 150, and 300 epochs. For the second and the third phase we use lr of 1e-2 and 1e-3 respectively for the classification layers and decayed by 0.1 lr for the backbone parameters. For IFSL experiments we use ResNet18 and follow the same pretraining steps as above. We use different architecture choices for GFSL and IFSL to remain comparable to previous works after the base pretraining. For the second phase we always train the model for 150 epochs, while for the third phase we use validation set to choose the number of epochs for each dataset. For videos we preextract features with C3D model [45] pretrained on large-scale Sports-1M [19] dataset. We apply average pooling over temporal domain to obtain one feature vector per video. As a backbone we use 2-layer MLP. We also clip gradients at value 100 for the experiments. More details can be found in the supplement.

Evaluation We evaluate the proposed framework primarily with respect to the harmonic mean ($hm_{/J}$) [38, 39, 23] that is computed between base and novel performance in the joint space. Additionally, we report performance of base and novel classes in their respective subspaces ($B_{/B}$ and $N_{/N}$), in the joint space ($B_{/J}$ and $N_{/J}$), and the arithmetic mean over the joint space ($am_{/J}$) as in [36]. Extended tables for all datasets are in the supplement. 5w1s and 5w5s denote 5 novel classes with 1 and 5 training samples per classes respectively. For the state-of-the-art comparison, we average over 600 episodes [39, 27, 8], for all other experiments over 100 episodes. *unlim* denotes access to the entire base training set, whereas for *lim* setup we use small subset. All

specifications are in supplement.

4.1. Comparison to state-of-the-art

Generalized Few-Shot Learning We compare our performance on image and video benchmarks: mini-ImageNet, tiered-ImageNet, Kinetics and UCF101 in Tables 1, 2, 3, and 4 respectively. For mini-ImageNet and tiered-ImageNet, we train the respective backbones from scratch on the base classes. For Kinetics and UCF101, we preextract video features as in [23, 49] and then train a shallow MLP model on the base classes.

On mini-ImageNet in Table 1 we provide an evaluation in separate and joint space on 5w1s and 5w5s setups. For both backbones, conv4 [39] and ResNet, we achieve significant improvements over state-of-the-art results in terms of $hm_{/J}$. Here, we can observe that previous methods drop in performance on both novel ($N_{/N}$) and base ($B_{/B}$) classes, whereas we address the problem by explicitly balancing between forgetting, learning, and calibration and achieve better performance.

On tiered-ImageNet, Table 2, we can observe a similar pattern and achieve strong improvements. Here, even with more base classes, we are able to calibrate the performance between novel and base classes.

We further evaluate the performance of the proposed idea on two video datasets. Our results on Kinetics, shown in Table 3, and UCF101, shown in Table 4, show that the proposed framework is able to perform well on the pre-extracted features. Results on UCF101 we present on two different splits for training and testing. In Table 4 the first two lines correspond to splits provided by [23] thus can be directly compared. The second part of the table shows the evaluation for the setup with the original UCF train/test split as defined in [41].

Note that on both image datasets we obtain significant gains in performance while applying *unlim* sampling strategy, while on the Kinetics and UCF101 there is a slight decrease in comparison to *lim*. We speculate that it happens due to the fixed feature preextraction whereas on images we train models on raw images. Across all the datasets, setting and architectures we consistently achieve significantly better performance than previous work.

Incremental Few-Shot Learning We compare our method with the current few-shot class incremental methods in Table 5. As in the previous experiments we use the hm accuracy that we compute between base (first set) and novel classes. We provide more detail on the performance of each task in the supplement, specifically showing performance of base and novel classes separately, as well as the standard accuracy of all classes in the joint space. Our method notably outperforms other methods in the field due to the fact that we address directly the balance between the performance on base and novel classes. We show that we obtain higher novel accuracy in the joint space for every task.

