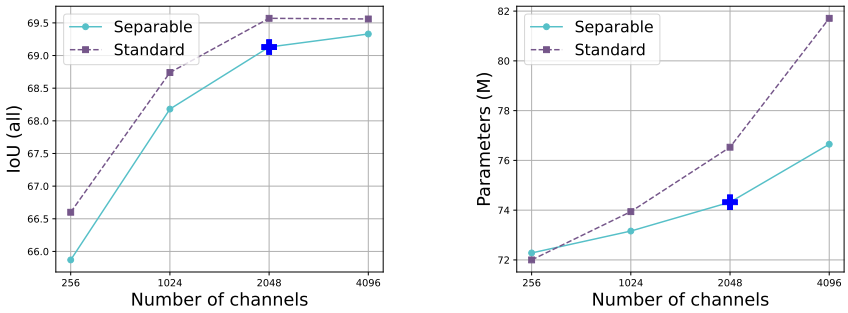

Figure 1: **IoU of our DCSS model with a different number of channels and standard or separable convolution.** (a) shows the achieved IoU while (b) compares models' size. The Blue cross shows the chosen setting for DCSS.

A Ablation study

Head size We tested a range of configurations to choose an optimal size and configuration for the HEAD. In particular, DCSS with a HEAD with 256, 1024, 2048 or 4096 channels, with or without the depthwise separable convolution. Figure A shows a plotted graph of the results. The larger number of channels increases the performance, with the gains diminishing with more than 2048 channels. HEAD with a separable convolution achieves worse results across all sizes, conforming that we lose some performance when compressing convolution blocks. The model parameter count in Figure A tells the other side of the story. Increasing the number of channels increases the size exponentially, undermining the use of a very large HEAD. However, we can recuperate some of the performance with the separable convolution. DCSS with 2048 channels and separable convolution, marked with a blue cross in the figures, achieves the best ratio of performance and efficiency and thus was used for all remaining experiments.

Dropout probability We have also experimented with the Dropout probability during the continual steps. Since regularisation helps us learn better and more reliable feature representation, its importance is lessened with frozen parameters. Results in Table 1 prove that we can achieve better results with no Dropout at steps $t > 1$. By training just one final classifier layer at t, we are essentially learning the mapping of features to final probabilities, which in our understanding, does not benefit from Dropout. Therefore, all results reported for DCSS use DROPOUT1D with $p = 0.3$ at step $t = 1$ and probability $p = 0$ in continual steps $t > 1$.

Table 1: **Comparison of the Dropout strategies.** (left) Removing Dropout for continual steps improves the performance of new tasks. (right) ScheduledDropout helps to learn better features while eventually still offering the benefits of Dropout.

Dropout p at step t		VOC 15-1 (6 tasks)				VOC 15-1 (6 tasks)		
$t = 1$	$t > 1$	0-15	16-20	all	ScheduledDropout	0-15	16-20	all
0.3	0.3	76.90	40.92	68.31	✓	**77.66**	**42.69**	**69.33**
0.3	0.1	77.14	41.45	68.65	✗	77.54	39.60	68.50
0.3	0.0	**77.66**	**42.69**	**69.33**	-	-	-	-

Scheduled Dropout Although Dropout maintains more features for online learning, the increased entropy of the output features can cause problems with proper learning. Therefore, we have tested the use of ScheduledDropout [4], where the probability increases linearly from 0 to p during training. Table 1 proves that ScheduledDropout improves the IoU by almost 1%, despite using the simplest, linear scheduling of the Dropout. Therefore we conclude that the optimal introduction of the regularisation can have a decisive effect on its success.

Table 2: **Ablation study for the weight decay and weight transfer.** (left) Increasing the weight decay decreases performance for new classes. (right) Impact of the weight transfer for the new classifier's parameters at step t.

		VOC 15-1 (6 tasks)				VOC 15-1 (6 tasks)		
Decay	λ	0-15	16-20	all	Weight transfer	0-15	16-20	all
ℓ_2	0.0001	**77.66**	**42.69**	**69.33**	*Random* $\rightarrow \phi_c^t$	**77.66**	**42.69**	**69.33**
ℓ_2	0.0005	77.43	38.74	68.21	$\phi_{c_u}^{t-1} \rightarrow \phi_c^t$	77.58	40.89	68.84
ℓ_2	0.001	75.72	33.66	65.70	-	-	-	-

Weight decay DCSS uses a standard SGD optimiser with a momentum of 0.9 and ℓ_2 weight decay of 0.0001, as in DeepLabV3 [2]. Inspired by Dropout's promising regularisation results, we tested an increase in the weight decay λ parameter. Results in Table 2 show that any increase from the original value of 0.0001 used in most semantic segmentation models decreases performance, especially in the ability to learn new classes, while old classes learned in the offline step were mainly unaffected. Therefore, we must wisely choose the regularisation technique to achieve a sparse and feature-rich representation.

