OrCo: Towards Better Generalization via Orthogonality and Contrast for Few-Shot Class-Incremental Learning

Supplementary Material

Within the supplement, we provide additional ablation studies in section 1, detailed breakdown tables and confusion matrices in section 2, an extended discussion on theory of orthogonality in section 3, formulation of contrastive losses in section 4 and additional implementation details in section 6.

1. More Ablations

What to pull & what to perturb in OrCo loss. Our OrCo loss comprises both pull and push components, influencing the distribution over the hypersphere. The pull effect is driven by cross-entropy loss (CE), where data features align with their assigned pseudo-targets. As illustrated in table 1, we show the advantage of aligning data features and pseudo-targets specifically from incremental sessions during the third phase. Introducing pseudo-targets assigned to the base classes to CE loss results in a performance degradation of approximately 1% in HM_8, due to an increased bias towards base classes. Next, we study the impact of perturbations, which create additional pushing forces, on different subsets of pseudo-targets. Our findings indicate that perturbing both incremental- and base-assigned pseudo-targets consistently hampers performance compared to perturbing only those assigned to incremental classes, resulting in about 9% improvement in HM_8. Higher base accuracy indicates that perturbations of both base- and incremental-assigned pseudo-targets provide more room for prevalent base classes, hindering the learning of novel classes and favouring base-class bias.

CE	Perturbed Pseudo-Targets	Base Acc	Inc Acc	HM_8	aHM	aACC
Inc	**Inc**	67.60	**43.80**	**53.12**	**58.12**	67.14
	Base+Inc	78.13	30.86	44.26	50.44	69.17
Base+Inc	Inc	69.65	41.85	52.30	57.76	67.90
	Base+Inc	**78.90**	29.53	43.00	48.77	**69.25**

Table 1. **What to pull & what to perturb.** CE denotes cross-entropy that pulls data features to the pseudo-targets; Inc denotes that only assigned to incremental sessions pseudo-targets participate in the CE loss, Base+Inc denotes both base- and incremental-assigned pseudo-targets. The choice of perturbed pseudo-targets can include incremental assigned pseudo-targets with unassigned pseudo-targets (Inc), or all assigned pseudo-targets with unassigned pseudo-targets (Base+Inc). Base/Inc Acc denotes accuracy from the last 8^{th} session. aACC denotes average accuracy over all sessions. Results on mini-ImageNet.

Assignment	Base Acc ↑	Base Decay ↓	aHM ↑	aACC ↑
Random	75.75	20.40	54.40	59.42
Reassignment	**83.30**	29.65	55.49	62.74
Greedy	**83.30**	**15.72**	**58.12**	**67.14**

Table 2. **Pseudo-targets assignment strategy.** Comparing our optimal assignment strategy against random assignment of pseudo-targets.

Pseudo-targets assignment strategy. In table 2, we highlight the crucial role of optimal initial alignment between pseudo-targets and class means. We compare a random assignment strategy to a Hungarian matching algorithm. Hungarian matching allows to find an optimal assignment based on distances between class means and pseudo-targets. We identify two optimal assignment strategies within hungarian matching 1) Reassignment and 2) Greedy Assignment. For the former, class means are reassigned to closest pseudo-targets at the beginning of each session whereas the later, carries forward the assignment from previous sessions.

We find that the random assignment strategy leads to a notable degradation in accuracy, particularly evident after the second phase for the base classes, amounting to approximately 8%. Greedy assignment performed better than reassignment. Despite reassignment being theoretically optimal, in practice we observe a performance drop likely due to noisy few-shot classes appearing geometrically closer to previously assigned pseudo-targets hence causing a shift of previously seen assigned classes and causing misalignment. This can be clearly seen in the loss of generalisation given a base decay of 29.65% vs 15.62% for best case.

Overall accuracy is substantially improved, demonstrating the critical contribution of the optimal assignment approach in addressing forgetting and achieving better alignment.

Number of exemplars. Due to the memory constraints inherent in FSCIL, it is common to utilize a constrained number of exemplars from the previous task. To investigate this, we conducted tests with 0, 1, and 5 exemplars, and the results are presented in table 3. We note that even with just 1 exemplar, our model achieves a performance improvement of 2.84% compared to our strong baseline, the IW method.

