SUPPLEMENTARY MATERIAL FOR
CPR: CLASSIFIER-PROJECTION REGULARIZATION FOR CONTINUAL LEARNING

Sungmin Cha[1], **Hsiang Hsu**[2], **Taebaek Hwang**[1], **Flavio P. Calmon**[2], **and Taesup Moon**[3*]
[1]Sungkyunkwan University [2]Harvard University [3]Seoul National University
csm9493@skku.edu, hsianghsu@g.harvard.edu, gxq9106@gmail.com,
fcalmon@g.harvard.edu, tsmoon@snu.ac.kr

1 MATHEMATICAL PROOFS

1.1 LEMMA 1 (COVER & THOMAS, 2012, THEOREM 11.6.1)

If $D_{KL}(Q\|P)$ is unbounded, then the inequality holds. Assume that $D_{KL}(Q\|P)$ is bounded, then it implies $D_{KL}(Q^*\|P) = \min_{Q\in\mathcal{Q}} D_{KL}(Q\|P)$ is also bounded. Since $\mathcal{Q}$ is a convex set, we consider a convex combination Q^θ of Q^* and Q, i.e., $Q^\theta = (1-\theta)Q^* + \theta Q \in \mathcal{Q}$, where $\theta \in [0,1]$. Since Q^* is the minimizer of $D_{KL}(Q\|P)$, we have

$$
\begin{aligned}
0 \;\leq\; & \frac{\partial}{\partial\theta} D_{KL}(Q^\theta\|P)\Big|_{\theta=0} && \text{(S.1)}\\
= \; & \frac{\partial}{\partial\theta} D_{KL}((1-\theta)Q^* + \theta Q\|P)\Big|_{\theta=0} && \text{(S.2)}\\
= \; & \frac{\partial}{\partial\theta} \int ((1-\theta)Q^* + \theta Q)\log\frac{(1-\theta)Q^*+\theta Q}{P}\Big|_{\theta=0} && \text{(S.3)}\\
= \; & \int \frac{\partial}{\partial\theta}\left[((1-\theta)Q^* + \theta Q)\log\frac{(1-\theta)Q^*+\theta Q}{P}\right]\Big|_{\theta=0} && \text{(S.4)}\\
= \; & \int \Bigg[(-Q^*+Q)\log\frac{(1-\theta)Q^*+\theta Q}{P} && \text{(S.5)}\\
& \qquad +((1-\theta)Q^*+\theta Q)\frac{P}{((1-\theta)Q^*+\theta Q)}\left(\frac{-Q^*+Q}{P}\right)\Bigg]\,|_{\theta=0}\\
= \; & \int (-Q^*+Q)\log\frac{Q^*}{P} - Q^* + Q && \text{(S.6)}\\
= \; & \int Q\log\frac{Q^*}{P} - Q^*\log\frac{Q^*}{P} && \text{(S.7)}\\
= \; & \int Q\log\frac{Q}{P} - Q\log\frac{Q^*}{Q} - Q^*\log\frac{Q^*}{P} && \text{(S.8)}\\
= \; & D_{KL}(Q\|P) - D(Q\|Q^*) - D(Q^*\|P), && \text{(S.9)}
\end{aligned}
$$

where the facts that the exchange of derivatives and integrals is guaranteed by the dominated convergence theorem and that the integrals $\int Q^* = \int Q = 1$. Therefore, we have $D_{KL}(Q\|P) \geq D(Q\|Q^*) + D(Q^*\|P)$, the desired result.

1.2 PROPOSITION 1

Note that $\mathcal{C}(P_U, \epsilon)$ is a convex set by definition since the KL divergence is convex, and hence Lemma 1 applies. By Lemma 1 and the information inequality (i.e., the KL divergence is always non-negative),

$$
D_{\mathsf{KL}}(P^{t-1*}_{Y|X}\|P^t_{Y|X}|P^{t-1}_X) \geq D_{\mathsf{KL}}(P^{t-1*}_{Y|X}\|P^{t*}_{Y|X}|P^{t-1}_X),\ \forall \mathbf{x}^1_n. \tag{S.10}
$$

*Corresponding author (E-mail: tsmoon@snu.ac.kr)

Therefore, we have

$$
\begin{aligned}
& - \mathbb{E}_{P_{Y|X}^{t-1*} P_X^{t-1}} \left[\log P_{Y|X}^{t} P_X^{t-1} \right] && \text{(S.11)} \\
= & \int P_{Y|X}^{t-1*} P_X^{t-1} \log \frac{1}{\log P_{Y|X}^{t} P_X^{t-1}} && \text{(S.12)} \\
= & \int P_{Y|X}^{t-1*} P_X^{t-1} \log \frac{P_{Y|X}^{t-1*} P_X^{t-1}}{\log P_{Y|X}^{t} P_X^{t-1}} - P_{Y|X}^{t-1*} P_X^{t-1} \log P_{Y|X}^{t-1*} P_X^{t-1} && \text{(S.13)} \\
= & D_{\mathsf{KL}}(P_{Y|X}^{t-1*} \| P_{Y|X}^{t} | P_X^{t-1}) - P_{Y|X}^{t-1*} P_X^{t-1} \log P_{Y|X}^{t-1*} P_X^{t-1} && \text{(S.14)} \\
\geq & D_{\mathsf{KL}}(P_{Y|X}^{t-1*} \| P_{Y|X}^{t*} | P_X^{t-1}) + - P_{Y|X}^{t-1*} P_X^{t-1} \log P_{Y|X}^{t-1*} P_X^{t-1} && \text{(S.15)} \\
= & - \mathbb{E}_{P_{Y|X}^{t-1*} P_X^{t-1}} \left[\log P_{Y|X}^{t*} P_X^{t-1} \right], && \text{(S.16)}
\end{aligned}
$$

where the inequality comes from (S.10).

