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

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

In the supplement in Section A we provide additional ablation experiments on the second and the third phase, further in Section B we expand implementation details by specifying the applied data augmentation, generalized and incremental few-shot learning setups, and the splits used for UCF101 dataset. Finally, Section C contains extended tables for all the datasets.

A. Additional ablation of the phases

In this section we discuss possible variations of our proposed framework and their influence on the performance. Specifically, we discuss the necessity of the second phase and the duration of the second phase. We also inspect the influence of knowledge preservation on the performance after the third phase and the impact of weight decay regularization.

A.1. The second phase

Skip the second phase In the proposed work, we address the problem of generalized few-shot learning with a three-phase framework. During the second phase we target to improve *novel class learning* and to mitigate *catastrophic forgetting* of the base classes. In Fig. 1 we show the development of the performance when we skip the second phase and directly proceed with the third phase. During the third, joint calibration phase, the training set consists of base (one sample per class) and all novel training samples. The performance of the base classes in the joint space B_J and the separate space B_B stays at high level even with few training samples. While the separate novel N_N performance can reach high values during the third phase, novel class learning in the joint space suffers from strong bias towards base classes (red curve on the figure stays low). It shows that our second phase for explicit novel learning in the joint space gives a significant boost to the overall performance in the joint space.

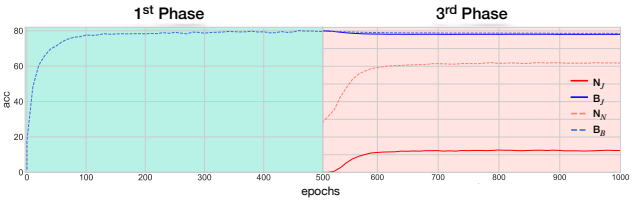

Figure 1: We skip the second phase in our three-phase framework and sequentially apply the first and the third phase instead. The red curve (N_J) shows that the performance of the model on novel samples in the joint space stays low and is not able to achieve high accuracy in the joint space without the second phase.

Skip the second phase and keep batch ratio during the third phase We further evaluate the performance of the model without the second phase but with the third phase adaptation. Specifically, we ensure consistent batch-wise ratio between novel and base classes during the third phase. In Table 1 the results show not only better performance than trivially skipping the second phase but also outperform the previous state-of-the-art [11]. Our proposed three-phase framework performs better on the novel classes.

batch size $\#N + \#B$	per batch ratio N/B	$N_{/J}$ ($5/69$)	$B_{/J}$ ($64/69$)	$hm_{/J}$
$5+1$	$83/17$	52.39	57.28	54.72
$5+2$	$71/29$	51.51	59.59	55.26
$5+3$	$63/37$	48.11	63.52	54.75
ANN [11]	-	45.61	63.92	53.24
LCwoF	3 phases	53.28	63.24	57.83

Table 1: 5w1s mini-ImageNet. No 3^{rd} phase, controlled batch ratio.

Interleave the second and the third phases In order to shed light onto the question if one should separate the sec-

ond and the third phases as proposed, in Table 2, we instead interleave the second and the third phases. Particularly, we alternate training on novel classes only (for X epochs) and balanced replay (for 1 epoch). We use $X = 10, 20, 30$. Phase alternation shows to be an effective alternative compared to the consecutive execution that still performs best.

epochs per period $X(novel) + 1(replay)$	period ratio N/B	$N_{/J}$ $(5/_{69})$	$B_{/J}$ $(64/_{69})$	$hm_{/J}$
$10+1$	$46/54$	43.08	70.06	53.35
$20+1$	$62/38$	49.20	65.40	56.15
$30+1$	$71/29$	51.37	64.20	57.07
LCwoF	3 phases	53.28	63.24	57.83

Table 2: 5w1s mini-ImageNet. Interleave of the 2^{nd} and the 3^{rd} phases.

A.2. Number of epochs of the second phase

For the evaluation, we train the model for a fixed number of epochs during the second phase for all the datasets and setups. Fig. 2 shows similar behaviour when we apply smaller (30) number of epochs during the second phase and compare it to a longer second phase (150 epochs). Due to the negligible differences we use the initially chosen value that equals 150 epochs throughout the main paper.