	mini-Kinetics 5w1s			mini-Kinetics 5w5s		
method	$N_{/J}$	$B_{/J}$	$hm_{/J}$	$N_{/J}$	$B_{/J}$	$hm_{/J}$
IW [32](i)	45.56	48.56	47.01	56.92	49.17	52.76
DFSL [13](c)	50.81	44.51	47.45	70.29	46.31	55.83
GFSV [49]	13.70	88.70	23.73	22.30	88.70	35.64
ANN [36](c)	46.13	35.96	40.41	56.99	43.21	49.15
LCwoF *lim*	47.51	50.84	**49.12**	63.65	54.55	**58.75**
LCwoF *unlim*	46.26	51.94	**48.93**	65.40	52.70	**58.37**

Table 3: Comparison to state-of-the-art on mini-Kinetics 5w1s (left) and 5w5s (right) with MLP backbone on pre-extracted features. $N_{/J}$ equal to $5_{/69}$, $B_{/J}$ to $64_{/69}$. *lim*, *unlim*, (i) and (c) see in Table 1.

	UCF101 50w1s		
method	$N_{/J}$	$B_{/J}$	$hm_{/J}$
ProtoG [23]	52.30	75.30	61.72
LCwoF (ours) *lim*	54.41	91.41	**68.22**
IW [32](i)	45.22	76.15	56.74
LCwoF (ours) *lim*	50.78	70.72	**59.11**
LCwoF (ours) *unlim*	49.12	69.98	**57.72**

Table 4: Comparison to state-of-the-art on UCF101 50w1s with MLP backbone on pre-extracted features. $N_{/J}$ equal to $50_{/101}$, $B_{/J}$ to $51_{/101}$. *lim*, *unlim* and (i) see in Table 1.

4.2. Ablation Studies

Here we investigate the influence of several components of our framework and the impact of various hyperparameters.

Calibration and forgetting during the second phase In this subsection we analyse the influence of the base-normalized cross entropy as a technique to address the calibration problem as well as the influence of knowledge preservation to address the forgetting issue. In Fig. 4 we show the behaviour of the training process during the second and the third phase with and without base-normalized cross entropy and knowledge preservation as explicit weight constraints L_2^{WC}. Removing both elements, as shown in the top left sub-figure, results in a drastic drop in $B_{/B}$ and $B_{/J}$ that prohibits the model to quickly recover during the replay (third) phase since all previous knowledge is lost. The left bottom sub-figure shows the impact of CE_{BN}. With CE_{BN} the model easily achieves good performance on $N_{/J}$ in the joint space that matches $N_{/N}$, but here both $B_{/B}$ and $B_{/J}$ drop during the second phase training that again prevents recuperation. Compared to that, the right top sub-figure includes base knowledge preservation that keeps both $B_{/B}$ and $B_{/J}$ relatively high, and further facilitates complete recovery for the base classes. But without CE_{BN} novel learning in the joint space suffers during both the second and the third phase. The right bottom sub-figure shows performance if both, CE_{BN} and knowledge preservation, are used. The model is able to keep a certain performance of $B_{/B}$ during the second phase while achieving high accuracy on the novel classes ($N_{/N}$ and $N_{/J}$). During the third phase we calibrate the model by the replay technique and achieve good balance

mini-ImageNet 5w5s								
method	2	3	4	5	6	7	8	9
$hm_{/J}$	+5	+10	+15	+20	+25	+30	+35	+40
FT$^{\diamond}$	7.23	7.39	4.87	2.40	2.06	1.84	1.57	1.40
Joint$^{\diamond}$	8.92	17.02	21.86	20.54	22.92	22.85	24.41	24.95
iCaRL [35]$^{\diamond}$	8.45	13.86	14.92	13.00	14.06	12.74	12.16	11.71
UCIR [18]$^{\diamond}$	9.62	14.14	15.58	13.19	13.63	13.11	12.76	11.96
PN [40]$^{\diamond}$	9.76	14.72	16.78	19.09	20.06	19.37	18.98	18.90
ILVQ [50]$^{\diamond}$	9.66	16.08	17.78	20.05	20.35	19.64	19.06	18.89
SDC [52]$^{\diamond}$	20.51	18.79	17.36	20.47	19.21	18.27	20.79	21.77
IW [32]$^{\diamond}$	25.32	20.45	22.62	25.48	22.54	20.66	21.27	22.27
IDLVQ [6]	21.69	20.44	21.98	25.19	22.99	20.82	21.56	22.65
LCwoF *lim*	**41.24**	**38.96**	**39.08**	**38.67**	**36.75**	**35.47**	**34.71**	**35.02**

