Supplemental Material

Algorithms

In Algorithm 2 and 1, we respectively present pseudo code for training and test for the incremental sessions. Before using either, we train the supervised feature extractor on the classes from the base session. Then augmented versions of images belonging to base classes are passed through the supervised and self-supervised feature extractors. The fusion layer is then trained on the concatenated features. We obtain the mean/variance for the base classes, which are used to synthesize data for subsequent incremental sessions. For each incremental session, the mean/variance for new classes are further included in the maintained sets for mean and variance.

Results on CIFAR100

We listed results for CUB200 and miniImageNet in main paper comparing FeSSSS against state of the art FSCIL approaches. Tab. 5 provides our comparison against other methods on CIFAR100 data set. While SPPR is the second best performer on miniImageNet (Tab. 2) and CIFAR100 (Tab. 5) data sets, they are among the worst performers on the more challenging CUB200 data set (Tab. 1). We should further note that while [63] demonstrate large gap in their SPPR-ive and SPPR versions on CIFAR100/miniImageNet data sets, there is no gain when SPPR method is used for more challenging CUB200 as admitted by authors in their paper: *"Since the number of incremental classes is twice than each of the above datasets, the forgetting rate at each session is much higher. It can be seen that the difficulty increases as the number of incremental classes and sessions increases. Different from that on above datasets, the initial classification accuracy (68.68%) is close to the best result ("Ours*"), so we only report one result ("Ours") on this dataset."* Specifically, the number of images in base session are 3000 for 100 base classes (30 images per class) compared to miniImageNet/CIFAR100 where base session is comprised of 30000 images belonging to 60 base classes (500 images per class). Additionally the number of incremental classes per session are twice (10) for CUB200 than that for miniImageNet/CIFAR100 (5). In our tables, we identify SPPR's ("Ours*") as SPPR and ("Ours") as SPPR-ive.

As evident from table, FeSSSS still outperforms other FSCIL including SPPR. However, the performance gain is smaller when compared to the gain on CUB200 and miniImageNet data sets. In our hypothesis, this is due to the resolution gap between the supervised and self-supervised feature extractors' training sets. Specifically all the explored self-supervised methods in this paper have been trained with conventional crop size of 224×224. For the CUB200 data set, the original images are of high resolution and the supervised model is trained on 224×224 crops. Similarly, for the miniImageNet data set the training images are first resized to 92 and then random crops of 84×84 are used for training. Whereas, in the case of the CIFAR100 data set the original images are of size 32×32 and a custom ResNet-20 is trained for the supervised model. Compared to that, the self-supervised model still operated at a resolution of 224×224, so an image of 32×32 is up-sampled to that resolution, generating data mostly due to interpolation with limited frequencies. This makes the self-supervised features, which were trained to include high-frequencies, not as well matched and hence not adding as much as in other data sets.

Self-Supervised Features

Tabs. 6, 7, and 8 provide the results of an ablation study on the use of different self-supervised features for FSCIL. Specifically, DeepCluster-v2 [11], SwAV [12], Moco-v2 [15], and SeLa-v2 [1] are explored for CUB200, miniImageNet, and CIFAR100 data sets. In each case the self-supervised features are learned on the ImageNet-2012 data set [50] with a ResNet-50 model. We can note that not all self-supervised features perform equally well for FSCIL, *e.g.*, while DeepCluster-v2 and SwAV features combined with supervised features outperform CEC, Moco-v2 and SeLa-v2 have lower performance. In the main paper, we avoided the overlap between miniImageNet and ImageNet by extracting the self-supervised features from OpenImages for experiment on miniImageNet. Tab. 7 demonstrates learning self-supervised features on ImageNet (super-set of miniImageNet) results in performance even superior to the fused features.

Algorithm 1 Inference of FeSSSS for incremental sessions.

Input: Pre-trained frozen supervised ($f_s(x;\theta_s)$) and self-supervised feature extractors ($f_{ss}(x;\theta_{ss})$). Test data $\mathcal{D}^i_{test}$ up to current (i-th) incremental session, comprised of classes $\mathcal{C}^0 \cup \mathcal{C}^1 \cdots \cup \mathcal{C}^i$ seen so far. Current classification module $l_c(\theta^i_c)$.
Output: Predicted class labels for all samples in $\mathcal{D}^i_{test}$.

```
1:  Initialize accumulator to hold all predictions P ← []
2:  for each image x in D^i_test do
3:      x̂_s ← f_s(CenterCrop(x); θ_s)
4:      x̂_ss ← f_ss(CenterCrop(x); θ_ss)
5:      x̄_s ← L^2-Normalize(x̂_s)
6:      x̄_ss ← L^2-Normalize(x̂_ss)
7:      x̄_t ← x̄_s | x̄_ss
8:      p_c ← l_c(x̄_t; θ^i_c)
9:      Accumulate p_c in accumulator P
10: end for
11: return Accumulator P containing predicted labels.
```

Algorithm 2 Training of FeSSSS for incremental sessions.

