Supplementary Material - CLR: Channel-wise Lightweight Reprogramming for Continual Learning

Yunhao Ge[1†], Yuecheng Li[1*], Shuo Ni[1*], Jiaping Zhao[2] Ming-Hsuan Yang[2], Laurent Itti[1†]
[1]University of Southern California [2]Google Research
[*]Equal contribution as second author, [†]correspondence to {yunhaoge, itti}@usc.edu

1. Details of our 53-dataset for continual learning and method performance

Fig. 1 shows a summary of the 53 datasets we used as the continual learning benchmark in our main paper. The figure also shows the detailed per-task accuracy of our methods and baselines after learning all 53 tasks in the task incremental continual learning setting.

2. Channel-wise linear reprogramming ability

To further understand the performance of channel-wise lightweight reprogramming achieved by channel-wise linear transformation, we conduct qualitative experiments to explore the ability of CLR layer to transfer the feature map from a Pre-trained immutable parameter weight set (starting point) to a target parameter weight set (goal).

Usually, the Pre-trained weight is not good enough due to the domain gap between the Pre-trained dataset/learning paradigm and the target dataset. And a relatively good performance could be achieved by either finetuning the whole backbone on the target dataset (FINETUNE) or learning from scratch (randomly initialized backbone) on the target task dataset (SCRATCH). We will show that with the help of a very cheap CLR layer, a feature map after a pretrained (non-optimal) model could be reprogrammed towards a "relatively optimal" feature map obtained by either finetuning the whole backbone (FINETUNE) or training from scratch (SCRATCH).

We choose two datasets: CLEVR dataset and the Kannada-MNIST dataset. Model performance on the CLEVR dataset reaches 46.09% with a Pre-trained ResNet-50 backbone + linear head, 97.66% with FINETUNE, and 91.41% with SCRATCH. In this scenario, pretrain has a large accuracy gap with FINETUNE and SCRATCH. It would be interesting to see if the CLR layer could reprogram a feature map obtained from pretrain towards a feature map obtained by FINETUNE and SCRATCH, which shows the ability of CLR layer to fill a large domain gap.

Model performance on the Kannada-MNIST dataset reaches 95.77% with a Pre-trained backbone + linear head, 99.62% with FINETUNE, and 100% with SCRATCH. Here, SCRATCH performs better than FINETUNE, which shows that the pretrained weights may have no benefit (or even harmful) for target task learning. Here we want to show that the CLR layer could reprogram a feature map obtained from pretrain towards the feature map obtained by SCRATCH. We use the feature map after the first convolutional layer in the different models (pretrain, FINETUNE, and SCRATCH). Taking the feature map after the pretrain model as input and the feature map after FINETUNE (or SCRATCH) as output, we utilize a CLR layer (3x3 2D depthwise convolutional kernels) to learn the mapping, i.e. the channel-wise linear transformation between them. The qualitative results are shown in Fig. 2. Specifically, in Fig. 2, we visualize the feature map that initially has a large initial gap between pretrain and FINETUNE (or SCRATCH). The results show that after channel-wise linear transformation, the feature after pretrain could be reprogrammed towards the goal feature (FINETUNE or SCRATCH)

3. Bootstrapping results

Fig.5 in the main paper shows the average accuracy as more tasks are learned. However, the gradient of the curve is also influenced by the order of the tasks (i.e., Hard tasks located in earlier sequence will cause average accuracy tends to increase, while easy tasks located in earlier sequence will cause average accuracy tends to decrease) which is entangled with the effect of catastrophic forgetting.

We use bootstrapping to show the tendency of average accuracy when more tasks are learned. Specifically, for any number of tasks ($t \in (1, ..., 53)$) that we want to conduct in one continual learning setting, we randomly sample t tasks from the 53 tasks 50,000 times with replacement and compute the mean accuracy (mean) and standard deviation (std). Fig. 3 shows the Bootstrapping statistic results, which show the change of mean and std when we increase the total number of tasks. The X-axis represents the task number t we want to conduct. For instance, if the continual learning task