2. More Results

Base and incremental accuracy breakdown. We show our SOTA results with a base and incremental session ac-

#	Session-wise Harmonic Mean (%) ↑								aHM	aACC
	1	2	3	4	5	6	7	8		
0	69.3	47.9	42.3	34.9	31.2	28.0	24.8	24.4	37.8	54.6
1	64.4	53.4	48.7	48.9	45.4	40.5	40.7	43.5	48.2	64.9
5	**68.7**	**63.9**	**60.9**	**58.0**	**55.3**	**52.4**	**52.7**	**53.1**	**58.1**	**67.1**

Table 3. Number of saved exemplars (#) for incremental sessions.

curacy breakdown for all sessions in tables 6 to 8. Note that, for Imprinted Weights (IW) [7] we use the implementation of a decoupled learning strategy from [3] and we initialise the method with our model pretrained during Phase 1. We note that LIMIT [12] has been unintentionally left out from figure 4 (main) for CUB200. Furthermore, we report the average accuracy metric to provide a comprehensive overview of our results. Note that the considerable data imbalance between base and incremental classes has an impact on this metric. The improved accuracy especially in the base classes, as illustrated in the breakdown tables, contributes to the overall enhancement of this measure.

Confusion matrices. In figure 2, we compare session-wise confusion matrices for a) OrCo, b) NC-FSCIL [8], and c) BiDist [10] during the final session of mini-ImageNet. Our benchmark involves assessing OrCo against its two closest competitors. OrCo plays a crucial role in finding a delicate balance between preserving knowledge of base classes and efficiently learning new ones, showcasing significantly enhanced learning capabilities in incremental classes. Notably, other methods exhibit a strong bias towards the base classes due to low transferability, while our pretraining session establishes a robust backbone. Moreover, our space reservation scheme, along with strong separation using perturbed targets and robust contrastive learning, enables us to learn a highly performant learner.

3. Theory of Orthogonality

This section covers the mathematical theory of orthogonality of independent vectors in high dimensional space. Let $X_1, X_2, \ldots, X_n$ be independent and identically distributed (i.i.d.) random vectors sampled from a normal distribution with mean 0 and variance 1. These vectors are in $\mathbb{R}^n$. The claim is that these vectors are mutually orthogonal on the unit sphere. To prove this, let's first establish that the vectors are normalized to have a length of 1.

Given a vector $X_i = (X_{i1}, X_{i2}, \ldots, X_{in})$, its length is given by:

$$\|X_i\| = \sqrt{X_{i1}^2 + X_{i2}^2 + \ldots + X_{in}^2}$$

Since each component X_{ij} is independently sampled from a normal distribution with mean 0 and variance 1, the expected value of X_{ij}^2 is 1. Therefore, the expected value of the length squared is:

$$\mathbb{E}[\|X_i\|^2] = \mathbb{E}[X_{i1}^2 + X_{i2}^2 + \ldots + X_{in}^2] = n$$

This means that $\frac{1}{\sqrt{n}} X_i$ has a length of 1 in expectation. Now, let's consider the inner product of two different vectors X_i and X_j (where $i \neq j$):

$$\mathbb{E}[X_i \cdot X_j] = \sum_{k=1}^{n} \sum_{k'=1}^{n} \mathbb{E}[X_{ik} \cdot X_{jk'}]$$

Since X_{ik} and X_{jk} are independent for $i \neq j$, the cross-terms in the summation will have an expected value of 0, and the only non-zero terms will be the ones where $k = k'$, resulting in:

$$\mathbb{E}[X_i \cdot X_j] = \sum_{k=1}^{n} \mathbb{E}[X_{ik} \cdot X_{jk}] = \sum_{k=1}^{n} \mathbb{E}[X_{ik}^2]\delta_{ij} = \delta_{ij} \cdot n$$

where δ_{ij} is the Kronecker delta. Therefore, the expected value of the inner product is n if $i = j$ and 0 otherwise. This means that the vectors are orthogonal in expectation. It is important to note that this property specifically holds for vectors drawn from a normal distribution with mean 0 and variance 1.