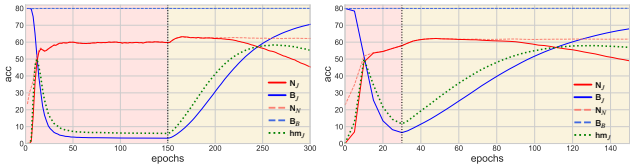

Figure 2: Different duration of the second phase. The behaviour and the quantitative performance is the same. *Left*: we use 150 epochs for the second phase; *right*: we use 30 epochs for the second phase.

A.3. High λ for the weight constraints

In the main paper in Fig.4 we show the range of appropriate λ to achieve good balanced performance for 5w1s and 5w5s setups. We claim that λ should not prevent novel class learning while preserving base performance in the base class space. In Fig.4 we exclude too low λ since empirically we found decrease in the performance on the base classes after the third phase. Table 3 presents the performance of the model with different λ after the third phase. Higher λ helps to better preserve knowledge of the base classes while it hinders novel class learning in the joint space.

A.4. Impact of weight decay regularization

While we apply constraints on the parameters of the model by applying L_2^{WC}, the question arises if and to which extend we would need standard weight decay as regularization on the model parameters. As shown in Fig. 3, while the contribution of the regularization term remains minor, it neither helps the performance nor harms.

λ	5w1s $N_{/J}$	$B_{/J}$	$hm_{/J}$
1e+1	53.27	60.03	56.45
1e+2	53.37	61.45	57.13
5e+2	53.28	63.24	57.83
5e+3	52.65	63.48	57.56

Table 3: Performance of the model after the third phase . λ stands for the importance weight of the L_2^{WC} term. Higher λ, higher knowledge preservation, higher accuracy $B_{/J}$. Performance of the model after the second phase for the corresponding λ can be found in Fig.4 in the main paper. Results on mini-ImageNet.

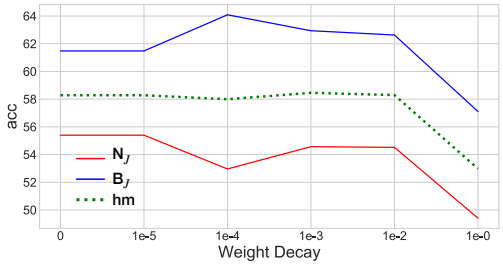

Figure 3: Contribution of weight decay regularization and influence on the performance of the framework in term of $N_{/J}$, $B_{/J}$ and $hm_{/J}$. Results on mini-ImageNet, 5w1s, averaged over 100 episodes.

A.5. 10w1s and 20w1s

We evaluate our approach for additional setups to directly compare to knowledge preservation methods, similarly as in Table 6 in the main paper. Table 4 confirms our finding that with little amount of data knowledge distillation (KD) [4, 10] performs worse than with L_2^{WC} constraints.

method	10w1s $N_{/J}$ $(10/_{74})$	$B_{/J}$ $(64/_{74})$	$hm_{/J}$	20w1s $N_{/J}$ $(20/_{84})$	$B_{/J}$ $(64/_{84})$	$hm_{/J}$
L_2^{WC}	40.84	58.31	47.75	29.76	56.68	38.87
KD	37.81	58.22	45.53	27.06	57.75	36.85

Table 4: Comparison of knowledge preservation techniques on mini-ImageNet for setups 10w1s and 20w1s, with 10 and 20 novel classes correspondingly.

B. Extended implementation details

This section covers additional details of the implementation. We first specify the exact augmentation for images and

then discuss the evaluation for the generalized and incremental setups. Our framework is built with PyTorch library and will be made publicly available.

Augmentation For training on images we apply standard augmentation with random resizing followed by random cropping to the size 84x84 and random horizontal flip. We also use color jittering that allows to randomly change brightness, contrast, and saturation. For the test we first resize an image to the size of 92x92 and then apply a central 84x84 crop.