Task ID	Task Name	Num Classes	Train Images	Test Imges	Ours	CI Baseline											Different Initialized Weight		
					CLR	CCLL	EFT	ER	EWC	LWF	ONLINE-EWC	PSP	SGD	SGD-LL	SI	SUPSUP	SWAV	DINO	Pascal_VOC
0	scenes	67	5092	536	69.22%	61.75%	36.19%	25.00%	1.12%	0.93%	2.43%	1.49%	1.31%	0.93%	0.83%	16.42%	76.68%	76.68%	65.49%
1	birds	200	5200	600	69.50%	62.17%	28.83%	27.50%	0.50%	0.33%	0.83%	0.50%	0.67%	0.17%	0.22%	8.33%	62.50%	67.00%	63.17%
2	wikiart	27	2531	243	47.33%	47.33%	28.81%	13.99%	2.88%	2.06%	2.47%	5.76%	2.47%	4.12%	1.94%	20.16%	56.79%	56.79%	45.27%
3	DescribableTextures	47	2538	235	64.68%	60.43%	36.60%	30.64%	3.40%	1.70%	2.98%	2.55%	0.85%	2.55%	1.61%	20.00%	76.60%	76.60%	64.68%
4	OfficeHome_Clipart	65	3307	408	85.54%	79.41%	58.09%	41.18%	1.47%	0.98%	1.47%	1.23%	1.96%	0.74%	0.86%	45.83%	84.31%	84.07%	80.39%
5	OfficeHome_Product	65	3361	421	92.16%	85.75%	61.28%	47.27%	1.66%	0.95%	1.66%	1.90%	0.95%	1.90%	0.85%	42.04%	90.02%	91.45%	87.89%
6	OfficeHome_Art	65	1800	250	74.80%	60.40%	13.60%	10.80%	1.20%	0.80%	0.80%	1.20%	2.40%	1.60%	0.80%	12.80%	68.80%	70.40%	64.40%
7	Food-101	101	5151	505	55.45%	45.15%	15.05%	17.62%	1.19%	0.59%	0.79%	1.19%	1.19%	0.99%	0.49%	7.33%	60.99%	62.18%	49.31%
8	EuroSAT	10	2560	260	97.31%	96.15%	92.31%	75.77%	10.38%	9.62%	6.92%	30.77%	14.23%	10.38%	9.51%	75.77%	97.31%	96.92%	96.92%
9	HistAerial	7	2562	259	88.42%	84.17%	82.24%	58.69%	6.56%	10.42%	9.65%	16.60%	13.90%	12.36%	6.81%	67.95%	86.49%	90.35%	82.63%
10	OriSet_classification	3	2559	255	87.06%	81.96%	73.33%	21.57%	19.22%	6.67%	10.20%	37.65%	26.67%	32.55%	6.57%	66.67%	88.24%	90.59%	80.39%
11	Rice_Image_Dataset	5	2560	255	100.00%	100.00%	100.00%	83.53%	9.02%	21.18%	18.82%	21.18%	12.94%	10.20%	16.60%	92.55%	100.00%	100.00%	100.00%
12	garbage_classification	12	2556	252	93.65%	89.68%	76.98%	18.65%	7.54%	8.73%	7.14%	17.06%	11.90%	9.92%	8.63%	56.75%	93.65%	94.84%	94.05%
13	PokemonData	150	4868	450	92.67%	89.78%	70.44%	68.00%	0.67%	0.89%	0.22%	5.33%	0.67%	1.33%	0.79%	41.11%	93.33%	95.11%	92.67%
14	Manga_Facial_Expressions	7	357	49	77.55%	57.14%	22.45%	42.86%	14.29%	6.12%	6.12%	20.41%	12.24%	14.29%	6.02%	32.65%	65.31%	73.47%	61.22%
15	oregon_wildlife	20	2560	260	83.85%	77.69%	43.08%	25.00%	4.62%	3.85%	5.38%	10.00%	5.77%	5.77%	3.75%	24.23%	81.92%	85.00%	78.08%
16	Blood_Cell_Dataset	4	2560	256	99.61%	95.70%	98.44%	44.26%	3.28%	0.00%	11.48%	36.07%	28.52%	13.11%	0.00%	62.30%	98.44%	99.61%	98.83%
17	OCT2017	4	2560	256	94.53%	92.97%	90.63%	60.70%	6.23%	2.72%	4.28%	10.12%	27.34%	5.45%	2.62%	75.88%	92.97%	92.58%	92.97%
18	Cataract_Dataset	4	479	61	73.77%	67.21%	63.93%	48.07%	16.74%	5.15%	9.44%	23.61%	49.18%	7.30%	5.04%	46.35%	73.77%	65.57%	68.85%
19	Apparel_Images_Dataset	24	2482	257	96.89%	94.55%	89.11%	36.08%	12.55%	5.10%	9.02%	23.92%	7.39%	21.57%	4.99%	40.39%	97.67%	97.28%	96.89%
20	Zalando_Fashion_Dataset	6	2186	233	86.70%	81.97%	71.24%	43.13%	15.00%	9.38%	14.37%	25.00%	20.17%	6.88%	9.28%	68.13%	87.12%	85.84%	83.26%
21	House_Room_Images	5	2527	255	88.24%	81.57%	65.88%	21.50%	1.04%	1.04%	1.46%	1.67%	20.00%	0.84%	0.96%	13.99%	87.06%	89.02%	86.27%
22	UIUC_Sports_Event_Dataset	8	1258	160	98.75%	96.88%	88.13%	100.00%	4.62%	1.54%	4.62%	55.38%	16.25%	4.62%	1.44%	98.46%	98.75%	99.37%	98.12%
23	Yoga-82	80	5035	479	75.78%	65.97%	45.72%	19.92%	2.39%	1.20%	1.20%	4.38%	1.67%	3.98%	1.09%	21.51%	77.04%	75.37%	73.49%
24	UMNIST	20	435	65	100.00%	100.00%	98.46%	69.62%	7.59%	15.19%	6.33%	59.49%	1.54%	34.18%	12.04%	82.28%	100.00%	100.00%	100.00%
25	electronic-components	36	2526	251	45.82%	39.04%	29.08%	26.36%	14.73%	13.18%	14.73%	29.46%	3.19%	15.50%	9.92%	42.64%	47.01%	50.60%	41.83%
26	Breast_Ultrasound	3	620	79	91.14%	84.81%	92.41%	96.88%	19.53%	0.78%	0.00%	96.09%	56.96%	48.44%	0.69%	99.22%	89.87%	91.14%	93.67%