The figure 1 shows the practical effects of the above theory which yields near orthogonality but not perfect orthogonality. Only near feature dimension = 2^{15} do we generate nearly orthogonal vectors. Which would lead to the projection head having ~ 68 million parameter making our framework incomparable to other methods. Explicit orthogonality as we have shown previously in table 3 (main) yields better results which is why we explore this practical constraint.

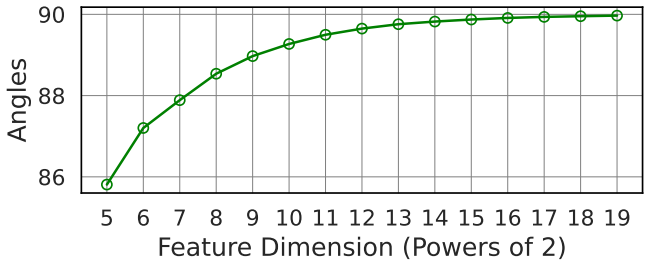


Figure 1. Practical effect of dimensions on average pair-wise angle of a 100 independant random vectors.

4. Contrastive losses

Supervised Contrastive Loss. Given a set of sample-label pairs $(x_i, y_i) \in Z^{SCL}$, we define the positive set P_i^{SCL} for x_i as the collection of pairs (x_j, y_j) where j varies over all instances such that $y_j = y_i$. Correspondingly, the negative

set N_i^{SCL} is defined as $Z^{SCL} \setminus P_i^{SCL}$. Then, supervised contrastive loss (SCL) is defined as:

$$\mathcal{L}_{SCL}(i;\theta) = \frac{-1}{|P_i^{SCL}|} \sum_{x_j \in P_i^{SCL}} \log \frac{\exp(x_i \cdot x_j/\tau)}{\sum_{x_k \in N_i^{SCL}} \exp(x_i \cdot x_k/\tau)}.$$

Self-Supervised Contrastive Loss. Given a set of samples $x_i \in Z^{SSCL}$, we define a positive for x_i as $A(x_i)$ where $A(\cdot)$ is a random transformation. Then, self-supervised contrastive loss (SSCL) loss is defined as:

$$\mathcal{L}_{SSCL}(i;\theta) = -1 \cdot \log \frac{\exp(x_i \cdot A(x_i)/\tau)}{\sum_{x_k \in Z^{SSCL}, i \neq k} \exp(x_i \cdot x_k/\tau)}.$$

5. Orthogonality Loss

In this section, we further expand on the orthogonality loss in our framework. We employ the orthogonality loss as an implicit geometric constraint on the set O predicated on the current batch. O contains the following: mean features for all classes within batch μ_j, assigned targets not represented within batch and all unassigned targets T_u^i.

Formally, let us assume the session i with data D^i and classes C^i. In order to define a set O we compute some preliminaries. Firstly, for every training batch B we compute the within-batch mean for all data features. This is computed as:

$$\mu_j = \frac{1}{|C^j|} \sum_{k=0}^{|C^j|} z_k, \forall j \in C_B^i \tag{1}$$

where $C_B^i \in C^i$ refers to all classes appearing in this particular batch. The combined set of all means can then be termed M_B. For the classes that did not appear in this batch we define as $\neg C_B^i = C^i \setminus C_B^i$. Subsequently we define a mapping function from seen class labels to the assigned pseudo target.

$$h : C^i \rightarrow T^i \tag{2}$$

We incorporate the remaining real data by adding the following set of assigned pseudo targets as $\neg T_B^i = h(\neg C_B^i)$. For completeness, we combine the above with the unassigned targets T_u^i leading to the following definition of O:

$$O = \{M_B \cup \neg T_B^i \cup T_u^i \mid B\} \tag{3}$$

Finally, the orthogonality loss takes the form:

$$\mathcal{L}_{ORTH}(O) = \frac{1}{|O|} \sum_{i=1}^{|O|} \log \sum_{j=1}^{|O|} e^{o_i \cdot o_j/\tau_o}, o_i, o_j \in O \tag{4}$$

In essence, the orthogonality loss introduces a subtle geometric constraint between real class features and the pseudo-targets. Additionally the batch-wise construction helps regularise the loss function.