Weight transfer SSUL relies heavily on the weight transfer from the unknown class ${c_b}^{t-1}$ to each of current classes $\mathcal{C}^t$. Weight initialisation to the foreground predictor's weights assumes that the classifier is more or less ready from the get-go to recognise new classes. In the context of the reduced set of trainable parameters in DCSS, we found that weight transfer is unnecessary and might prevent the model from finding an optimal solution. Table 2 shows that random initialisation outperforms weight transfer from both the background class and the unknown class.

B Additional experimental results

We have also carried additional experiments as introduced by Cha et al. [1], shown in Table 3. DCSS performs slightly worse in **2-1** and **2-2** scenario. The shared HEAD module of DCSS means that we only have a single 1×1 trainable vector per class. In these extreme scenarios, the frozen encoder does not contain enough information and can benefit from the additional trainable layer of SSUL. We conclude that the number of initial classes used for offline training plays a crucial role in further offline training.

Table 3: **IoU results on the more extreme scenarios with low initial number of classes**. DCSS struggles if the frozen model was trained only on a few classes.

	VOC 10-1 (11 tasks)			VOC 5-1 (16 tasks)			VOC 2-1 (19 tasks)			VOC 2-2 (10 tasks)		
Method	0-10	11-20	all	0-5	6-20	all	0-2	3-20	all	0-2	3-20	all
PLOP [3]	44.03	15.51	30.45	0.12	9.00	6.46	0.01	5.22	4.47	24.05	11.92	13.66
SSUL [1]	71.31	45.98	59.25	69.32	40.38	48.65	**62.35**	**34.32**	**38.32**	**62.38**	**42.46**	**45.31**
DCSS (ours)	**73.34**	**50.20**	**62.32**	**70.22**	**40.59**	**49.05**	61.06	32.60	36.67	56.94	41.36	43.59

B.1 Per-class performance

Table 4 shows the summarized results of DCSS model on the PASCAL VOC dataset by each class.

Table 4: **IoU performance per task per class** of our DCSS model on the PASCAL VOC dataset.

	bg	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	all
10-1 (11 tasks)	88.76	83.91	38.36	88.48	65.01	79.16	87.63	88.08	85.94	33.34	68.05	29.83	70.31	46.79	73.45	79.43	27.85	53.63	24.66	47.80	48.22	62.32
15-1 (6 tasks)	90.54	88.97	37.15	89.05	70.35	81.02	86.78	88.38	94.17	35.71	80.14	56.00	89.86	84.27	84.66	85.41	30.98	58.78	25.15	55.97	42.59	69.33
5-3 (6 tasks)	87.75	76.87	32.98	83.85	54.33	72.79	53.00	71.94	73.18	10.94	49.95	24.15	65.78	47.90	67.00	77.42	24.84	54.23	21.15	48.31	42.76	54.34
19-1 (2 tasks)	92.63	89.36	39.67	89.05	73.74	80.75	92.23	87.17	92.01	40.42	84.27	57.49	90.49	83.64	85.65	84.82	58.85	83.04	51.11	87.99	36.85	75.30
15-5 (2 tasks)	91.01	86.45	39.14	88.22	68.61	79.07	93.03	86.97	92.31	34.85	79.66	57.84	89.49	83.01	85.46	84.81	35.14	64.54	30.73	74.29	52.74	71.30

References

[1] Sungmin Cha, Beomyoung Kim, Youngjoon Yoo, and Taesup Moon. SSUL: Semantic Segmentation with Unknown Label for Exemplar-based Class-Incremental Learning, 2021. arXiv:2106.11562.

[2] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking Atrous Convolution for Semantic Image Segmentation, 2017. arXiv:1706.05587.

[3] Arthur Douillard, Yifu Chen, Arnaud Dapogny, and Matthieu Cord. PLOP: Learning without Forgetting for Continual Semantic Segmentation, 2021. arXiv:2011.11390.

[4] Thomas Spilsbury and Paavo Camps. Don't ignore Dropout in Fully Convolutional Networks, 2019. arXiv:1908.09162.