mini-ImageNet 5w5s				
$B_{/J}$ (60)	2	5	7	9
Joint$^{\diamond}$	63.30	62.18	61.86	61.89
SDC [52]$^{\diamond}$	63.58	60.29	59.05	59.87
IW [32]$^{\diamond}$	63.52	61.17	60.63	59.64
IDLVQ [6]	63.77	61.22	60.97	60.44
LCwoF *lim*	57.33	51.38	47.60	47.73
$N_{/J}$	2	5	7	9
$\#cl$	5	20	30	40
Joint$^{\diamond}$	4.80	12.30	14.01	15.62
SDC [52]$^{\diamond}$	12.23	12.33	10.81	13.30
IW [32]$^{\diamond}$	15.81	16.09	12.45	13.69
IDLVQ [6]	13.07	15.86	12.55	13.94
LCwoF *lim*	**32.20**	**31.12**	**28.27**	**27.65**

Table 5: IFSL. Comparison to state-of-the-art on mini-ImageNet. Number of base classes for the first task is 60. Each next task increases joint space by 5 novel classes with 5 training samples per class. $^{\diamond}$ indicates results copied from IDLVQ [6]. Left: $hm_{/J}$ between base and novel classes in the joint space for each task; right top: base classification accuracy in the joint space; right bottom: novel classification accuracy in the joint space.

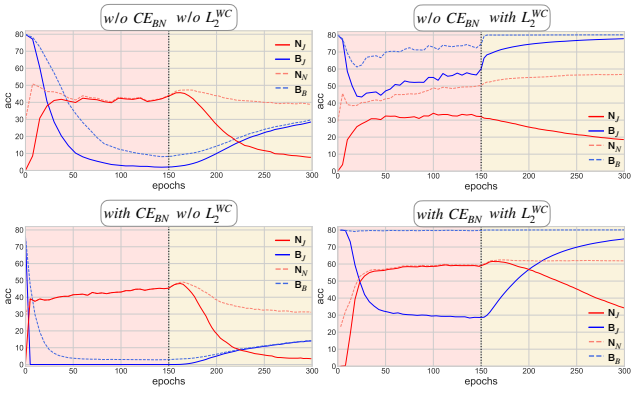

Figure 4: Influence of the proposed CE_{BN} and L_2^{WC} on the training during the 2^{nd} and the 3^{rd} phase. Red lines correspond to novel classes, blue to base classes. Solid lines represent performance in the joint space, dashed lines in separate corresponding subspaces. *Top left*: base performance drops as $B_{/B}$ and $B_{/J}$ and is not able to recover during the 3^{rd} phase; *bottom left*: both $N_{/N}$ and $N_{/J}$ achieve high performance, base performance still is not able to recover; *top right*: L_2^{WC} helps to recuperate base performance in the joint space while novel performance drops; *bottom right*: by addressing both $N_{/J}$ with CE_{BN} and $B_{/B}$ with L_2^{WC} we allow the model recover during the 3^{rd} phase on $B_{/J}$ while not losing drastically on novel classes.

between the two disjoint sets of classes.

Knowledge Preservation One important factor of the method is that we aim to retain the knowledge that the model obtained in the first phase, specifically the $B_{/B}$ performance, during the second phase. In this section we evaluate two different methods to achieve this objective, comparing the proposed explicit weight constraints L_2^{WC} with knowledge distillation that is formulated via KL-divergence. Knowledge distillation is a common technique to preserve knowledge in incremental learning [5, 9, 24, 25, 35, 48], where abundant training data is available for new classes.