2 Experimental details of Section 3.1

For training models on CIFAR100, CIFAR10/100 and Omniglot, we used the Adam (Kingma & Ba, 2015) optimizer with initial learning rate 0.001 for 100 epochs. For training CUB200, we set the initial learning rate as 0.0005 and trained the model for 50 epochs. Here we also used the learning rate scheduler which drops the learning rate by half when validation error is not decreased. All experiments was implemented in PyTorch 1.2.0 with CUDA 9.2 on NVIDIA 1080Ti GPU.

Following Ahn et al. (2019), we use a simple CNN model for training CL benchmark dataset except for CUB200 and details of an architecture is in Table 1 and 2.

Table 1: Network architecture for Split CIFAR-10/100 and Split CIFAR-100

Layer	Channel	Kernel	Stride	Padding	Dropout
32×32 input	3				
Conv 1	32	3×3	1	1	
Conv 2	32	3×3	1	1	
MaxPool			2	0	0.25
Conv 3	64	3×3	1	1	
Conv 4	64	3×3	1	1	
MaxPool			2	0	0.25
Conv 5	128	3×3	1	1	
Conv 6	128	3×3	1	1	
MaxPool			2	1	0.25
Dense 1	256				
Task 1 : Dense 10					
...					
Task i : Dense 10					

Table 2: Network architecture for Omniglot

Layer	Channel	Kernel	Stride	Padding	Dropout
28×28 input	1				
Conv 1	64	3×3	1	0	
Conv 2	64	3×3	1	0	
MaxPool			2	0	0
Conv 3	64	3×3	1	0	
Conv 4	64	3×3	1	0	
MaxPool			2	0	0
Task 1 : Dense C_1					
...					
Task i : Dense C_i					

3 Additional Experimental Results of Section 3.2

3.1 Experimental Results of Wide Local Minima using Test Data

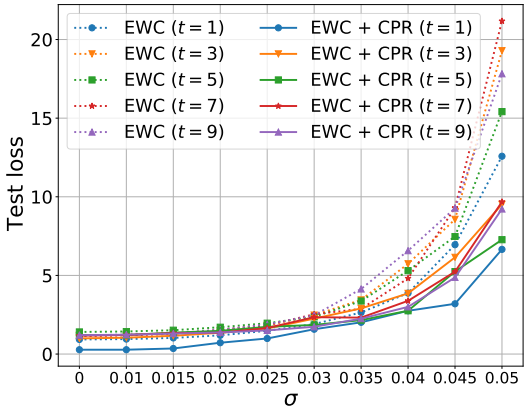
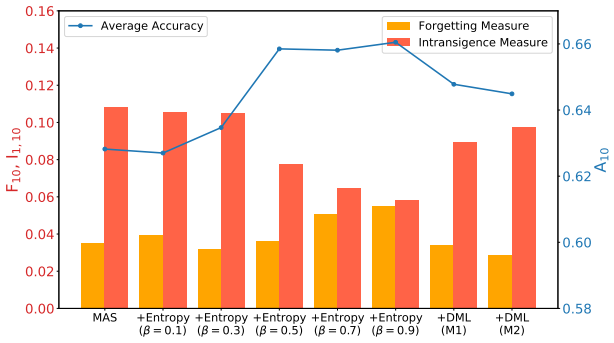


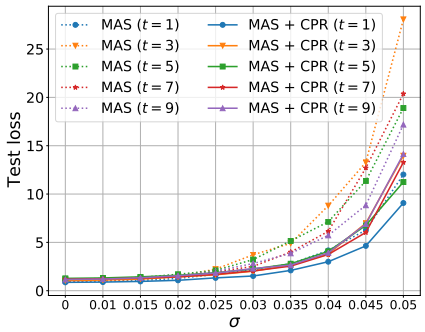
Figure 1: Experimental result of adding Gaussian noise to test data

Figure 1 shows the experimental result of Section 3.2 using test data. We clearly see that test loss of EWC + CPR slowly increases than EWC in all tasks.