27	boat-types-recognition	9	1054	129	89.15%	79.07%	60.47%	100.00%	22.22%	27.78%	16.67%	72.22%	20.93%	33.33%	26.02%	100.00%	85.27%	86.05%	82.95%
28	concrete-crack	2	2560	256	100.00%	99.61%	100.00%	32.03%	15.23%	22.66%	10.94%	36.33%	48.83%	24.22%	19.51%	54.69%	100.00%	100.00%	100.00%
29	Malacca_Historical_Buildings	3	126	18	100.00%	33.33%	33.33%	68.16%	13.93%	9.95%	11.94%	40.80%	33.33%	24.88%	9.85%	80.10%	100.00%	100.00%	100.00%
30	African_countries	4	2560	256	84.77%	76.17%	73.44%	38.08%	11.92%	11.54%	8.85%	14.62%	33.98%	10.00%	8.38%	43.85%	82.81%	83.59%	82.03%
31	SurgicalTools	4	1607	201	98.51%	99.00%	97.01%	10.32%	8.33%	8.73%	6.35%	12.70%	21.39%	13.49%	8.62%	23.41%	100.00%	100.00%	100.00%
32	Galaxy10	10	2560	260	77.31%	70.77%	61.15%	47.78%	2.59%	2.22%	3.33%	19.26%	7.69%	1.85%	2.12%	38.15%	71.54%	73.46%	73.08%
33	Stanford_Online_Products	12	2556	252	73.41%	67.46%	41.67%	51.76%	5.88%	7.84%	7.06%	24.71%	5.95%	7.84%	7.74%	48.24%	76.19%	77.38%	67.86%
34	NWPU-RESISC45	45	2565	270	84.81%	75.93%	57.04%	26.56%	12.11%	18.36%	11.72%	24.22%	2.22%	12.50%	15.34%	42.97%	88.52%	90.00%	82.59%
35	ilab80m-shell	15	2565	255	92.94%	81.57%	74.12%	61.57%	12.16%	7.84%	6.27%	43.92%	6.27%	17.25%	7.73%	69.80%	91.37%	94.12%	87.06%
36	CLEVR_v1.0	8	2560	256	87.11%	70.31%	62.50%	78.04%	9.02%	3.92%	3.92%	36.08%	12.50%	5.49%	3.81%	87.45%	73.83%	73.44%	63.67%
37	CelebA	5	2560	255	85.88%	80.39%	79.61%	63.85%	13.85%	8.46%	15.38%	35.38%	20.39%	10.77%	8.35%	50.77%	85.49%	83.92%	85.88%
38	Vegetable_images_Dataset	15	2565	255	99.61%	98.82%	95.29%	62.11%	6.64%	3.91%	13.67%	55.86%	9.80%	22.66%	3.82%	81.25%	100.00%	100.00%	100.00%
39	Monkey_Species	10	1007	130	99.23%	97.69%	80.77%	76.56%	0.39%	4.30%	7.03%	36.72%	9.23%	23.44%	4.18%	64.84%	97.69%	99.23%	99.23%
40	aptos2019	5	1650	182	73.63%	65.93%	61.54%	53.85%	10.99%	15.38%	6.59%	53.85%	25.27%	21.98%	11.04%	51.65%	69.78%	68.13%	66.48%
41	freiburg_groceries_dataset	25	2488	250	82.40%	72.00%	57.60%	46.00%	3.20%	4.00%	2.40%	15.20%	6.40%	4.00%	3.90%	35.60%	83.60%	88.80%	76.40%
42	Weather_Type_Dataset	4	875	112	100.00%	98.21%	96.43%	64.29%	19.64%	15.18%	18.75%	81.25%	40.18%	23.21%	12.98%	94.64%	97.32%	97.32%	99.11%
43	Simpsons_Characters_Data	42	2077	220	85.91%	72.27%	67.73%	57.27%	2.73%	3.18%	0.91%	20.00%	3.64%	3.18%	3.08%	50.00%	85.45%	85.00%	84.09%
44	Hurricane_Damage_Dataset	2	2560	256	97.27%	96.88%	97.27%	75.78%	1.17%	0.39%	0.39%	90.23%	56.25%	62.89%	0.30%	92.19%	96.48%	97.27%	97.27%
45	Kannada-MNIST	10	2560	260	99.62%	98.08%	100.00%	95.38%	7.31%	11.92%	10.00%	98.08%	12.69%	9.23%	7.38%	98.85%	99.62%	99.62%	99.62%
46	dragon-ball-super-saiyan-dataset	6	112	18	94.44%	16.67%	16.67%	55.56%	22.22%	5.56%	16.67%	61.11%	16.67%	16.67%	5.45%	50.00%	66.67%	61.11%	66.67%
47	ip02-dataset	102	5100	510	42.55%	37.45%	25.10%	20.98%	1.18%	0.59%	1.18%	9.22%	1.76%	0.59%	0.48%	16.67%	50.00%	50.00%	40.20%
48	planets-and-moons-dataset-ai-in-space	11	1307	165	100.00%	100.00%	100.00%	98.79%	9.09%	4.85%	9.09%	98.79%	9.70%	11.52%	4.74%	99.39%	100.00%	100.00%	100.00%
49	polish-craft-beer-labels	100	5060	500	100.00%	100.00%	100.00%	99.60%	0.80%	1.60%	2.00%	94.20%	2.20%	0.40%	1.50%	97.00%	100.00%	100.00%	100.00%
50	the-kvasircapsule-dataset	13	2113	219	93.61%	100.00%	100.00%	69.41%	7.76%	15.07%	7.76%	49.77%	10.50%	10.05%	11.97%	74.89%	93.15%	93.61%	90.41%
51	atari_shell	67	5092	536	100.00%	100.00%	100.00%	99.81%	3.17%	12.31%	2.99%	99.44%	43.66%	0.37%	8.12%	98.88%	100.00%	100.00%	100.00%
52	deepvp_shell	9	2556	252	88.10%	81.75%	85.71%	87.70%	82.94%	77.78%	83.73%	86.51%	82.14%	18.25%	75.28%	77.38%	83.73%	86.11%	86.90%
	Average				86.05%	78.28%	67.86%	53.13%	9.36%	8.23%	8.31%	34.91%	16.72%	12.48%	7.28%	56.69%	85.12%	85.77%	82.50%