Although the improvement from the Orthogonality loss are not prominent like in PSCL, it remains measurable and consistent across settings (e.g. in table 2 of the original paper, line 1 vs line 2, line 3 vs line 4), and thus contributes to our results. Additionally, we would like to highlight that each loss term in equation 6 in the main paper incorporates orthogonality constraints either implicitly or explicitly. E.g. the orthogonality enforced on the pseudo-targets implies an implicit orthogonality among the features as they are pushed to these targets.

6. Further Implementation Details

Model parameters. Our representation learning framework is composed of:

- Encoder/Backbone Network (f) We use the ResNet18 and 12 [4] variant for our experiments. Depending on the variant of the encoder, the representation vector has output dimensions D_E = 512 or 640, respectively. In table 4 we compare our choice of architecture against other FSCIL studies.
- Projection Network (projector)(g). Following the encoder, a projection MLP maps the representation vector from f to the contrastive subspace. We use a two layer projection head with a hidden dimension of size 2048. By convention output projections are normalised to the hypersphere. For simplicity, the dimension of the projected hyper sphere is initialised as $d = 2\lceil log(C) \rceil$ where C is the total number of classes for our FSCIL task. Following convention, we assume normalised feature vectors.

Training details. For CUB200 dataset, we skip the pretraining phase given that it is common in literature to use pretrained ImageNet weights for the backbone [3, 8, 10, 11]. For any incremental sessions (> 0) we finetune only the projection head from phase 2. For the PSCL loss we choose a perturbation magnitude λ_{pert} = 1e-2 for our experiments and perturb only the incremental targets and unassigned targets. We train the projection head for 10 epochs in the 0-th incremental session and 100 epochs for all following sessions. Cosine scheduling is employed with warmup for a few epochs for all our phases with a maximum learning rate set to 0.4 in phase 1, 0.25 in phase 2 and 0.1 in phase 3. Given the equation 4 (main), we double our batch size by over sampling target perturbations such that the number of perturbed targets is always the same as the original training batch size. Orthogonality loss is applied batch wise. More concretely, the orthogonality loss takes as input, the within class average features inside a batch along with any other pseudo targets not in the batch. Cross entropy is applied

Method	Model		
	mini-ImageNet	CIFAR100	CUB200
IW [7]	ResNet18	ResNet12	ResNet18
FACT [11]	ResNet18	ResNet20	ResNet18
CEC [9]	ResNet18	ResNet20	ResNet18
C-FSCIL [5]	ResNet12	ResNet12	-
LIMIT [12]	ResNet18	ResNet20	ResNet18
LCwoF [6]	ResNet18	-	-
BiDist [10]	ResNet18	ResNet18	ResNet18
NC-FSCIL [8]	ResNet12	ResNet12	ResNet18
OrCo	ResNet18	ResNet12	ResNet18

Table 4. ResNet architectures used in FSCIL literature

exclusively to incremental class data as base classes are already aligned.

For pseudo target generation we employ an SGD optimiser, with 1e-2 learning rate for 2000 epochs to minimize the loss. For CUB200 the dimension of the projection head is higher which constitutes longer training cycle to fully orthogonalize the pseudo targets. CUB200 was most susceptible to forgetting for which reason we ensured that base classes were incorporated inside the CE loss component of the loss function. With the lack of a pretrain step in CUB200, we must also finetune the backbone during phase 2 to align base classes while also capturing specific representation which is important for learning effectively on a fine-grained dataset. For pretraining, we use RandAug [2] for mini-Imagenet and AutoAugment policy [1] for CIFAR100. Additionally, following the implementation of [11] we apply auto augment policy for CIFAR100 during the incremental sessions as well.