3.2 Experimental Results on MAS (Aljundi et al., 2018) and Deep Mutual Learning (Zhang et al., 2018)



(a) Selecting wide-local minimum method



(b) Adding Gaussian noise to MAS + CPR

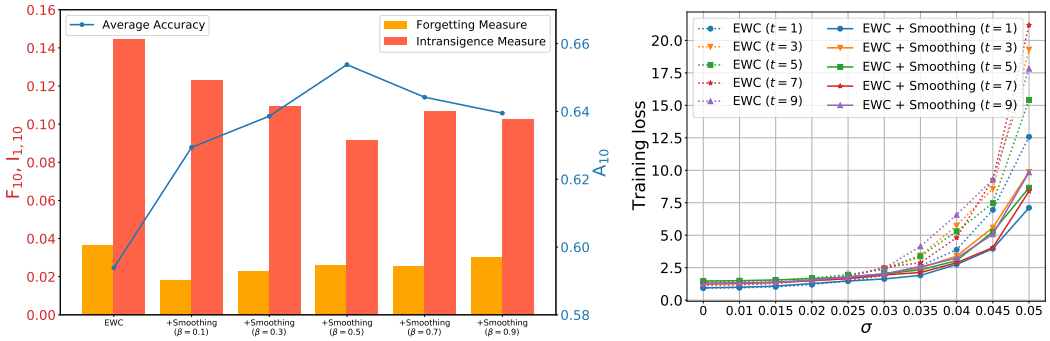
Figure 2: Experiments for selecting the regularization on CIFAR100

We did the same experiments of Section 3.2 using MAS (Aljundi et al., 2018), and Figure 2 shows the results. In Figure 2(a), we observe that MAS shows a clear trade-off between F_{10} and $I_{1,10}$ as β increases, unlike the result of EWC in the manuscript. (We note SI (Zenke et al., 2017), RWalk (Chaudhry et al., 2018) and AGS-CL (Jung et al., 2020) showed similar trend as EWC (Kirkpatrick et al., 2017) in the manuscript). MAS + CPR achieves the highest accuracy in the range of $0.5 \leq \beta \leq 0.9$ but we can see that $\beta = \{0.7, 0.9\}$ shows a worse F_{10} compared with MAS. Therefore, we can select $\beta = 0.5$ as the best hyperparameter using the criteria for selecting β proposed in Section 3.2 of the manuscript.

We also experimented Deep Mutual Learning (DML) (Zhang et al., 2018) as the regularization for converging wide local minima. We used $\beta = 1$ only because DML reports the best result (with $\beta = 1$) which is converging to a better wide local minima compared to Entropy Maximization (Pereyra et al., 2017). In our experiment, DML shows an increased A_{10} and decreased $F_{10}, I_{1,10}$ but it is not as effective as our CPR. Most decisively, DML requires training at least more than two models so we excluded DML from our consideration.

Figure 2(b) shows the experimental result on adding Gaussian noise to the parameters which is trained on CIFAR-100. We clearly observe that *test* loss of each task more slowly increases by applying CPR to MAS. We believe this is another evidence that CPR can be generally applied to regularization-based CL methods, promoting the wide-local minima.