Figure 1. Statistics of the datasets and per-task accuracy of our method and baselines after learning all 53 tasks in the continual learning setting. Ablation columns indicate our methods with different initialization weights.

number t=10, then we randomly sample 10 tasks from the 53-dataset and calculate the mean accuracy. We repeat the sampling 50000 times and get the std. Y-axis shows the mean Accuracy (solid blue line) on the sampled tasks with replacement and std as the shaded light blue range. Since in our CLR method, the order of task is not mattered (we have the same performance on a specific task no matter the sequence), this allows us to simulate what would happen if we learn a different sample of tasks given a specific task number t. We observe that the mean accuracy is stable and not influenced by t when the sample number is large. For std, when the task number t is small, the std is relatively large, and the std decreases with task number t increase. When t=53, the std becomes zero.

Fig. 4 shows the Bootstrapping statistic results with detailed max and minimum accuracy logs. The X-axis represents the number of tasks t in a specific continual learning task. Y-axis shows the mean Accuracy (solid blue line) on the sampled tasks with replacement. The shaded light blue range shows the min and max range in the given task number t among 50000 times task samples. We use the solid red line to represent our reported results in the main paper Fig.5, which filled in the shaded light blue range.

4. More experiments to explore the trade-off between parameter and performance

Several other versions of our method may include methods with higher accuracy but higher cost. Our main method - CLR (the one in the paper) adds the CLR layer after all the original convolutional kernels except for the 1×1 kernels, saving many parameters.

CLR-Full version applying the CLR layer to all convolutional kernels in the pretrained model which reaches 85.85% in accuracy and cost $1.69\times$ parameters compared to our main method (CLR).