Input: Frozen supervised ($f_s(x;\theta_s)$) and self-supervised feature extractors ($f_{ss}(x;\theta_{ss})$). Training data $\mathcal{D}^i_{train}$ for current (i-th) incremental session. Number of epochs ($numEpochs$) for the current session. Classification module trained on previous session $l_c(\theta_c^{i-1})$. Centroids (μ) and variances (σ) of n old classes seen so far.
Parameters: N, K, numAug
Output: Updated classification module $l_c(\theta_c^i)$, updated μ and σ.

Initialize θ_c^i with θ_c^{i-1}.
Extend $l_c(\theta_c^i)$ with N nodes.
Initialize new connections with random weights.
for each new class m in $\mathcal{D}^i_{train}$ **do**
    Empty list containing features: $A_m \leftarrow []$
end for
Empty list containing augmented & generated data: $AG \leftarrow []$
for each image x in $\mathcal{D}^i_{train}$ **do**
    $j \leftarrow 0$
    for $j <$ numAug **do**
        $\hat{x}_s \leftarrow f_s(RandAug(x);\theta_s)$
        $\hat{x}_{ss} \leftarrow f_{ss}(RandAug(x);\theta_{ss})$
        $\bar{x}_s \leftarrow L^2$-Normalize($\hat{x}_s$)
        $\bar{x}_{ss} \leftarrow L^2$-Normalize($\hat{x}_{ss}$)
        $\bar{x}_t \leftarrow \bar{x}_s | \bar{x}_{ss}$
        extend AG with $\bar{x}_t$, $AG \leftarrow [AG; \bar{x}_t]$
        extend respective A_m with $\bar{x}_t$, $A_m \leftarrow [A_m; \bar{x}_t]$
    end for
end for
$k \leftarrow 0$
while $k < n$ **do**
    using respective mean (μ_k) and variance (σ_k) generate data D_g with Gaussian Generator
    for each generated sample $x_g \in D_g$ **do**
        extend AG with x_g, $AG \leftarrow [AG; x_g]$
    end for
    $k \leftarrow k + 1$
end while
$e \leftarrow 0$
while $e < numEpochs$ **do**
    for each random batch in epoch e **do**
        using selected batch X_r from AG, train $l_c(\theta_c^i)$
    end for
    $e \leftarrow e + 1$
end while
for each new class m in $\mathcal{D}^i_{train}$ **do**
    Compute mean (μ_m) and variance (σ_m) using data in respective accumulator A_m
    Extend (μ) and (σ) with (μ_m) and (σ_m)
end for
Updated class counter $n \leftarrow n + N$
return Updated $l_c(\theta_c^i)$, μ, and σ

Table 5. Comparison of FeSSSS with the state-of-the-art on CIFAR100 data set, DeepCluster-v2 [11] trained on ImageNet-2012 has been used as self-supervised feature extractor. ‡ indicates results copied from CEC [59], * identifies the few-shot approaches adapted by [59] for FSCIL, and † shows the results for approaches taken from their respective papers. Further ◇ identifies that the results have been approximated from graphs since tabular results are unavailable from respective papers.