3.3 Experimental Results on Label Smoothing Szegedy et al. (2016)

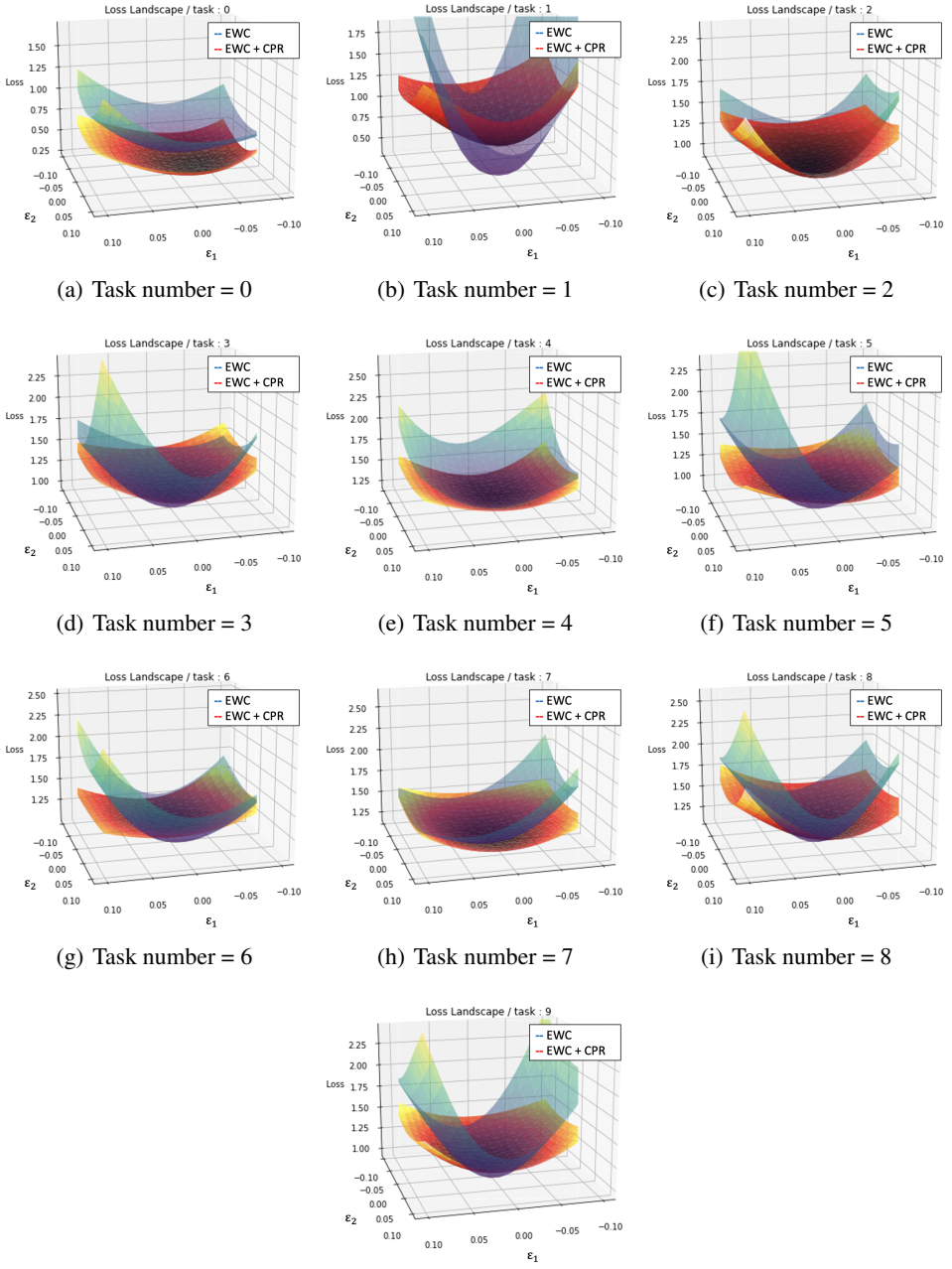


(a) Selecting regularization strength

(b) Adding Gaussian noise

Figure 3: Experimental results on Label Smoothing

Figure 3 shows the experimental results for using Label Smoothing (Szegedy et al., 2016) as regularization for the softmax output. Similar with the case of using Entropy Maximization, Figure 3(a) shows that we can find the best β ($= 0.5$) which minimizes F_{10} and $I_{1,10}$ at the same time. Also, we experimentally checked that Label Smoothing with the best β also makes a model converge to more wider local minima, as shown in Figure 3(b).



(a) Task number = 0 (b) Task number = 1 (c) Task number = 2

(d) Task number = 3 (e) Task number = 4 (f) Task number = 5

(g) Task number = 6 (h) Task number = 7 (i) Task number = 8

(j) Task number = 9

Figure 4: Visualization results of the loss landscape

3.4 Additional visualization of the loss landscape using PyHessian

In Figure 4, we visualized the loss landscape of all 10 tasks using PyHessian (Yao et al., 2019). For visualization, we used training data of each task and each single model, EWC and EWC + CPR which are finished to be trained on CIFAR-100 (10 tasks). The visualization results show that CPR really make the loss landscape of all tasks become wider than EWC only case.

4 SELECTED BEST HYPERPARAMETERS

Table 3: Best hyperparameters for each regularization-based CL method and CPR

Best λ / Best β	CIFAR100	CIFAR10/100	CIFAR50/10/50	CIFAR100/10	Omniglot	CUB200
EWC	12,000 / 0.5	25,000 / 0.4	12,000 / 0.8	20,000 / 0.6	100,000 / 1.0	300,000 / 0.4
SI	1 / 0.8	0.9 / 0.2	2 / 0.9	2 / 0.5	8 / 0.7	50 / 0.6
MAS	3 / 0.5	1 / 0.2	2 / 0.1	2 / 0.4	10 / 0.6	50 / 0.6
RWalk	8 / 0.9	4 / 0.4	10 / 0.6	10 / 0.8	3,000 / 0.6	300 / 0.9
AGS-CL	$\lambda = 400$, $\mu = 10, \rho = 0.3$	$\lambda = 7000$, $\mu = 20, \rho = 0.2$	$\lambda = 9000$, $\mu = 10, \rho = 0.3$	$\lambda = 8000$, $\mu = 10, \rho = 0.3$	$\lambda = 1000$, $\mu = 7, \rho = 0.5$	$\lambda = 2100$, $\mu = 0.5, \rho = 0.1$

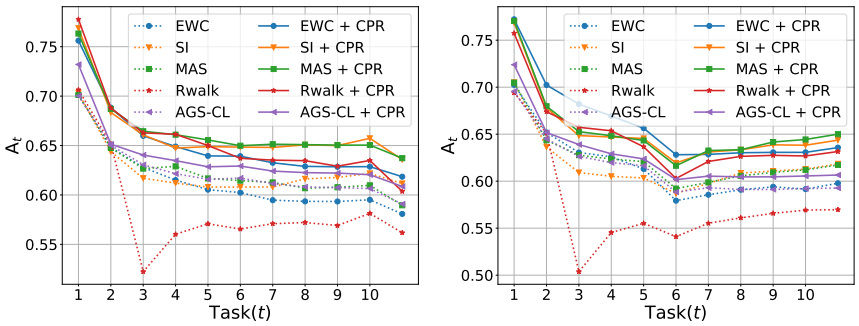
For each dataset, we firstly searched best λ for each regularization-based CL method and then we selected best β for CPR. All best hyperparameters are proposed in Figure 3.