Evaluation For each image and video dataset for testing we use 15 samples per class for both base and novel classes. We train two parametric classifiers for base and novel classes respectively, to evaluate the performance in the joint space we concatenate the logit vectors (dimensionalilty C_B and C_N for base and novel classes accordingly) and predict the class by applying argmax operator over the concatenated output vector of dimensionality $C_B + C_N$.

lim & ***unlim*** For each dataset we conduct experiments with the two following setups: *lim* denotes *limited* access to the base training data during the third phase, whereas for *unlim* we allow the model to have *unlimited* access to the base training samples during the third phase. During the third phase we target to have balanced training set, thus the replay set consists of all $|C_N| \cdot K$ novel samples and $|C_B| \cdot K$ base samples. K refers to the number of samples for each novel class, the notation corresponds the standard notation for few-shot learning [1, 3, 6], e.g. 5w1s denotes 5 novel classes with 1 training sample per class, thus $K = 1$. *lim*: for each episode we draw at random the replay set only **once** before the third phase and reuse that replay set for each epoch for that episode. *unlim*: for each episode we draw random base training samples for the replay set before each epoch anew.

B.1. Generalized few-shot learning setup

Episodic training is a common way to evaluate few-shot learning methods, further we detail the difference from the standard training protocol [3, 11, 18]. As before C_B stands for base classes. Generally, the few-shot setup is formulated in a N-way K-shot notation (C_N from the main paper equals to N in this formulation), specifically each few-shot episode consists of C_N novel classes with K training samples per class. In the generalized setup each episode includes C_B base classes for the classification along with C_N novel classes. Each dataset includes T classes in the novel test set, usually $|T| \gg |C_N|$, e.g. in mini-ImageNet there are $T = 24$ classes in the test set whereas we evaluate on 5w1s and 5w5s (for both cases $C_N = 5$). To this end, following standard practice [3, 11, 18] we evaluate the performance as average over 600 episodes such that for each episode we repeat:

1) randomly draw C_N novel classes from T
2) randomly draw K training samples per each class
3) apply training framework to the current training data
4) randomly draw 15 samples per class to test the framework
5) reset the framework to initial state and clean train data

B.2. Incremental few-shot learning setup

We refer to recent work [2, 8] for an extensive overview and taxonomy on incremental (continual) learning. In our work we aim at class-incremental learning where all seen classes should be classified in the joint space. Whilst another popular choice of continual learning is task-incremental learning with an objective to achieve high accuracy in the disjoint spaces, in our notation we could refer to this setup as having high $B_{/B}$ base performance and high $N_{/N}$ novel performance separately. Such formulation of the task is easier than to achieve a joint balanced, high performance. The usual way to evaluate class incremental learning [7, 15, 1] is to continuously measure performance of the model in the growing joint space. The first task (sometimes called session) is to evaluate performance on the base classes C_B. Following few-shot N-way (C_N) K-shot notation, the second task increases the joint space by $|C_N|$ classes, the third task increases again by $|C_N|$ classes resulting in $|C_B| + 2|C_N|$ classes and so on up to 9 tasks in mini-ImageNet, namely $|C_B| + 8|C_N|$ classes. We follow [15, 1] and use the same division into the tasks, training and test samples.

B.3. UCF101 splits

As mentioned in the main paper in Section 4, for UCF101 we have introduced a novel split. We observe that the $B_{/B}$ performance achieves almost 99 points with the previously introduced split by Dwivedi [6] that we report in Table 8. We do not change the division into base and novel classes but instead we filter out some videos that share the same group [14] from train and test splits. When comparing the results of $B_{/B}$ from the previous split and our novel split we indeed see a drop in the performance indicating that the proposed split corresponds to a harder task. Subsequently, the performance $N_{/N}$, $B_{/J}$ and $N_{/J}$ also drops. We will make the novel split publicly available.