Architecture	Parameter Count (million)	aHM	Base Acc
ResNet-18	**12.49**	58.12	83.30
ResNet-12	15.06	**59.30**	**83.65**

Table 5. Comparing ResNet-12 to ResNet-18

7. ResNet 12 with Mini-ImageNet

In this section we measure the efficacy of our method on a different backbone. More specifically we train a ResNet-12 backbone used by [8] with our method. In table 5 we show the results. Our reported results with ResNet-18 are 58.12, while the results with ResNet-12 are 59.30 indicating an improvement with the wider ResNet-12 architecture. ResNet-18 remains as our elected architecture due to its prominence in prior works, low parameter count, while still maintaining state-of-the art performance.

References

[1] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation policies from data. *arXiv preprint arXiv:1805.09501*, 2018. 4

[2] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *CVPR workshops*, 2020. 4

[3] Matthias De Lange and Tinne Tuytelaars. Continual prototype evolution: Learning online from non-stationary data streams. In *ICCV*, 2021. 2, 3

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 3

[5] Michael Hersche, Geethan Karunaratne, Giovanni Cherubini, Luca Benini, Abu Sebastian, and Abbas Rahimi. Constrained few-shot class-incremental learning. In *CVPR*, 2022. 4, 5, 6

[6] A Kukleva, H Kuehne, and B Schiele. Generalized and incremental few-shot learning by explicit learning and calibration without forgetting. in 2021 ieee. In *ICCV*, 2021. 4, 5

[7] Hang Qi, Matthew Brown, and David G Lowe. Low-shot learning with imprinted weights. In *CVPR*, 2018. 2, 4, 5, 6

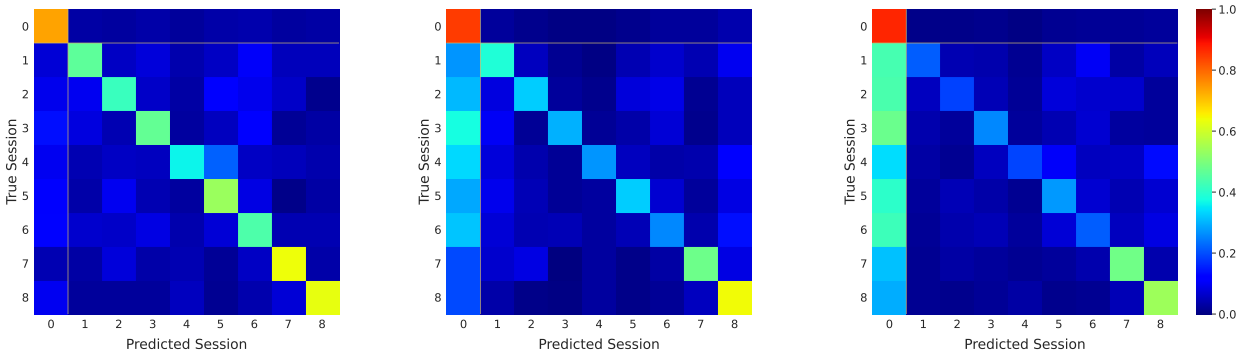
[8] Yibo Yang, Haobo Yuan, Xiangtai Li, Zhouchen Lin, Philip Torr, and Dacheng Tao. Neural collapse inspired feature-classifier alignment for few-shot class-incremental learning. In *ICLR*, 2023. 2, 3, 4, 5, 6

[9] Chi Zhang, Nan Song, Guosheng Lin, Yun Zheng, Pan Pan, and Yinghui Xu. Few-shot incremental learning with continually evolved classifiers. In *CVPR*, 2021. 4, 5, 6

[10] Linglan Zhao, Jing Lu, Yunlu Xu, Zhanzhan Cheng, Dashan Guo, Yi Niu, and Xiangzhong Fang. Few-shot class-incremental learning via class-aware bilateral distillation. In *CVPR*, 2023. 2, 3, 4, 5, 6

[11] Da-Wei Zhou, Fu-Yun Wang, Han-Jia Ye, Liang Ma, Shiliang Pu, and De-Chuan Zhan. Forward compatible few-shot class-incremental learning. In *CVPR*, 2022. 3, 4, 5, 6