5 EXPERIMENTAL RESULTS ON CIFAR100/10, CIFAR50/10/50

As an additional experiments of Section 3.3 in the manuscript, we experimented on CIFAR100/10 and CIFAR50/10/50, which are the different versions of CIFAR10/100. Namely, we changed the order of the tasks and varied the location for which CIFAR-10 task is inserted. Table 4 and Figure 5 show the results. We can achieve better relative improvements on all metrics compared to CIFAR-10/100.

Table 4: Experimental results on continual learning senarios with and without CPR.

Dataset	Method	Average Accuracy (A_{10})			Forgetting Measure (F_{10})			Intransigence Measure ($I_{1,10}$)		
		W/o CPR	W/ CPR	diff (W-W/o)	W/o CPR	W/ CPR	diff (W/-W/o)	W/o CPR	W/ CPR	diff (W-W/o)
CIFAR50/10/50 ($T = 11$)	EWC	0.5978	0.6346	**+0.0379 (+6.3%)**	0.0288	0.0277	**-0.0011 (-3.8%)**	0.1682	0.1313	**-0.0370 (-22.0%)**
	SI	0.6184	0.6468	**+0.0284 (+4.6%)**	0.0598	0.0532	**-0.0066 (-11.0%)**	0.1194	0.0970	**-0.0224 (-18.8%)**
	MAS	0.6172	0.6238	**+0.0066 (+1.1%)**	0.0484	0.0448	**-0.0036 (-7.4%)**	0.1310	0.1277	**-0.0033 (-2.5%)**
	Rwalk	0.5697	0.6315	**+0.0619 (+10.9%)**	0.0781	0.0548	**-0.0233 (-29.8%)**	0.1515	0.1109	**-0.0406 (-26.8%)**
	AGS-CL	0.5921	0.6055	**+0.0134 (+2.3%)**	0.0137	0.0132	**-0.0006 (-4.4%)**	0.1870	0.1736	**-0.0134 (-7.2%)**
CIFAR100/10 ($T = 11$)	EWC	0.5808	0.6158	**+0.0376 (+6.5%)**	0.0304	0.0238	**-0.0066 (-21.7%)**	0.1694	0.1378	**-0.0317 (-18.7%)**
	SI	0.6116	0.6332	**+0.0216 (+3.5%)**	0.0681	0.0692	**-0.0011 (-1.6%)**	0.1044	0.0832	**-0.0212 (-20.3%)**
	MAS	0.6138	0.6363	**+0.0214 (+3.5%)**	0.0536	0.0532	**-0.0004 (-0.7%)**	0.1153	0.0942	**-0.0211 (-18.3%)**
	Rwalk	0.5618	0.6113	**+0.0495 (+8.8%)**	0.0924	0.0852	**-0.0072 (-7.8%)**	0.1322	0.0892	**-0.0430 (-32.5%)**
	AGS-CL	0.6065	0.6205	**+0.0140 (+2.3%)**	0.0122	0.0091	**-0.0031 (-25.4%)**	0.1761	0.1618	**-0.0143 (-8.1%)**



(a) CIFAR100/10 (b) CIFAR50/10/50

Figure 5: Average accuracy for CIFAR10/100 and CIFAR50/10/50

6 EXPERIMENTS ON PERMUTED/ROTATED MNIST DATASETS

Table 5: Experimental results on Permuted/Rotated MNIST datasets

Dataset	Method	Average Accuracy (A_{10})			Forgetting Measure (F_{10})			Intransigence Measure ($I_{1,10}$)		
		W/o CPR	W/ CPR	diff (W-W/o)	W/o CPR	W/ CPR	diff (W/-W/o)	W/o CPR	W/ CPR	diff (W-W/o)
PermutedMNIST ($T = 10$)	EWC	0.7564	0.7811	**+0.0264 (+3.5%)**	0.0942	0.0893	**-0.005 (-5.3%)**	0.1353	0.1150	**-0.0202 (-14.9%)**
	MAS	0.7487	0.8239	**+0.0752 (+10.0%)**	0.1175	0.0640	**-0.0535 (-45.6%)**	0.1220	0.0949	**-0.0271 (-22.2%)**
	AGS-CL	0.7684	0.7752	**+0.0058 (+0.7%)**	0.0030	0.0009	**-0.0021 (-70.0%)**	0.2043	0.2006	**-0.0038 (-1.9%)**
RotatedMNIST ($T = 10$)	EWC	0.9852	0.9856	**+0.0002 (+0.02%)**	0.0038	0.0033	**-0.0005 (-13.2%)**	0.0075	0.0077	**+0.0003 (+4.0%)**
	MAS	0.9837	0.9839	**0.0002 (+0.02%)**	0.0040	0.0043	**+0.0003 (+7.5%)**	0.0088	0.0082	**-0.0005 (-5.9%)**
	AGS-CL	0.9702	0.9746	**+0.0044 (+0.45%)**	0	0	**0**	0.0254	0.0211	**-0.0044 (-17.3%)**

7 Feature map visualization using UMAP

We present next two-dimensional UMAP (McInnes et al., 2018) embeddings to visualize the impact of CPR on learnt representations. We compare representations produced by models trained on CIFAR-100 in two cases: (i) an oracle model which learns from the first and the t-th task at training time t, and (ii) sequential CL using EWC and EWC + CPR. We sample 30% of the test data for producing the visualization. Details and parameters for UMAP are provided in the SM.