In Table 6 we evaluate the performance on 100 episodes

method	5w1s $N_{/J}$ ($5/_{69}$)	5w1s $B_{/J}$ ($64/_{69}$)	5w1s $hm_{/J}$	5w5s $N_{/J}$ ($5/_{69}$)	5w5s $B_{/J}$ ($64/_{69}$)	5w5s $hm_{/J}$
L_2^{WC}	53.28	63.24	**57.83**	68.61	64.73	66.61
KD	50.28	57.72	53.74	70.29	64.67	**67.36**
KD^+	45.87	67.20	54.52	68.01	65.88	66.93
$L_2^{WC} + KD$	53.43	60.51	56.75	68.45	63.47	65.86

Table 6: Comparison of knowledge preservation techniques, such as L_2^{WC} as explicit weights constraints, knowledge distillation (KD) for old classes during the 2^{nd} phase, KD^+ includes KD on the additional 1000 unlabeled images during the second phase, and combination $L_2^{WC} + KD$. Results are computers on mini-ImageNet.

on mini-ImageNet for the 5w1s and 5w5s settings. Comparing L_2^{WC} and KD knowledge preservation techniques, we find that the latter marginally outperforms the other if more data is available, as in the 5w5s setting, whereas plain weight constraints are more efficient in the lowest data regime with 1 training sample per class (5w1s). Additionally, we evaluate KD by including additional unlabeled 1000 images from the validation set during the second phase for KD loss computations, denoted as KD^+ in Table 6. We find that it helps to improve in the 5w1s setting, but still stays lower than L_2^{WC}. Applying both techniques at the same time does not give an improvement.

Impact of λ In this section we study the influence of λ for the L_2^{WC} loss and how to preserve more knowledge and drop less on the base classes $B_{/J}$, but also how the model behaviour changes if we enforce it to preserve even more. In Fig. 5 we plot the accuracy after training in the second phase, before the replay phase for different λ. In Fig. 5 we fill these areas with gray color, that allow to reach our two objectives for the second phase, i.e., achieving good performance in the joint space on novel classes $N_{/J}$ and keeping good accuracy in the base space $B_{/B}$. Specifically higher λ helps to further keep $B_{/B}$ performance while novel learning $N_{/N}$ and $N_{/J}$

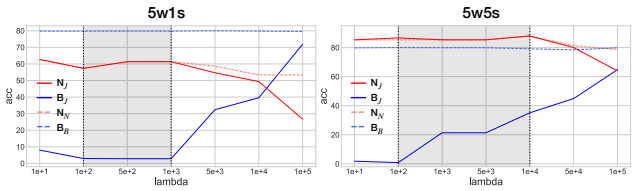

Figure 5: Evaluation of the performance after the 2^{nd} phase of the framework on the base and novel classes that depends on importance weight λ of the L_2^{WC} term. Filled areas correspond appropriate λ that allows us to achieve the desirable performance (before $N_{/J}$ drops). Results on mini-ImageNet.

	5w1s		
λ	$N_{/J}$	$B_{/J}$	$hm_{/J}$
1e+4	50.89	50.68	50.78
5e+4	53.06	58.56	55.67
1e+5	52.65	58.70	55.51
LCwoF	53.28	63.24	**57.83**

Table 7: Best possible performance of the model that can be achieved during the second phase (without the third phase). λ stands for the importance weight of the L_2^{WC} term. Higher λ, higher knowledge preservation, higher accuracy $B_{/J}$. We apply GCE and L_2^{WC} during the second phase with various λ. Results on mini-ImageNet.