The CLR-Reduced version adds a smaller version of

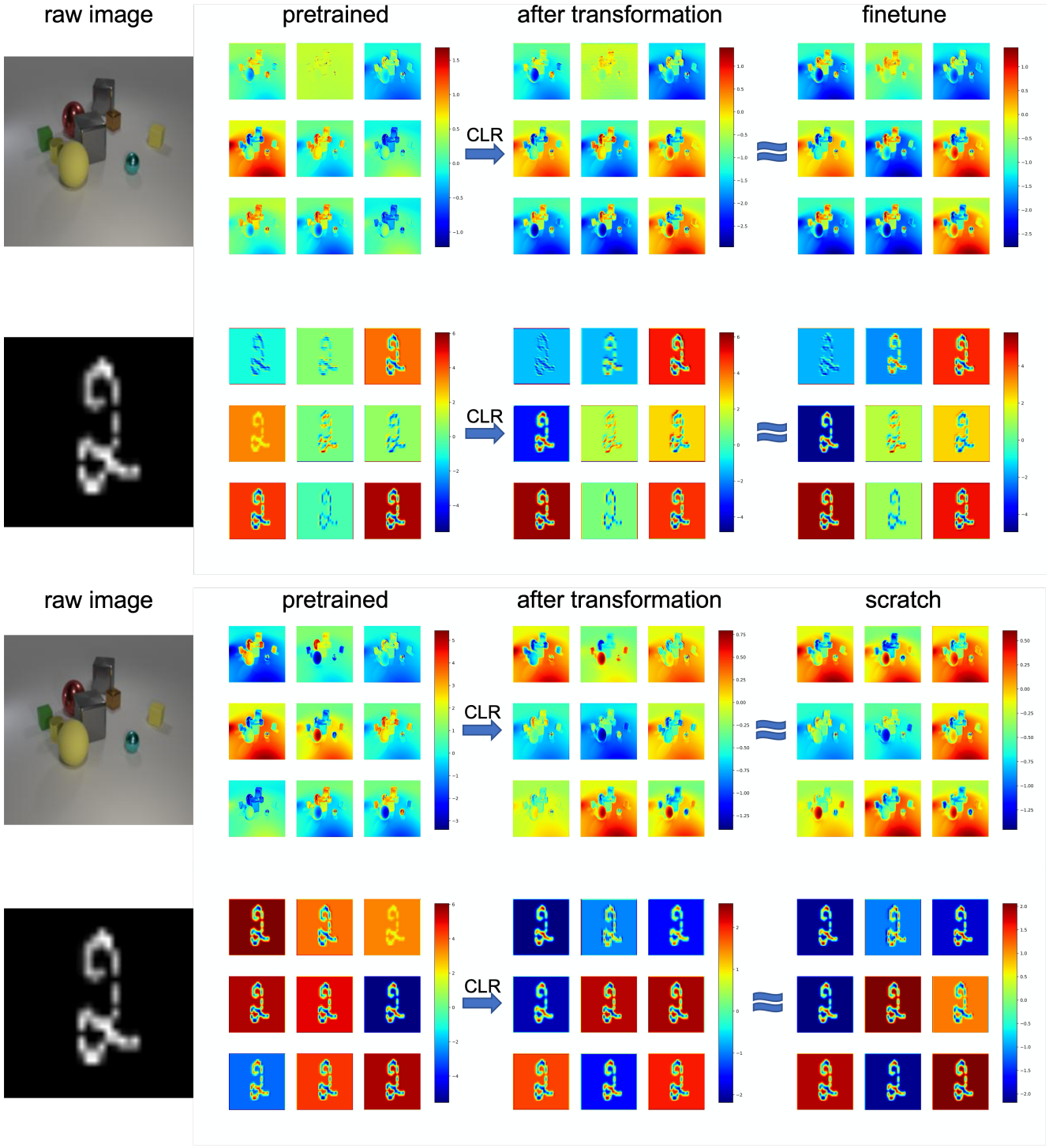

Figure 2. The Figure shows quantitative results of the CLR transformation ability on CLEVR and Kannada-MNIST datasets. We visualize the feature maps in the first residual group of ResNet-50 that initially has a large initial gap between pre-train and FINETUNE (or SCRATCH). The results show that after channel-wise linear transformation, the feature after pre-train could be reprogrammed towards the goal feature (FINETUNE or SCRATCH). Pretrained indicates the frozen Imagenet pretrained ResNet-50 backbone. Finetune is a finetuned ResNet-50 backbone with Imagenet pretrained initialization while Scratch is a trained ResNet-50 backbone with random initialization.

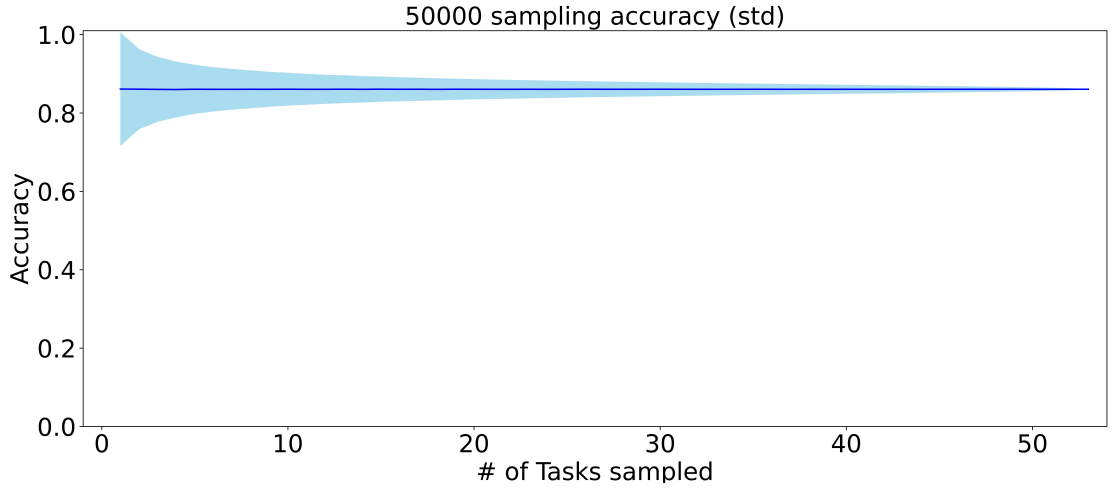

Figure 3. Bootstrapping statistic results. The X-axis represents the number of tasks t in a specific continual learning task. Y-axis shows the mean Accuracy (solid blue line) on the sampled tasks with replacement and std as the shaded light blue range.

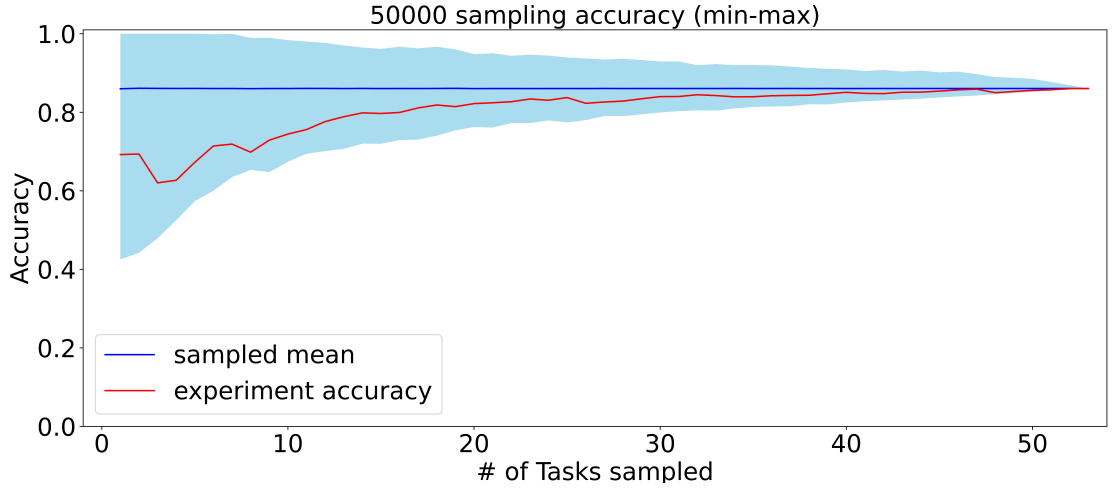

Figure 4. Bootstrapping statistic results with detailed accuracy log. The X-axis represents the number of tasks t in a specific continual learning task. Y-axis shows the mean Accuracy (solid blue line) on the sampled tasks with replacement. The shaded light blue range shows the min and max range in the given task number t. We use the solid red line to represent our reported results in the main paper Fig.5, which filled in the shaded light blue range.