We first visualize $O_{t,1}$, defined as the output feature map of the first output layer given the first task's test data after training the t-th task. The first row of Fig. 6 displays the respective embeddings, where c_t corresponds to the center point of the cluster for the t-th task. In the ideal case (in terms of stability), there would be little to no change in $O_{t,1}$ during CL. This is evident in the embeddings for the joint model, which show that each cluster $O_{t,1}$ is almost perfectly centered. In contrast, the resulting embedding from EWC has a slightly scattered c_t when compared to the joint (oracle) model. This indicates that, whenever the model is trained on a new task, feature maps of the output layer may drift despite EWC's regularization for previous task parameters. EWC + CPR, in turn, display more centered c_t than EWC, indicating that by applying CPR to EWC model parameters become more robust to change after training future tasks.

In order to provide further evidence that CPR provides better plasticity on new tasks, we visualized h_t, defined as the embedding for the feature map of the last hidden layers given t-th test data after training the t-th task. In the second row of Fig. 6, Joint and EWC + CPR show closer feature embeddings. EWC, in turn, has a first and second task feature maps divided from other tasks. Strikingly, the feature embeddings for the first task are completely separated. Therefore, we believe that CPR helps the model share feature representations from the start of training, potentially explaining the improvement of the intransigence measure observed in Sec 3.4 of the manuscript. We are unaware of

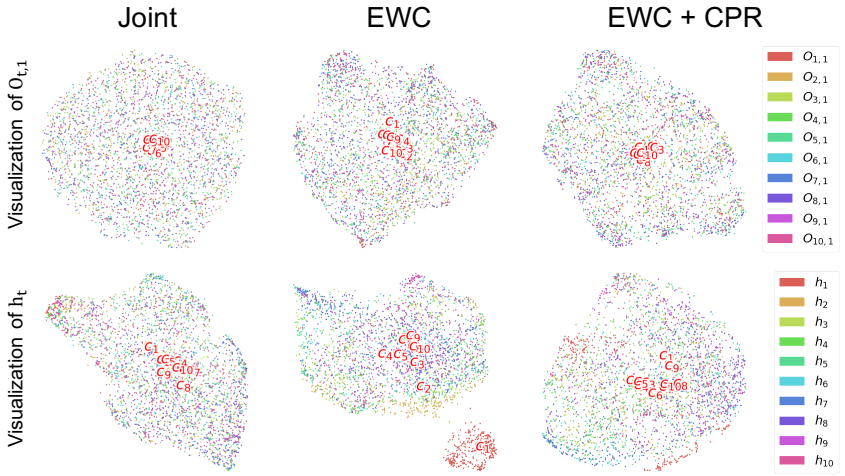


Figure 6: Feature map visualization using UMAP

prior work that makes use of feature embedding to identify reasons for catastrophic forgetting and limited plasticity of CL methods, and hope that such feature map visualizations become a useful tool for the field.

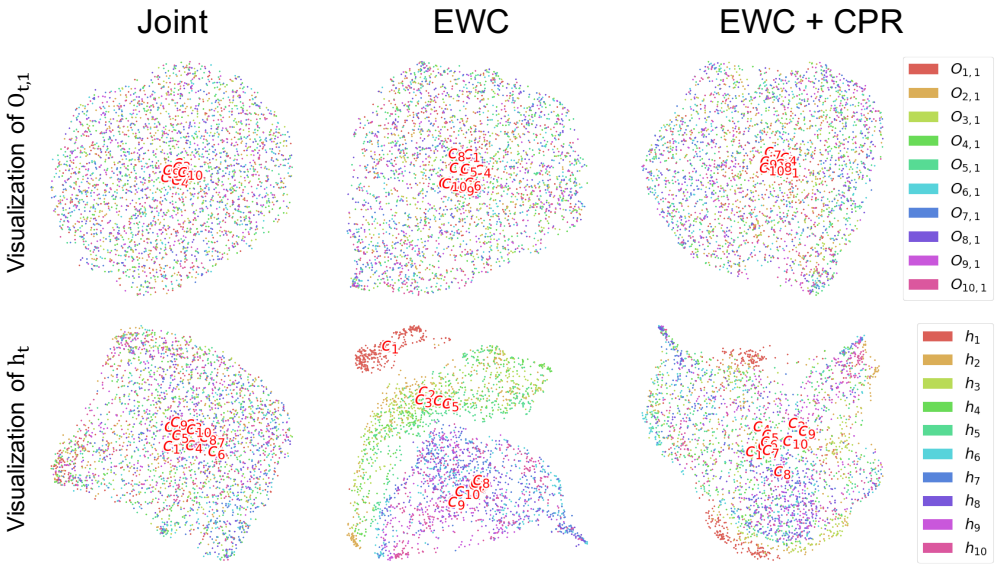
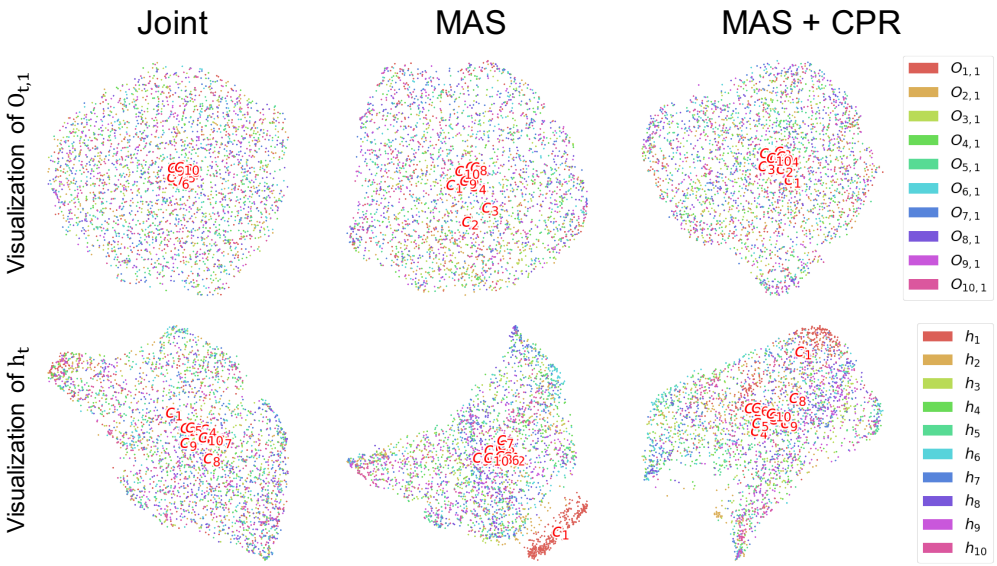