Method	Acc. in each session (%) ↑									Avg. ↑	our relative improvement
	0	1	2	3	4	5	6	7	8		
Ft-CNN‡	64.1	36.91	15.37	9.8	6.67	3.8	3.7	3.14	2.65	16.23	**+44.69**
iCaRL‡ [47]	64.1	53.28	41.69	34.13	27.93	25.06	20.41	15.48	13.73	32.86	**+28.06**
EEIL‡ [13]	64.1	53.11	43.71	35.15	28.96	24.98	21.01	17.26	15.85	33.79	**+27.13**
NCM‡ [29]	64.1	53.05	43.96	36.97	31.61	26.73	21.23	16.78	13.54	34.21	**+26.71**
TOPIC‡ [51]	64.1	55.88	47.07	45.16	40.11	36.38	33.96	31.55	29.37	42.62	**+18.30**
LEC-Net† [57]	64.1	53.23	44.19	41.87	38.54	39.54	37.34	34.73	34.73	43.14	**+17.78**
SS-NCM-CNN†◇ [16]	64.1	62.22	61.11	58.0	54.22	50.66	48.88	46.0	44.44	54.40	**+6.52**
SPPR-ive†◇ [63]	64.1	66.66	63.33	57.66	54.33	50.66	48.33	45.66	43.0	54.85	**+6.07**
Decoupled-DeepEMD‡ [58]*	69.75	65.06	61.2	57.21	53.88	51.40	48.80	46.84	44.41	55.39	**+5.53**
ERL† [18]	73.62	66.79	63.67	60.54	56.98	53.63	50.92	48.73	46.33	57.91	**+3.01**
Decoupled-Cosine‡ [52]*	74.55	67.43	63.63	59.55	56.11	53.80	51.68	49.67	47.68	58.23	**+2.69**
ERL++† [18]	73.62	68.22	65.14	61.84	58.35	55.54	52.51	50.16	48.23	59.29	**+1.63**
CEC‡ [59]	73.07	68.88	65.26	61.19	58.09	55.57	53.22	51.34	49.14	59.52	**+1.40**
SPPR†◇ [63]	76.33	72.33	67.33	63.33	59.0	55.33	53.0	50.33	47.33	60.47	**+0.45**
FeSSSS (Ours)	75.35	70.81	66.7	62.73	**59.62**	**56.45**	**54.33**	**52.10**	**50.23**	**60.92**	

Table 6. Ablation conducted on CUB200 for various self-supervised features, *i.e.*, DeepCluster-v2 [11], SwAV [12], Moco-v2 [15], and SeLa-v2 [1]. We notice that not all self-supervised features are equally well suited for FSCIL. Also, the Gaussian Generator's synthetic data for old classes improves performance over just centroid.

Method	self-supervised model	features	feature fusion layer	Gaussian generation	variance	Acc. in each session (%) ↑											Avg. ↑	improvement over CEC
						0	1	2	3	4	5	6	7	8	9	10		
CEC	-	-	-	-	-	75.85	71.94	68.50	63.5	62.43	58.27	57.73	55.81	54.83	53.52	52.28	61.33	-
FeSSSS (Ours)	**DeepClust-v2**	self-supervised	✓	✗	n.a.	73.21	65.09	62.24	58.26	54.76	51.82	49.78	47.60	44.54	44.46	43.56	54.12	**-7.21**
		supervised	✓	✗	n.a.	74.37	67.86	64.54	60.81	58.19	54.62	53.22	51.61	49.55	48.63	46.63	57.27	**-4.06**
		concat	✓	✗	n.a.	79.60	72.19	69.47	65.63	63.55	58.78	58.01	56.64	54.09	53.65	52.81	62.22	**+0.89**
		concat	✓	✓	vector	79.60	72.70	70.02	66.33	63.87	59.40	58.19	57.09	54.78	54.62	52.91	62.68	**+1.35**
		concat	✓	✓	scalar	79.60	73.46	70.32	66.38	63.97	59.63	58.19	57.56	55.01	54.31	52.98	62.85	**+1.52**
	SwAV	self-supervised	✓	✗	n.a.	70.70	63.76	59.38	57.45	53.12	50.27	48.11	45.97	43.37	43.46	41.78	52.48	**-8.85**
		supervised	✓	✗	n.a.	74.37	67.86	64.54	60.81	58.19	54.62	53.22	51.61	49.55	48.63	46.63	57.27	**-4.06**
		concat	✓	✗	n.a.	79.12	72.98	69.41	65.77	63.13	58.83	57.62	56.56	53.47	53.89	52.69	62.13	**+0.80**
		concat	✓	✓	vector	79.12	73.87	69.70	66.36	63.33	59.52	58.08	57.34	54.36	54.09	53.22	62.63	**+1.30**
		concat	✓	✓	scalar	79.12	74.51	70.20	66.55	63.55	59.29	58.40	57.29	54.11	54.05	52.84	62.71	**+1.38**
	Moco-v2	self-supervised	✓	✗	n.a.	58.72	50.97	47.37	45.03	40.71	38.71	37.08	36.02	33.15	32.56	31.27	41.05	**-20.28**
		supervised	✓	✗	n.a.	74.37	67.86	64.54	60.81	58.19	54.62	53.22	51.61	49.55	48.63	46.63	57.27	**-4.06**
		concat	✓	✗	n.a.	77.51	70.66	67.11	63.65	60.77	56.88	55.57	54.53	51.86	51.43	49.98	59.99	**-1.34**
		concat	✓	✓	vector	77.51	71.61	68.10	64.37	61.24	57.65	56.39	55.28	52.76	52.45	51.01	60.76	**-0.57**
		concat	✓	✓	scalar	77.51	72.09	68.57	64.61	62.01	58.18	56.69	55.40	52.86	52.32	51.12	61.03	**-0.30**
	SeLa-v2	self-supervised	✓	✗	n.a.	61.73	53.73	51.04	48.90	43.29	41.70	39.68	37.67	35.03	34.96	33.20	43.72	**-17.61**
		supervised	✓	✗	n.a.	74.37	67.86	64.54	60.81	58.19	54.62	53.22	51.61	49.55	48.63	46.63	57.27	**-4.06**
		concat	✓	✗	n.a.	77.44	70.15	67.43	63.92	60.84	57.07	56.26	54.65	51.95	51.94	50.60	60.20	**-1.13**
		concat	✓	✓	vector	77.44	71.65	68.22	64.67	61.74	57.60	56.71	55.03	52.84	52.60	51.43	60.90	**-0.43**
		concat	✓	✓	scalar	77.44	71.68	68.48	64.56	61.86	57.85	56.52	54.89	52.59	52.63	51.43	60.90	**-0.43**