CLR layer with 1×1 2D reprogramming kernels after all 1×1 original Conv kernels and normal CLR layer with 3×3 2D reprogramming kernels after the rest CONV kernels. It reaches 85.7% in accuracy and costs 1.08× parameters compared to our main method (CLR).

The CLR-mixed version learns a weighted combination of the original and our reprogrammed feature maps. The intuition is that we keep some proportion of the original feature and add the new features learned after reprogramming. Specifically, A trainable parameter **A** decides the weight of the summation of the reprogrammed feature and the original feature map. Equation 1 shows the details of the weighted combination.

$$\hat{x'}_k = A * CLR_k(x'_k) + (1 - A) * x'_k \tag{1}$$

where x'_k is the kth channel of the feature map from the original Convolutional layer and CLR_k is the corresponding linear transformation. It reaches 86.25% in accuracy and costs 1.79× parameters compared to our main method (CLR).

The results are shown in Fig. 5. Theoretically, more trainable parameters could lead to better performance, and it is true for CLR-mixed version, which achieves +0.2% than our main method. Interestingly, the CLR-full version achieves lower average accuracy than the main method,

Method	Average Acc
LwF	24%
iCARL	49%
RPS	57%
CCLL	85%
EWC	41%
SI	52%
CLR (Ours)	**94.2%**

Table 1. We applied our method on CIFAR-100 dataset with 10 tasks, each containing 10 classes with comparisons to baselines from CCLL, using ResNet-18 as the backbone.

while most of the per-task accuracy is higher (43 out of 53 tasks).

5. Transfer learning with CLR-based model

We apply our CLR method to the transfer learning problem, where we only care about the accuracy of the transferred dataset while do not need to maintain performance on previous datasets.

Datasets. We use the same 53-dataset to evaluate transfer learning performance. Specifically, we use the ImageNet pretrained ResNet-50 model as initialization and apply our method and 4 baseline transfer learning methods 53 times, on 53 different classification tasks.

Baselines. We have four baseline methods: 1) learn from scratch (SCRATCH), where the backbone ResNet-50 is randomly initialized with no prior knowledge, and then uses the training set of each task to train the whole network from scratch. 2) finetune the whole backbone and last layer (FINETUNE), 3) finetune only the last layer (LINEAR), 4) Head2Toe method [1] use the fixed backbone and need two steps: 1) feature selection: train the model by adding a large fully connection between all intermediate features and the last layer and select the important connection by adding regularization. 2) keep the important skip connection and retrain the added layers.

Fig. 6 shows the average accuracy on all 53 classification tasks and the details of each task, and Fig. 7 shows the detailed result for transfer learning. Our CLR achieves the best average accuracy on the 53-dataset compared with all baselines. Specifically, CLR achieves almost 5% average improvement on 53 datasets over Head2Toe, and even larger improvement over LINEAR, FINETUNE, and SCRATCH. This shows the effectiveness of the CLR-based model in transfer learning problems.

6. CIFAR-100 Result

We also show our method's result on incremental CIFAR-100 dataset with other previous baselines in table 1