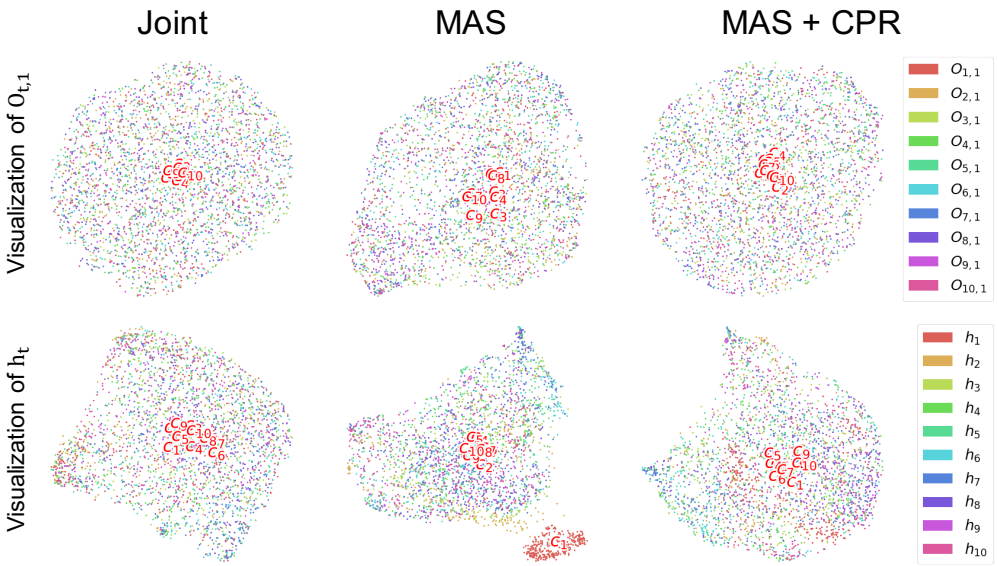
Figure 7: Visualization Result on EWC (seed = 9)

We visualize $O_{t,1}$ and h_t of Joint, EWC (Kirkpatrick et al., 2017), EWC (Kirkpatrick et al., 2017) + CPR with a different seed and visualizations are shown in Figure 7. We hold the experimental settings and we can see the similar pattern of $O_{t,1}$ and h_t, which is already shown in Section 3.5 of the manuscript. Especially, $O_{t,1}$ of EWC showed clearly divided clusters compared with the visualization result in the manuscript, nevertheless, we confirm that the feature maps become to be more shared and centered by applying CPR to EWC.

We also did same visualization using MAS (Aljundi et al., 2018) and the results are shown in Figure 8. We checked the similar results of $O_{t,1}$ and h_t, and we could see that, by applying CPR to MAS, $O_{t,1}$ and h_t are more centered than before. From these additional visualizations, we want to emphasize that the pattern of $O_{t,1}$ and h_t is a general phenomenon of regularization-based CL methods, and these can show why the typical regularization-based CL methods still suffer from the stability-plasticity dilemma at the feature map level. Also, we could check again that CPR increases the stability and plasticity of the regularizaion-based CL methods by alleviating this phenomenon.