C. Extended tables

In Table 5 we summarize Tables 9, 10 and 11 by reporting performance with different metrics after all 9 tasks for incremental few-shot learning. In the supplement we account *base biased* and *balanced hm* performance for our framework. *base biased* stands for the performance of the model that shows higher accuracy on base classes, whereas *balanced hm* indicates more balanced performance between the disjoint sets that we control by number of epochs for the

third phase. The discrepancy is caused by the difference in the number of base and novel classes ($|C_B| \gg |C_N|$) and the initial bias of the network towards base classes due to larger number of training samples and further knowledge preservation. Therefore, in Table 5 $J_{/J}$ performance mainly depends on the performance of the base classes $B_{/J}$, e.g. for Joint training method $B_{/J}$ and $J_{/J}$ show the highest 61.89 points and 43.38 points respectively among all other methods. All other methods that achieve high $B_{/J}$ performance (60.44 for IDLVQ, 59.64 for IW) accordingly reach high performance on the joint set of base and novel classes $J_{/J}$ (41.84 for IDLVQ, 41.26 for IW). At the same time these methods perform poorly on the novel samples that corresponds to $N_{/J}$ column in Table 5 (15.62 for Joint, 13.94 for IDLVQ, 13.69 for IW). On the contrary, in our framework we explicitly address novel class learning in the joint $N_{/J}$ space via base-normalized cross entropy and, thus, we are able to surpass all the previous methods on novel classes by more than 10 points, we reach 27.65 points. Our *base biased* model outperforms previous state-of-the-art models by large margin on novel classes $N_{/J}$, harmonic mean $hm_{/J}$, and sets a new benchmark for the joint classes $J_{/J}$. By *balanced hm* we show that better balance can be achieved in terms of $N_{/J}$ and $hm_{/J}$, while $B_{/J}$ and accordingly $J_{/J}$ decreases.

Tables 6, 7, and 8 are extension of Tables 2, 3, and 4 from the main paper respectively. For all the datasets we report additionally $N_{/N}$, $B_{/B}$, and $am_{/J}$.

mini-ImageNet				
method	$B_{/J}$	$N_{/J}$	$J_{/J}$	$hm_{/J}$
FT$^\diamond$	1.46	1.36	1.42	1.40
Joint$^\diamond$	**61.89**	15.62	**43.38**	24.95
iCaRL [10]$^\diamond$	24.47	7.70	17.76	11.71
UCIR [5]$^\diamond$	21.57	8.27	16.25	11.96
PN [13]$^\diamond$	56.47	11.35	38.42	18.90
ILVQ [17]$^\diamond$	56.49	11.34	38.43	18.89
SDC [19]$^\diamond$	59.87	13.30	41.24	21.77
IW [9]$^\diamond$	59.64	13.69	41.26	22.27
IDLVQ [1]	**60.44**	13.94	41.84	22.65
TOPIC [15]	-	-	24.42	-
LCwoF (base biased)	55.98	23.12	**42.84**	32.73
LCwoF (balanced hm)	47.73	**27.65**	39.70	**35.02**

Table 5: IFSL. Comparison to state-of-the-art on mini-ImageNet using metrics $B_{/J}$, $N_{/J}$, $J_{/J}$, $hm_{/J}$ and $am_{/J}$ after the last (9) task. $^\diamond$ indicates results copied from IDLVQ [1].

References

[1] Kuilin Chen and Chi-Guhn Lee. Incremental few-shot learning via vector quantization in deep embedded space. In *ICLR*, 2021. 3, 4, 6, 7

[2] Matthias Delange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Ales Leonardis, Greg Slabaugh, and Tinne Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *TPAMI*, 2021. 3

[3] Spyros Gidaris and Nikos Komodakis. Dynamic few-shot visual learning without forgetting. In *CVPR*, 2018. 3, 5

[4] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 2

[5] Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. Learning a unified classifier incrementally via rebalancing. In *CVPR*, 2019. 4, 6, 7

[6] 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. 3, 5

[7] Yaoyao Liu, Yuting Su, An-An Liu, Bernt Schiele, and Qianru Sun. Mnemonics training: Multi-class incremental learning without forgetting. In *CVPR*, 2020. 3

[8] Ameya Prabhu, Philip HS Torr, and Puneet K Dokania. GDumb: A simple approach that questions our progress in continual learning. In *ECCV*, 2020. 3

[9] Hang Qi, Matthew Brown, and David G Lowe. Low-shot learning with imprinted weights. In *CVPR*, 2018. 4, 5, 6, 7

[10] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. iCaRL: incremental classifier and representation learning. In *CVPR*, 2017. 2, 4, 6, 7