Task ID	Task Name	Ours	Different Versions of Ours			Task ID	Task Name	Ours	Different Versions of Ours		
		CLR	CLR-Full	CLR-Mixed	CLR-Reduced			CLR	CLR-Full	CLR-Mixed	CLR-Reduced
0	scenes	69.22%	70.90%	73.88%	71.27%	27	boat-types-recognition	89.15%	89.92%	89.92%	86.82%
1	birds	69.50%	70.50%	71.00%	69.50%	28	concrete-crack	100.00%	100.00%	100.00%	100.00%
2	wikiart	47.33%	49.79%	50.21%	44.86%	29	Malacca_Historical_Buildings	100.00%	100.00%	100.00%	100.00%
3	DescribableTextures	64.68%	68.51%	71.91%	66.38%	30	African_countries	84.77%	82.42%	82.81%	83.20%
4	OfficeHome_Clipart	85.54%	86.03%	84.56%	85.78%	31	SurgicalTools	98.51%	99.00%	100.00%	98.51%
5	OfficeHome_Product	92.16%	91.45%	91.45%	91.45%	32	Galaxy10	77.31%	75.38%	78.85%	75.00%
6	OfficeHome_Art	74.80%	74.80%	74.40%	72.80%	33	Stanford_Online_Products	73.41%	75.40%	76.59%	74.21%
7	Food-101	55.45%	57.62%	60.00%	56.63%	34	NWPU-RESISC45	84.81%	84.81%	87.41%	81.85%
8	EuroSAT	97.31%	97.31%	97.69%	98.08%	35	ilab80m-shell	92.94%	93.33%	91.76%	92.55%
9	HistAerial	88.42%	88.80%	86.10%	88.42%	36	CLEVR_v1.0	87.11%	67.58%	69.53%	86.33%
10	OriSet_classification	87.06%	87.06%	88.63%	85.88%	37	CelebA	85.88%	85.88%	87.06%	87.84%
11	Rice_Image_Dataset	100.00%	100.00%	100.00%	100.00%	38	Vegetable_images_Dataset	99.61%	100.00%	99.61%	99.61%
12	garbage_classification	93.65%	95.63%	96.43%	94.05%	39	Monkey_Species	99.23%	99.23%	100.00%	99.23%
13	PokemonData	92.67%	94.44%	94.44%	93.11%	40	aptos2019	73.63%	69.78%	69.23%	71.43%
14	Manga_Facial_Expressions	77.55%	77.55%	77.55%	75.51%	41	freiburg_groceries_dataset	82.40%	84.40%	84.80%	81.60%
15	oregon_wildlife	83.85%	84.23%	81.92%	81.92%	42	Weather_Type_Dataset	100.00%	99.11%	99.11%	99.11%
16	Blood_Cell_Dataset	99.61%	99.22%	99.22%	98.83%	43	Simpsons_Characters_Data	85.91%	89.09%	89.09%	86.36%
17	OCT2017	94.53%	95.70%	94.53%	92.58%	44	Hurricane_Damage_Dataset	97.27%	97.66%	98.05%	97.66%
18	Cataract_Dataset	73.77%	75.41%	72.13%	75.41%	45	Kannada-MNIST	99.62%	100.00%	100.00%	99.62%
19	Apparel_Images_Dataset	96.89%	99.22%	98.44%	97.67%	46	dragon-ball-super-saiyan-dataset	94.44%	72.22%	83.33%	88.89%
20	Zalando_Fashion_Dataset	86.70%	87.12%	85.41%	86.70%	47	ip02-dataset	42.55%	44.71%	45.49%	42.55%
21	House_Room_Images	88.24%	89.02%	89.41%	89.41%	48	planets-and-moons-dataset-ai-in-space	100.00%	100.00%	100.00%	100.00%
22	UIUC_Sports_Event_Dataset	98.75%	98.75%	98.12%	98.12%	49	polish-craft-beer-labels	100.00%	100.00%	100.00%	100.00%
23	Yoga-82	75.78%	75.16%	77.24%	74.95%	50	the-kvasircapsule-dataset	93.61%	95.43%	94.52%	94.06%
24	UMNIST	100.00%	100.00%	100.00%	100.00%	51	atari_shell	100.00%	100.00%	100.00%	100.00%
25	electronic-components	45.82%	47.41%	48.21%	46.61%	52	deepvp_shell	88.10%	85.71%	86.11%	88.49%
26	Breast_Ultrasound	91.14%	97.47%	94.94%	91.14%		Average	86.05%	85.85%	86.25%	85.70%

Figure 5. Per-task accuracy of our main method and other versions of our method after learning all 53 tasks in the continual learning setting.

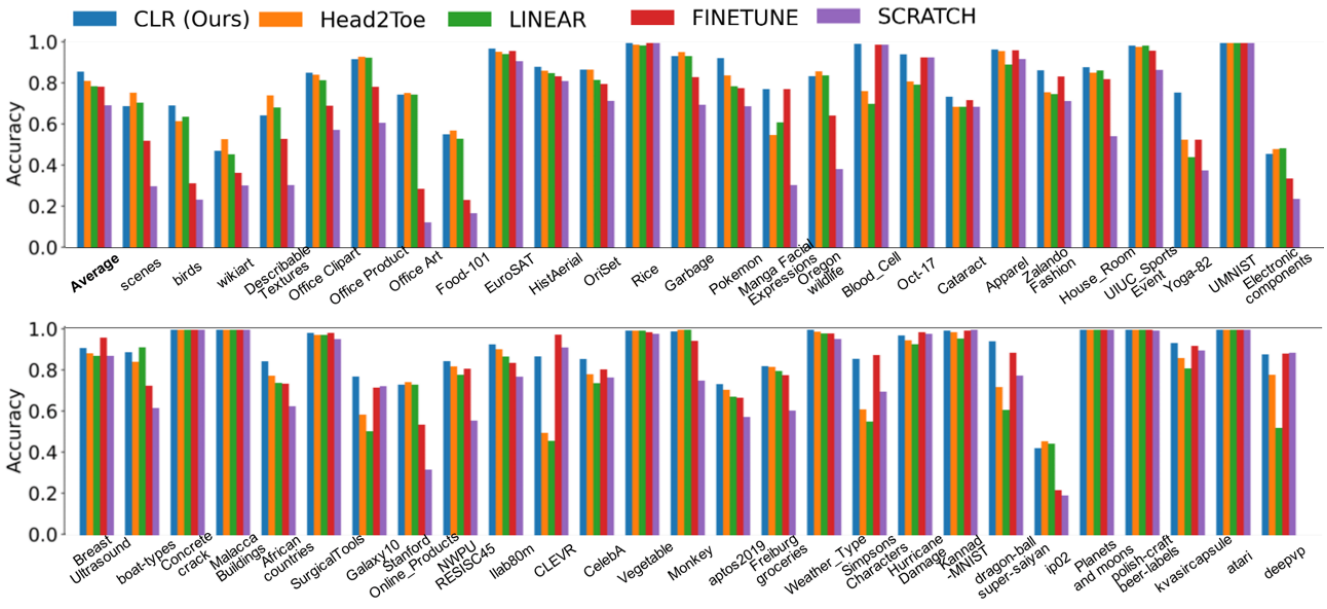

Figure 6. Bar plot of transfer learning performance on 53-dataset.