	$N_{/J}$	$B_{/J}$	$hm_{/J}$
no	54.39	59.13	56.66
yes	53.28	63.24	**57.83**

Table 8: Influence of freezing batch norm during the 2^{nd} and the 3^{rd} phases, results on mini-ImageNet 5w1s.

starts decreasing with higher values. It shows the start of the decrease depends on the number of available training samples: the more training data, the more we can keep from base classes by choosing a higher λ.

Early Stopping: Second Phase We observe that during the second phase, usually $B_{/J}$ starts dropping even when $N_{/J}$ has already achieved some reasonable accuracy, as can e.g. be seen in the bottom right subplot of Figure 4. We, therefore, also report the best performance that can be achieved with different λ during the second phase in Table 7. Note that λ influences the contribution of the knowledge preservation part L_2^{WC}. Thus, $B_{/J}$ will drop faster if a lower λ is chosen. By adjusting λ, we find that the proposed technique can also be helpful during the second phase. It shows that in this case, balanced model performance can be reached with higher λ and that we can achieve high performance even without the third replay phase.

Impact of batch normalization We use batch norm layers in the model that capture statistic from the base classes during the first phase. At the second phase our data is highly limited thus we fix batch norm during further training. Table 8 shows that the performance drops more than 1 point when the model tries to accumulate new statistic from 1 training sample per class and to adapt parameters respectively.

		$N_{/J}$	$B_{/J}$	$hm_{/J}$
1	cos.norm.	47.91	59.69	53.15
2	bias	52.96	62.05	57.14
3	no bias	53.28	63.24	**57.83**

Table 9: Comparison between different linear layers: cos is cosine normalization of the weights and the embedding space, bias stands for linear layer with the bias term, no bias stands for linear layer without the bias term. Results on mini-ImageNet 5w1s.

Impact of normalization One of the common strategies to unify magnitudes of base and novel classifiers is to use cosine normalization of the embedding and weights [18, 36, 39, 30, 14, 35]. We experiment with such a setup (Table 9, lines 1 & 3) for our framework and find a decline in performance for both, $N_{/J}$ and $B_{/J}$. Note that the performance of the model after the first phase on $B_{/B}$ is the same as without cosine normalization. But if we attempt to match $N_{/J}$ and $N_{/N}$ during the second phase, we find that constrained magnitudes of logits due to normalization restrict the performance and do not allow to achieve our second phase objectives.

Analysis of classification layer As default, we conduct all our experiments with a linear classification layer without bias term (Table 9, lines 2 & 3). We therefore assess the performance of the model with and without bias. We find that during the first phase, it is the same, but that during the second and the third phase it is beneficial to use the latter giving a boost of about 0.7 in the performance $hm_{/J}$.

5. Conclusion

This paper addresses major challenges in generalized few-shot and incremental few-shot learning with our three-phase framework. First, we learn a powerful representation by training a model on base classes. In the second phase, concerned with novel class learning, we employ base-normalized cross entropy that calibrates novel class classifiers to overcome the bias towards base classes. Additionally, during that phase we preserve knowledge about base classes via weight constraints. In the third phase, to achieve calibrated classifiers across both base and novel classes, we employ balanced replay. We show that each phase of the framework allows to study and address the essential problems of the task explicitly. We evaluate our proposed framework on four benchmark image and video datasets and achieve state-of-the-art performance across all settings. This work can be seen as a first step towards more explicitly addressing calibration, learning and knowledge preservation jointly to further improve deep learning for imbalanced settings beyond the ones addressed in this paper.

References

[1] Hongjoon Ahn, Jihwan Kwak, Subin Lim, Hyeonsu Bang, Hyojun Kim, and Taesup Moon. Ss-il: Separated softmax for incremental learning. *arXiv:2003.13947*, 2020. 2

[2] Ali Ayub and Alan R Wagner. Cognitively-inspired model for incremental learning using a few examples. In *CVPRW*, 2020. 2

[3] Eden Belouadah and Adrian Popescu. Il2m: Class incremental learning with dual memory. In *ICCV*, 2019. 2, 3

[4] Eden Belouadah and Adrian Popescu. ScaIL: Classifier weights scaling for class incremental learning. In *WACV*, 2020. 2