[12] Da-Wei Zhou, Han-Jia Ye, Liang Ma, Di Xie, Shiliang Pu, and De-Chuan Zhan. Few-shot class-incremental learning by sampling multi-phase tasks. *IEEE TPAMI*, 2022. 2, 4, 5, 6



(a) OrCo (b) NC-FSCIL (c) BiDist

Figure 2. Visualising the session-wise confusion matrix for mini-ImageNet using a) OrCo, b) NC-FSCIL [8], and c) BiDist [10]. Each matrix demonstrates the predictive accuracy for base and incremental sessions, separated by yellow lines. High values on the diagonal (indicative of correct session predictions) are contrasted with low off-diagonal values (representing misclassifications). The first column in each matrix highlights potential prediction bias towards base classes. Our method's performance, as illustrated, demonstrates both high diagonal accuracy and a balanced approach in reducing base class bias, as compared to the results of the competing methods.

Method	Class Group	Session-wise Accuracy (%)									Means	aACC
		0	1	2	3	4	5	6	7	8		
IW [7]	Base	**83.10**	**81.17**	**80.58**	**79.93**	**79.55**	**78.88**	**78.38**	**78.08**	**77.25**	**79.66**	68.77
	Incremental	-	35.60	31.30	32.27	32.65	31.16	29.13	29.91	32.40	31.80	
FACT [11]	Base	75.78	75.22	74.83	74.47	74.30	74.05	73.82	73.47	73.37	74.37	60.86
	Incremental	-	16.60	17.10	17.20	15.15	13.24	11.93	12.17	12.43	14.48	
CEC [9]	Base	72.17	70.77	70.05	69.53	69.27	68.95	68.65	68.25	67.95	69.51	59.57
	Incremental	-	20.60	20.60	19.93	19.75	17.68	16.63	16.80	17.18	18.65	
C-FSCIL [5]	Base	76.60	76.15	74.70	73.82	73.18	71.10	70.08	68.27	67.58	72.39	61.21
	Incremental	-	5.20	11.90	17.80	20.40	23.08	23.37	26.06	26.35	19.27	
LIMIT [12]	Base	73.27	70.43	69.47	68.68	68.18	67.73	67.30	67.07	66.68	68.76	58.04
	Incremental	-	29.80	24.00	22.73	21.85	19.72	19.40	19.71	20.18	22.17	
LCwoF [6]	Base	64.45	57.33	53.31	52.87	51.38	48.25	47.60	47.51	47.73	52.27	46.80
	Incremental	-	32.20	30.70	31.00	31.12	29.68	28.27	27.34	27.65	29.75	
BiDist [10]	Base	74.67	73.63	72.50	71.03	70.63	70.37	68.70	67.98	69.25	70.97	61.25
	Incremental	-	32.60	31.40	30.33	30.30	25.48	25.23	27.09	25.10	28.44	
NC-FSCIL [8]	Base	84.37	78.25	76.00	75.73	74.80	75.42	75.52	75.13	74.77	76.67	67.82
	Incremental	-	51.80	51.00	44.33	41.20	37.04	34.20	33.37	32.77	40.71	
OrCo	Base	83.30	76.40	74.10	72.00	71.20	70.50	69.20	68.10	67.60	72.49	67.14
	Incremental	**-**	**62.40**	**56.10**	**52.80**	**48.90**	**45.40**	**42.20**	**42.90**	**43.80**	**49.31**	

Table 6. Base and Incremental accuracy shown per session for mini-ImageNet.