		Ours	Transfer Learning						Ours	Transfer Learning			
Task ID	Task Name	CLR	LINEAR	SCRATCH	FINETUNE	Head2Toe	Task ID	Task Name	CLR	LINEAR	SCRATCH	FINETUNE	Head2Toe
0	scenes	69.22%	70.90%	30.04%	52.24%	75.75%	27	boat-types-recognition	89.15%	91.47%	62.02%	72.87%	84.50%
1	birds	69.50%	64.00%	23.50%	31.50%	61.83%	28	concrete-crack	100.00%	100.00%	100.00%	100.00%	100.00%
2	wikiart	47.33%	45.68%	30.45%	36.63%	53.09%	29	Malacca_Historical_Buildings	100.00%	100.00%	100.00%	100.00%	100.00%
3	DescribableTextures	64.68%	68.51%	30.64%	53.19%	74.47%	30	African_countries	84.77%	74.22%	62.89%	73.83%	77.73%
4	OfficeHome_Clipart	85.54%	81.86%	57.60%	69.36%	84.56%	31	SurgicalTools	98.51%	97.51%	95.52%	98.51%	97.51%
5	OfficeHome_Product	92.16%	92.87%	61.05%	78.62%	93.35%	32	Galaxy10	77.31%	50.77%	72.69%	71.92%	58.85%
6	OfficeHome_Art	74.80%	74.80%	12.40%	28.80%	75.60%	33	Stanford_Online_Products	73.41%	73.41%	32.14%	53.97%	74.60%
7	Food-101	55.45%	53.27%	16.83%	23.37%	57.23%	34	NWPU-RESISC45	84.81%	78.15%	55.93%	81.11%	82.22%
8	EuroSAT	97.31%	94.62%	91.15%	96.15%	95.77%	35	ilab80m-shell	92.94%	87.06%	77.25%	83.92%	90.59%
9	HistAerial	88.42%	85.33%	81.47%	83.78%	86.49%	36	CLEVR_v1.0	87.11%	46.09%	91.41%	97.66%	50.00%
10	OriSet_classification	87.06%	81.96%	71.76%	80.00%	87.06%	37	CelebA	85.88%	74.12%	76.86%	80.78%	78.43%
11	Rice_Image_Dataset	100.00%	98.82%	100.00%	100.00%	99.22%	38	Vegetable_images_Dataset	99.61%	99.61%	98.04%	98.82%	99.61%
12	garbage_classification	93.65%	93.65%	69.84%	83.33%	95.63%	39	Monkey_Species	99.23%	100.00%	75.38%	94.62%	100.00%
13	PokemonData	92.67%	78.89%	69.11%	78.00%	84.22%	40	aptos2019	73.63%	67.58%	57.69%	67.03%	70.88%
14	Manga_Facial_Expressions	77.55%	61.22%	30.61%	77.55%	55.10%	41	freiburg_groceries_dataset	82.40%	80.00%	60.80%	78.00%	82.00%
15	oregon_wildlife	83.85%	84.23%	38.46%	64.62%	86.15%	42	Weather_Type_Dataset	100.00%	98.21%	95.54%	98.21%	99.11%
16	Blood_Cell_Dataset	99.61%	70.31%	99.22%	99.22%	76.56%	43	Simpsons_Characters_Data	85.91%	55.45%	70.00%	87.73%	61.36%
17	OCT2017	94.53%	79.69%	92.97%	92.97%	81.25%	44	Hurricane_Damage_Dataset	97.27%	92.97%	98.05%	98.83%	94.92%
18	Cataract_Dataset	73.77%	68.85%	68.85%	72.13%	68.85%	45	Kannada-MNIST	99.62%	95.77%	100.00%	99.62%	98.85%
19	Apparel_Images_Dataset	96.89%	89.49%	92.22%	96.50%	96.11%	46	dragon-ball-super-saiyan-dataset	94.44%	61.11%	77.78%	88.89%	72.22%
20	Zalando_Fashion_Dataset	86.70%	75.11%	71.67%	83.69%	75.97%	47	ip02-dataset	42.55%	44.71%	19.61%	22.16%	45.88%
21	House_Room_Images	88.24%	86.67%	54.51%	82.35%	85.49%	48	planets-and-moons-dataset-ai-in-space	100.00%	100.00%	100.00%	100.00%	100.00%
22	UIUC_Sports_Event_Dataset	98.75%	98.75%	86.87%	96.25%	98.13%	49	polish-craft-beer-labels	100.00%	100.00%	99.60%	100.00%	100.00%
23	Yoga-82	75.78%	44.26%	37.79%	52.82%	52.82%	50	the-kvasircapsule-dataset	93.61%	81.28%	89.95%	92.24%	86.30%
24	UMNIST	100.00%	100.00%	100.00%	100.00%	100.00%	51	atari_shell	100.00%	100.00%	100.00%	100.00%	100.00%
25	electronic-components	45.82%	48.61%	23.90%	33.86%	48.21%	52	deepvp_shell	88.10%	52.38%	88.89%	88.49%	78.17%
26	Breast_Ultrasound	91.14%	87.34%	87.34%	96.20%	88.61%		Average	86.05%	78.90%	69.59%	78.72%	81.53%

Figure 7. Transfer learning result on 53-dataset of our method and other baselines (LINEAR, SCRATCH, FINETUNE, and Head2Toe)

References

[1] Utku Evci, Vincent Dumoulin, Hugo Larochelle, and Michael C Mozer. Head2toe: Utilizing intermediate representations for better transfer learning. *arXiv preprint arXiv:2201.03529*, 2022. 5