[5] Francisco M Castro, Manuel J Marín-Jiménez, Nicolás Guil, Cordelia Schmid, and Karteek Alahari. End-to-end incremental learning. In *ECCV*, 2018. 2, 4, 7

[6] Kuilin Chen and Chi-Guhn Lee. Incremental few-shot learning via vector quantization in deep embedded space. In *ICLR*, 2021. 1, 2, 4, 5, 7

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 5

[8] Guneet Singh Dhillon, Pratik Chaudhari, Avinash Ravichandran, and Stefano Soatto. A baseline for few-shot image classification. In *ICLR*, 2020. 4, 5

[9] Arthur Douillard, Matthieu Cord, Charles Ollion, Thomas Robert, and Eduardo Valle. PODNet: Pooled outputs distillation for small-tasks incremental learning. In *ECCV*, 2020. 4, 7

[10] Theodoros Evgeniou and Massimiliano Pontil. Regularized multi–task learning. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2004. 4

[11] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, 2017. 2

[12] Golnaz Ghiasi, Tsung-Yi Lin, and Quoc V Le. Dropblock: A regularization method for convolutional networks. 2018. 5

[13] Spyros Gidaris and Nikos Komodakis. Dynamic few-shot visual learning without forgetting. In *CVPR*, 2018. 1, 2, 5, 6

[14] Spyros Gidaris and Nikos Komodakis. Generating classification weights with gnn denoising autoencoders for few-shot learning. In *CVPR*, 2019. 2, 8

[15] Bharath Hariharan and Ross Girshick. Low-shot visual recognition by shrinking and hallucinating features. In *ICCV*, 2017. 2

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 2

[17] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 4

[18] Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. Learning a unified classifier incrementally via rebalancing. In *CVPR*, 2019. 2, 3, 7, 8

[19] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *CVPR*, 2014. 5

[20] Will Kay, João Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, Mustafa Suleyman, and Andrew Zisserman. The kinetics human action video dataset. *CoRR*, 2017. 5

[21] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13), 2017. 4

[22] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *NIPS*, 2012. 2

[23] Sai Kumar Dwivedi, Vikram Gupta, Rahul Mitra, Shuaib Ahmed, and Arjun Jain. Protogan: Towards few shot learning for action recognition. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2019. 1, 2, 5, 6

[24] Kibok Lee, Kimin Lee, Jinwoo Shin, and Honglak Lee. Overcoming catastrophic forgetting with unlabeled data in the wild. In *ICCV*, 2019. 4, 7

[25] Zhizhong Li and Derek Hoiem. Learning without forgetting. *TPAMI*, 40, 2017. 2, 4, 7

[26] Yaoyao Liu, Yuting Su, An-An Liu, Bernt Schiele, and Qianru Sun. Mnemonics training: Multi-class incremental learning without forgetting. In *CVPR*, 2020. 2, 4

[27] Tiange Luo, Aoxue Li, Tao Xiang, Weiran Huang, and Liwei Wang. Few-shot learning with global class representations. In *ICCV*, 2019. 2, 5

[28] Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24. Elsevier, 1989. 2

[29] Nikhil Mishra, Mostafa Rohaninejad, Xi Chen, and Pieter Abbeel. A simple neural attentive meta-learner. In *ICLR*, 2018. 2

[30] Sudhanshu Mittal, Silvio Galesso, and Thomas Brox. Essentials for class incremental learning. *arXiv:1904.07734*, 2021. 2, 8

[31] Ameya Prabhu, Philip HS Torr, and Puneet K Dokania. GDumb: A simple approach that questions our progress in continual learning. In *ECCV*, 2020. 2

[32] Hang Qi, Matthew Brown, and David G Lowe. Low-shot learning with imprinted weights. In *CVPR*, 2018. 1, 2, 5, 6, 7