Method	Class Group	Session-wise Accuracy (%)									Means	aACC
		0	1	2	3	4	5	6	7	8		
IW [7]	Base	78.58	75.45	75.15	74.65	74.38	74.07	73.80	73.55	73.03	74.74	64.05
	Incremental	-	29.40	31.90	28.53	27.60	26.36	27.20	26.91	25.88	27.97	
C-FSCIL [5]	Base	77.35	76.70	76.17	75.52	75.35	74.22	73.92	73.63	72.87	75.08	61.42
	Incremental	-	19.80	17.40	16.20	13.65	14.92	14.53	13.66	14.00	15.52	
LIMIT [12]	Base	79.63	75.40	74.47	73.70	73.22	72.52	72.22	72.02	71.32	73.83	61.66
	Incremental	-	27.20	24.60	21.47	21.00	20.76	21.10	20.54	20.13	22.10	
CEC [9]	Base	72.93	72.13	71.42	70.72	70.12	69.20	68.67	68.43	67.75	70.15	59.53
	Incremental	-	29.60	27.00	22.60	21.80	22.40	22.13	21.66	21.08	23.53	
FACT [11]	Base	78.72	76.23	75.30	74.63	73.90	73.07	72.58	72.28	71.73	74.27	62.55
	Incremental	-	29.80	25.60	21.20	20.70	20.24	22.33	21.69	21.95	22.94	
BiDist [10]	Base	69.68	68.45	67.55	66.47	65.80	64.87	64.77	64.27	64.50	66.26	56.91
	Incremental	-	36.80	31.20	28.80	25.75	24.36	22.87	22.43	20.35	26.57	
NC-FSCIL [8]	Base	**82.52**	**79.55**	**78.63**	**77.98**	**77.60**	**75.98**	**74.45**	**75.18**	**73.98**	**77.32**	67.50
	Incremental	-	44.00	41.60	36.47	31.95	31.32	33.97	31.31	29.30	34.99	
OrCo	Base	80.08	67.37	68.12	63.30	63.40	63.93	61.45	61.08	58.22	65.22	62.11
	Incremental	-	**77.60**	**60.20**	**51.67**	**48.90**	**45.80**	**48.23**	**44.94**	**43.15**	**52.56**	

Table 7. Base and Incremental accuracy shown per session for CIFAR100.

Method	Class Group	Session-wise Accuracy (%)											Means	aACC
		0	1	2	3	4	5	6	7	8	9	10		
IW [7]	Base	67.53	67.07	66.83	66.55	66.48	66.31	66.13	65.99	65.85	65.85	65.75	66.39	59.72
	Incremental	-	29.03	27.74	25.00	25.95	26.61	26.51	25.45	24.34	26.08	26.93	26.36	
CEC [9]	Base	75.64	74.27	73.88	73.64	72.66	72.31	71.75	71.09	70.98	70.64	70.46	72.48	61.33
	Incremental	-	45.16	41.52	33.1	36.34	33.1	34.17	34.34	32.96	34.41	34.16	35.93	
BiDist [10]	Base	75.98	74.23	73.71	73.85	73.08	72.35	71.68	71.65	71.68	71.12	70.36	72.70	62.91
	Incremental	-	55.20	46.11	38.19	41.67	37.62	38.11	39.28	36.81	39.73	39.83	41.26	
FACT [11]	Base	77.23	75.04	74.83	74.79	74.41	74.20	73.60	73.46	73.22	72.87	72.84	74.22	64.42
	Incremental	-	53.05	47.17	38.08	40.21	38.37	39.66	40.25	38.30	40.11	39.93	41.51	
LIMIT [12]	Base	79.63	**79.02**	**78.81**	**78.77**	**78.39**	**78.04**	**77.72**	**77.51**	**77.24**	**76.68**	73.46	**77.75**	65.49
	Incremental	-	49.82	44.88	37.38	39.35	37.35	38.40	40.01	38.47	39.92	42.15	40.77	
NC-FSCIL [8]	Base	**80.45**	76.89	77.62	78.63	77.23	77.13	76.85	76.29	76.61	75.94	**76.19**	77.26	67.29
	Incremental	-	66.67	45.41	42.59	45.88	41.72	44.11	44.99	41.16	42.66	43.07	45.83	
OrCo	Base	75.59	65.54	63.79	66.76	64.91	65.05	65.50	65.01	66.24	66.27	66.62	66.48	62.36
	Incremental	-	**79.93**	**65.37**	**53.47**	**55.41**	**51.03**	**51.31**	**50.90**	**47.69**	**49.70**	**49.25**	**55.41**	

Table 8. Base and Incremental accuracy shown per session for CUB200.