[11] Mengye Ren, Renjie Liao, Ethan Fetaya, and Richard Zemel. Incremental few-shot learning with attention attractor networks. In *NIPS*, 2019. 1, 3, 5

[12] Xiahan Shi, Leonard Salewski, Martin Schiegg, Zeynep Akata, and Max Welling. Relational generalized few-shot learning. 2020. 5

[13] Jake Snell, Kevin Swersky, and Richard S Zemel. Prototypical networks for few-shot learning. In *NIPS*, 2017. 4, 5, 6, 7

[14] 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. 3, 5

[15] Xiaoyu Tao, Xiaopeng Hong, Xinyuan Chang, Songlin Dong, Xing Wei, and Yihong Gong. Few-shot class-incremental learning. In *CVPR*, 2020. 3, 4, 7

[16] 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. 5

[17] Ye Xu, Furao Shen, and Jinxi Zhao. An incremental learning vector quantization algorithm for pattern classification. *Neural Computing and Applications*, 21(6), 2012. 4, 6, 7

[18] 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. 3

[19] 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. 4, 6, 7

method	tiered-ImageNet 5w1s						tiered-ImageNet 5w5s					
	$N_{/N}$ $(5_{/5})$	$B_{/B}$ $(200_{/200})$	$N_{/J}$ $(5_{/205})$	$B_{/J}$ $(20_{/205})$	$hm_{/J}$	$am_{/J}$	$N_{/N}$ $(5_{/5})$	$B_{/B}$ $(200_{/200})$	$N_{/J}$ $(5_{/205})$	$B_{/J}$ $(200_{/205})$	$hm_{/J}$	$am_{/J}$
PN [13]*	-	-	-	-	-	30.04	-	-	-	-	-	41.38
IW [9](i)	60.88	70.19	44.95	62.53	52.30	53.74	79.26	70.25	71.85	56.11	63.01	63.98
DFSL [3] (c)	59.52	47.53	47.32	36.10	40.96	41.71	75.89	47.98	67.94	39.08	49.61	53.51
AAN [11](c)	61.37	62.44	54.39	55.85	55.11	55.12	77.91	62.36	72.09	57.76	64.13	64.93
AAN [11](orig)	-	-	-	-	-	56.11	-	-	-	-	-	65.52
LCwoF (ours) *lim*	64.71	70.55	57.13	60.39	**58.71**	**58.76**	79.72	70.58	69.05	63.44	**66.12**	66.25
LCwoF (ours) *unlim*	64.67	70.59	57.54	60.09	**58.78**	**58.82**	80.02	70.59	70.20	63.01	**66.41**	66.61

Table 6: Comparison to state-of-the-art on tiered-ImageNet 5w1s (left) and 5w5s (right) with ResNet backbone. *lim* denotes limited access to base train samples during the third phase, for *unlim* we do not apply such restrictions. $^\circ$ indicates results copied from RGFSL [12], * indicates results from AAN [11], (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	mini-Kinetics 5w1s						mini-Kinetics 5w5s					
	$N_{/N}$ $(5_{/5})$	$B_{/B}$ $(64_{/64})$	$N_{/J}$ $(5_{/69})$	$B_{/J}$ $(64_{/69})$	$hm_{/J}$	$am_{/J}$	$N_{/N}$ $(5_{/5})$	$B_{/B}$ $(64_{/64})$	$N_{/J}$ $(5_{/69})$	$B_{/J}$ $(64_{/69})$	$hm_{/J}$	$am_{/J}$
IW [9](i)	62.57	58.42	45.56	48.56	47.01	47.06	74.61	56.67	56.92	49.17	52.76	53.05
DFSL [3] (c)	65.35	56.04	50.81	44.51	47.45	47.66	81.11	56.46	70.29	46.31	55.83	58.30
GFSV [16]	-	-	13.70	88.70	23.73	51.20	-	-	22.30	88.70	35.64	55.50
AAN [11](c)	59.36	57.99	46.13	35.96	40.41	41.05	76.83	59.49	56.99	43.21	49.15	46.18
LCwoF (ours) *lim*	55.97	64.84	47.51	50.84	**49.12**	49.18	74.76	65.06	63.65	54.55	**58.75**	59.10
LCwoF (ours) *unlim*	55.39	65.01	46.26	51.94	**48.93**	49.10	73.77	65.18	65.40	52.70	**58.37**	59.05