Table 7. Ablation conducted on miniImageNet for various self-supervised features, *i.e*., DeepCluster-v2 [11], SwAV [12], Moco-v2 [15], and SeLa-v2 [1]. Since all the self-supervised models are learned using ImageNet-2012 which is a super-set of miniImageNet, we notice training classification heads solely on these self-supervised features outperform the concatenated and even supervised features as well. Since a good separation using much more data has already been learned by training on full ImageNet-2012, these self-supervised features are resulting in astonishing performance due to overlapping classes.

Method	self-supervised model	features	feature fusion layer	Gaussian generation	variance	Acc. in each session (%) ↑									Avg. ↑	improvement over CEC
						0	1	2	3	4	5	6	7	8		
CEC	-	-	-	-	-	72.00	66.83	62.97	59.43	56.70	53.73	51.19	49.24	47.63	57.74	-
FeSSSS (Ours)	**DeepClust-v2**	self-supervised	✓	✗	n.a.	92.01	88.52	85.47	83.13	81.65	79.6	78.53	77.52	77.06	82.61	**+24.87**
		supervised	✓	✗	n.a.	72.05	67.29	62.75	59.16	56.1	53.01	50.33	48.46	47.02	57.35	**-0.39**
		concat	✓	✗	n.a.	88.86	85.06	81.02	78.52	76.47	73.35	71.26	70.21	69.3	77.11	**+19.37**
		concat	✓	✓	vector	88.86	84.70	80.9	78.54	76.16	73.45	71.24	70.51	69.01	77.04	**+19.30**
		concat	✓	✓	scalar	88.86	84.58	80.8	78.04	75.86	73.08	70.84	69.65	67.95	76.62	**+18.88**
	SwAV	self-supervised	✓	✗	n.a.	92.43	89.29	85.78	83.76	81.9	79.83	78.23	77.03	76.49	82.74	**+25.00**
		supervised	✓	✗	n.a.	72.05	67.29	62.75	59.16	56.1	53.01	50.33	48.46	47.02	57.35	**-0.39**
		concat	✓	✗	n.a.	88.75	84.69	81.21	78.62	76.72	73.81	71.52	70.41	69.23	77.21	**+19.47**
		concat	✓	✓	vector	88.75	84.56	81.0	78.57	76.92	73.67	71.3	70.28	69.04	77.12	**+19.38**
		concat	✓	✓	scalar	88.75	84.6	80.75	78.33	76.72	73.28	71.11	69.67	68.27	76.83	**+19.09**
	Moco-v2	self-supervised	✓	✗	n.a.	89.93	87.23	83.92	81.76	81.27	80.2	78.78	78.02	77.96	82.11	**+24.37**
		supervised	✓	✗	n.a.	72.05	67.29	62.75	59.16	56.1	53.01	50.33	48.46	47.02	57.35	**-0.39**
		concat	✓	✗	n.a.	87.46	84.92	81.94	79.72	78.72	77.28	76.28	75.89	75.68	79.76	**+22.02**
		concat	✓	✓	vector	87.46	84.72	81.88	79.68	78.73	77.41	76.22	75.61	75.29	79.66	**+21.92**
		concat	✓	✓	scalar	87.46	85.03	81.9	79.66	78.73	77.23	76.03	75.64	75.23	79.65	**+21.91**
	SeLa-v2	self-supervised	✓	✗	n.a.	90.88	87.41	84.37	81.93	81.1	79.71	77.91	77.31	77.07	81.96	**+24.22**
		supervised	✓	✗	n.a.	72.05	67.29	62.75	59.16	56.1	53.01	50.33	48.46	47.02	57.35	**-0.39**
		concat	✓	✗	n.a.	87.4	83.52	79.71	76.62	75.46	73.35	71.3	70.76	69.62	76.41	**+18.67**
		concat	✓	✓	vector	87.4	83.63	79.11	76.89	75.46	73.10	70.62	69.89	68.81	76.10	**+18.36**
		concat	✓	✓	scalar	87.4	83.69	79.6	76.58	75.0	73.27	70.87	70.10	68.89	76.15	**+18.41**