[33] Aniruddh Raghu, Maithra Raghu, Samy Bengio, and Oriol Vinyals. Rapid learning or feature reuse? towards understanding the effectiveness of maml. *arXiv preprint arXiv:1909.09157*, 2019. 4

[34] Jathushan Rajasegaran, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Mubarak Shah. iTAML : An incremental task-agnostic meta-learning approach. *CVPR*, 2020. 4

[35] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. iCaRL: incremental classifier and representation learning. In *CVPR*, 2017. 2, 4, 7, 8

[36] Mengye Ren, Renjie Liao, Ethan Fetaya, and Richard Zemel. Incremental few-shot learning with attention attractor networks. In *NIPS*, 2019. 1, 2, 4, 5, 6, 8

[37] Mengye Ren, Eleni Triantafillou, Sachin Ravi, Jake Snell, Kevin Swersky, Joshua B Tenenbaum, Hugo Larochelle, and Richard S Zemel. Meta-learning for semi-supervised few-shot classification. In *ICLR*, 2018. 5

[38] Edgar Schonfeld, Sayna Ebrahimi, Samarth Sinha, Trevor Darrell, and Zeynep Akata. Generalized zero-and few-shot learning via aligned variational autoencoders. In *CVPR*, 2019. 5

[39] Xiahan Shi, Leonard Salewski, Martin Schiegg, Zeynep Akata, and Max Welling. Relational generalized few-shot learning. 2020. 1, 2, 4, 5, 6, 8

[40] Jake Snell, Kevin Swersky, and Richard S Zemel. Prototypical networks for few-shot learning. In *NIPS*, 2017. 2, 5, 7

[41] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human action classes from videos in the wild. *CRCV-TR-12-01*, 2012. 5, 6

[42] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *CVPR*, 2018. 2

[43] Xiaoyu Tao, Xiaopeng Hong, Xinyuan Chang, Songlin Dong, Xing Wei, and Yihong Gong. Few-shot class-incremental learning. In *CVPR*, 2020. 1, 2, 4, 5

[44] Yonglong Tian, Yue Wang, Dilip Krishnan, Joshua B Tenenbaum, and Phillip Isola. Rethinking few-shot image classification: a good embedding is all you need? 2020. 2, 3

[45] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, 2015. 5

[46] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. In *NIPS*, 2016. 2, 5

[47] Yu-Xiong Wang, Ross Girshick, Martial Hebert, and Bharath Hariharan. Low-shot learning from imaginary data. In *CVPR*, 2018. 1, 2

[48] Yue Wu, Yinpeng Chen, Lijuan Wang, Yuancheng Ye, Zicheng Liu, Yandong Guo, and Yun Fu. Large scale incremental learning. In *CVPR*, 2019. 2, 3, 4, 7

[49] Yongqin Xian, Bruno Korbar, Matthijs Douze, Bernt Schiele, Zeynep Akata, and Lorenzo Torresani. Generalized many-way few-shot video classification. *arXiv preprint arXiv:2007.04755*, 2020. 2, 5, 6

[50] Ye Xu, Furao Shen, and Jinxi Zhao. An incremental learning vector quantization algorithm for pattern classification. *Neural Computing and Applications*, 21(6), 2012. 7

[51] Han-Jia Ye, Hexiang Hu, De-Chuan Zhan, and Fei Sha. Few-shot learning via embedding adaptation with set-to-set functions. In *CVPR*, 2020. 2

[52] Lu Yu, Bartlomiej Twardowski, Xialei Liu, Luis Herranz, Kai Wang, Yongmei Cheng, Shangling Jui, and Joost van de Weijer. Semantic drift compensation for class-incremental learning. In *CVPR*, 2020. 7

[53] Richard S Zemel and Michael C Mozer. Localist attractor networks. *Neural Computation*, 13(5), 2001. 2

[54] Bo Zhao, Shixiang Tang, Dapeng Chen, Hakan Bilen, and Rui Zhao. Continual representation learning for biometric identification. In *WACV*, 2021. 2