Table 7: Comparison to state-of-the-art on mini-Kinetics 5w1s (left) and 5w5s (right) with 2-layers MLP backbone. *lim* denotes limited access to base train samples during the third phase, for *unlim* we do not apply such restrictions. (c) denotes that we run available code on the corresponding data, (i) states for our re-implementation of the respective method.

method	UCF101 50w1s					
	$N_{/N}$ $(50_{/50})$	$B_{/B}$ $(51_{/51})$	$N_{/J}$ $(50_{/101})$	$B_{/J}$ $(51_{/101})$	$hm_{/J}$	$am_{/J}$
ProtoG [6]	-	-	52.30	75.30	61.73	63.80
LCwoF (ours)	57.13	98.97	54.41	91.41	68.22	72.91
IW [9]	54.08	85.59	45.22	76.15	56.73	60.69
LCwoF (ours) *lim*	55.98	84.16	50.78	70.72	59.11	60.75
LCwoF (ours) *unlim*	54.35	82.33	49.12	69.98	57.72	59.55

Table 8: Comparison to state-of-the-art on UCF101 50w1s with 2 layers MLP backbone on pre-extracted features. (i) states for our re-implementation of the respective method. *lim* denotes limited access to base train samples during the third phase, for *unlim* we do not apply such restrictions. Top: splits from ProtoG [6]; bottom: original UCF101 train/test splits as in [14].

mini-ImageNet									
method	1	2	3	4	5	6	7	8	9
$hm_{/J}$	60	+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 [10]$^{\diamond}$	-	8.45	13.86	14.92	13.00	14.06	12.74	12.16	11.71
UCIR [5]$^{\diamond}$	-	9.62	14.14	15.58	13.19	13.63	13.11	12.76	11.96
PN [13]$^{\diamond}$	-	9.76	14.72	16.78	19.09	20.06	19.37	18.98	18.90
ILVQ [17]$^{\diamond}$	-	9.66	16.08	17.78	20.05	20.35	19.64	19.06	18.89
SDC [19]$^{\diamond}$	-	20.51	18.79	17.36	20.47	19.21	18.27	20.79	21.77
IW [9]$^{\diamond}$	-	25.32	20.45	22.62	25.48	22.54	20.66	21.27	22.27
IDLVQ [1]	-	21.69	20.44	21.98	25.19	22.99	20.82	21.56	22.65
LCwoF (base biased)	-	25.56	30.59	27.29	28.08	29.91	27.97	30.30	32.73
LCwoF (balanced hm)	-	**41.24**	**38.96**	**39.08**	**38.67**	**36.75**	**35.47**	**34.71**	**35.02**

Table 9: IFSL. Comparison to state-of-the-art on mini-ImageNet based on harmonic mean metric between base and novel classes. $^{\diamond}$ indicates results copied from IDLVQ [1].