Table 8. Ablation conducted on CIFAR100 for various self-supervised features, *i.e*., DeepCluster-v2 [11], SwAV [12], Moco-v2 [15], and SeLa-v2 [1]. We notice that not all self-supervised features are equally well suited for FSCIL. The Gaussian Generator's synthetic data for old classes improves performance over just centroid.

Method	self-supervised model	features	feature fusion layer	Gaussian generation	variance	Acc. in each session (%) ↑									Avg. ↑	improvement over CEC
						0	1	2	3	4	5	6	7	8		
CEC	-	-	-	-	-	73.07	68.88	65.26	61.19	58.09	55.57	53.22	51.34	49.14	59.52	-
FeSSSS (Ours)	**DeepClust-v2**	self-supervised	✓	✗	n.a.	56.88	53.58	49.78	46.92	44.26	41.38	39.31	37.89	37.06	45.22	**-14.30**
		supervised	✓	✗	n.a.	73.23	66.52	62.4	58.10	54.11	51.75	49.18	47.41	45.73	56.49	**-3.03**
		concat	✓	✗	n.a.	75.35	70.84	66.64	62.65	59.22	56.02	54.22	51.09	49.38	60.60	**+1.08**
		concat	✓	✓	vector	75.35	70.92	67.41	63.41	60.22	57.55	55.63	52.88	51.05	61.60	**+2.08**
		concat	✓	✓	scalar	75.35	70.81	66.7	62.73	59.62	56.45	54.33	52.10	50.23	60.92	**+1.40**
	SwAV	self-supervised	✓	✗	n.a.	58.38	54.63	50.94	46.85	45.36	42.14	40.44	38.61	37.55	46.1	**-13.42**
		supervised	✓	✗	n.a.	73.23	66.52	62.4	58.10	54.11	51.75	49.18	47.41	45.73	56.49	**-3.03**
		concat	✓	✗	n.a.	75.36	70.63	67.1	62.49	59.5	56.16	53.75	50.95	48.97	60.54	**+1.02**
		concat	✓	✓	vector	75.36	70.95	67.1	63.28	60.18	57.45	55.13	52.67	51.29	61.49	**+1.97**
		concat	✓	✓	scalar	75.36	70.81	66.94	62.70	59.86	56.37	54.0	51.44	49.72	60.8	**+1.28**
	Moco-v2	self-supervised	✓	✗	n.a.	46.83	42.52	38.98	36.41	33.86	31.14	28.94	28.18	26.66	34.83	**-24.69**
		supervised	✓	✗	n.a.	73.23	66.52	62.4	58.10	54.11	51.75	49.18	47.41	45.73	56.49	**-3.03**
		concat	✓	✗	n.a.	74.16	69.21	64.77	60.45	57.73	54.28	52.08	49.47	47.44	58.84	**-0.68**
		concat	✓	✓	vector	74.16	70.03	66.11	62.28	59.35	56.27	54.14	51.76	50.33	60.49	**+0.97**
		concat	✓	✓	scalar	74.16	69.36	65.14	61.06	58.18	54.97	52.96	50.6	48.63	59.45	**-0.07**
	SeLa-v2	self-supervised	✓	✗	n.a.	53.96	51.27	47.78	44.36	42.17	38.98	37.24	35.76	34.76	42.92	**-16.60**
		supervised	✓	✗	n.a.	73.23	66.52	62.4	58.10	54.11	51.75	49.18	47.41	45.73	56.49	**-3.03**
		concat	✓	✗	n.a.	75.28	69.86	65.78	61.26	58.48	55.50	53.11	50.21	49.35	59.87	**+0.35**
		concat	✓	✓	vector	75.28	70.52	67.12	62.82	60.27	57.28	55.0	52.70	50.78	61.30	**+1.78**
		concat	✓	✓	scalar	75.28	70.32	66.31	62.01	59.36	56.24	53.58	51.11	49.8	60.44	**+0.92**