mini-ImageNet									
$B_{/J}$ (60)	1	2	3	4	5	6	7	8	9
FT$^{\diamond}$	64.25	32.28	20.87	6.95	3.17	3.16	1.92	1.53	1.46
Joint$^{\diamond}$	64.25	63.30	62.83	62.16	**62.18**	**62.68**	**61.86**	**61.87**	**61.89**
iCaRL [10]$^{\diamond}$	64.25	51.66	48.97	45.62	37.39	30.86	28.68	26.83	24.47
UCIR [5]$^{\diamond}$	64.25	52.87	50.16	44.78	37.48	28.75	25.58	22.97	21.57
PN [13]$^{\diamond}$	64.25	59.27	58.88	58.69	58.22	57.63	57.03	56.80	56.47
ILVQ [17]$^{\diamond}$	64.25	60.24	59.62	59.02	58.61	57.71	57.16	56.83	56.49
SDC [19]$^{\diamond}$	64.62	63.58	62.78	61.12	60.29	59.37	59.05	59.97	59.87
IW [9]$^{\diamond}$	64.71	63.52	62.96	62.13	61.17	61.27	60.63	59.86	59.64
IDLVQ [1]	**64.77**	**63.77**	**63.22**	**62.44**	61.22	61.47	60.97	60.66	60.44
LCwoF (base biased)	64.45	63.53	62.07	61.55	60.85	59.26	58.25	57.23	55.98
LCwoF (balanced hm)	64.45	57.33	53.31	52.87	51.38	48.25	47.60	47.51	47.73
$N_{/J}$	1	2	3	4	5	6	7	8	9
$\#cl$	-	5	10	15	20	25	30	35	40
FT$^{\diamond}$	-	4.07	4.49	3.75	1.93	1.53	1.77	1.61	1.36
Joint$^{\diamond}$	-	4.80	9.84	13.26	12.30	14.03	14.01	15.21	15.62
iCaRL [10]$^{\diamond}$	-	4.60	8.09	8.92	7.87	9.10	8.19	7.86	7.70
UCIR [5]$^{\diamond}$	-	5.29	8.23	9.43	8.00	8.93	8.81	8.83	8.27
PN [13]$^{\diamond}$	-	5.32	8.41	9.79	11.42	12.14	11.67	11.39	11.35
ILVQ [17]$^{\diamond}$	-	5.25	9.29	10.47	12.09	12.35	11.86	11.45	11.34
SDC [19]$^{\diamond}$	-	12.23	11.05	10.12	12.33	11.46	10.81	12.58	13.30
IW [9]$^{\diamond}$	-	15.81	12.21	13.83	16.09	13.81	12.45	12.93	13.69
IDLVQ [1]	-	13.07	12.19	13.34	15.86	14.14	12.55	13.11	13.94
LCwoF (base biased)	-	16.00	20.30	17.53	18.25	20.00	18.40	20.60	23.12
LCwoF (balanced hm)	-	**32.20**	**30.70**	**31.00**	**31.12**	**29.68**	**28.27**	**27.34**	**27.65**

Table 10: IFSL. Comparison to state-of-the-art on mini-ImageNet. Top: performance of the base samples in the joint space after each task. Bottom: performance of the novel samples in the joint space after each novel task. $^{\diamond}$ indicates results copied from IDLVQ [1].

	mini-ImageNet both$^{\uplus}$								
	1	2	3	4	5	6	7	8	9
$J_{/J}$	60	+5	+10	+15	+20	+25	+30	+35	+40
FT$^{\diamond}$	64.25	30.11	18.53	6.31	2.86	2.86	1.87	1.56	1.42
Joint$^{\diamond}$	64.25	58.80	55.26	52.38	49.71	48.37	45.91	44.68	43.38
iCaRL [10]$^{\diamond}$	64.25	48.04	43.13	38.28	30.01	24.46	21.85	19.84	17.76
UCIR [5]$^{\diamond}$	64.25	49.21	44.17	37.71	30.11	22.92	19.99	17.96	16.25
PN [13]$^{\diamond}$	64.25	55.12	51.67	48.91	46.52	44.25	41.91	40.07	38.42
ILVQ [17]$^{\diamond}$	64.25	56.01	52.43	49.31	46.98	44.37	42.06	40.11	38.43
SDC [19]$^{\diamond}$	64.62	59.63	55.39	50.92	48.30	45.28	42.97	42.51	41.24
IW [9]$^{\diamond}$	64.71	59.85	55.71	52.47	49.90	47.31	44.57	42.57	41.26
IDLVQ [1]	64.77	59.87	55.93	52.62	49.88	47.55	44.83	43.14	41.84
TOPIC [15]	61.31	50.09	45.17	41.16	37.48	35.52	32.19	29.46	24.42
LCwoF (base biased)	64.45	**59.88**	**56.10**	**52.75**	**50.20**	**47.71**	**44.97**	**43.74**	**42.84**
LCwoF (balanced hm)	64.45	55.40	50.08	48.49	46.28	42.78	41.16	40.08	39.70

Table 11: IFSL. Comparison to state-of-the-art on mini-ImageNet based on joint performance of base and novel samples in the joint space. $^{\diamond}$ indicates results copied from IDLVQ [1].