(a) Visualization result on MAS (seed = 0)



(b) Visualization result on MAS (seed = 9)

Figure 8: Feature map visualization of MAS

7.1 Hyperapameter Settings and Visualization Details of UMAP

From several visualizations, we found out that best hyperparameters for UMAP(McInnes et al., 2018) as {n_neighbors = 200, min_dist = 0.1, n_components = 2} and we got all visualization results with these hyperparameters. We used raw features of $O_{t,1}$ as a input of UMAP, however, for visualizing h_t, we reduced the dimension of h_t to 50 by using PCA.

Table 6: Results on broader classes of algorithms (CIFAR-100, Task-IL).

Representative Algorithms		Average Accuracy (A_{10})			Forgetting Measure (F_{10})			Intransigence Measure ($I_{1,10}$)		
		W/o CPR	W/ CPR	diff (W-W/o)	W/o CPR	W/ CPR	diff (W-W/o)	W/o CPR	W/ CPR	diff (W-W/o)
Reg-based Method	UCL (Ahn et al., 2019)	0.6368	0.6387	**+0.0018 (+0.28%)**	0.0630	0.0558	**-0.0072 (-11.43%)**	0.0767	0.0813	**+0.0046 (+6.00%)**
	LWF (Li & Hoiem, 2017)	0.5209	0.5516	**+0.0306 (+5.87%)**	0.1402	0.1757	**+0.0355 (+25.32%)**	0.1231	0.0605	**-0.0626 (-50.85%)**
Parameter Isolation Method	HAT (Serra et al., 2018)	0.5534	0.5882	**+0.0348 (+6.29%)**	0.0553	0.0561	**+0.0008 (+1.45%)**	0.1670	0.1314	**-0.0356 (-21.32%)**
	PNN (Rusu et al., 2016)	0.7116	0.7193	**+0.0077 (+1.08%)**	0	0	**0**	0.0541	0.0464	**-0.0077 (-14.23%)**
Replay Method	ER (Chaudhry et al., 2019)	0.6987	0.7274	**0.0287 (+4.11%)**	0.0541	0.0410	**-0.0130 (-24.03%)**	0.0247	0.0081	**-0.0166 (-67.21%)**

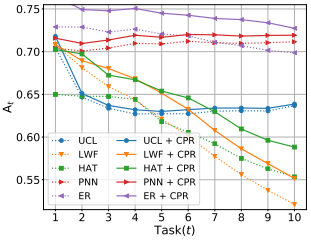


Figure 9: Accuracy across tasks

8 Applying CPR to broader classes of recent continual learning algorithms

We report the results on applying CPR to broader classes of recent continual learning algorithms in Table 6 and Figure 9 (for CIFAR-100). Following the taxonomy given in Fig.1 of the survey (De Lange et al., 2019), we took some representative methods in 3 different categories as shown in the table; UCL is a Bayesian-based, LwF is a data (distillation)-focused, HAT is a mask-based, PNN is a dynamic architecture-based, and ER (with memory size 5000) is a rehearsal-based method. From Table 6, we again observe adding CPR improves the average accuracy for all methods, and Figure 9 shows the gain is attained across all tasks for all methods. We note that adding the CPR term to the loss functions, PNN, and ER is quite natural, whereas adding it to UCL, LwF, and HAT does not clearly match with each algorithm's philosophy. Namely, it is not clear how to connect CPR with the variances of the parameters in Bayesian NN (UCL), whether the wide-local minima intuition of CPR would also hold for data-driven regularization (of LwF), and what is the notion of attaining wide-local minima for learning the attention masks (in HAT). Despite these mismatches, which may cause the glitches for F_{10} and $I_{1,10}$ (red), we believe the uniform, positive impact on the accuracy convincingly shows the high potential of CPR as a general regularizer for various continual learning methods. We believe above result significantly strengthens our method and believe our work is the first to state and verify the link between generalization and forgetting.

9 Reinforcement learning

9.1 Details on network architectures

We used the same architecture which was proposed in (Jung et al., 2020). Figure 7 shows the details of an agent model.

Table 7: Network architecture for Atari

Layer	Channel	Kernel	Stride	Padding	Dropout
84×84 input	4				
Conv 1	32×4	8×8	4	0	
ReLU					
Conv 2	32×4	4×4	2	0	
ReLU					
Conv 2	64×4	3×3	1	0	
ReLU					
Flatten					
Linear1	32×4×7×7				
Task 1 : Dense C_1					
...					
Task i : Dense C_i					

9.2 Hyperparameters of PPO and CPR

Figure 8 shows hyperparameters that we used for PPO. We evaluate each method every 40 updates, therefore, *i.e.* we have 30 evaluations during training each task. We trained the model using Adam

optimizer (lr = 0.0003) and we used default hyperparameters as in (Schulman et al., 2017). We did hyperparameter search for β of CPR, as a result, we found out that $\beta = 0.05$ shows the best result for all methods.

Table 8: Details on hyperparameters of PPO.

Hyperparameters	Value
# of steps of each task	10m
# of processes	128
# of steps per iteration	64
PPO epochs	10
entropy coefficient	0
value loss coefficient	0.5
γ for accumulated rewards	0.99
λ for GAE	0.95
mini-batch size	64

REFERENCES

Hongjoon Ahn, Sungmin Cha, Donggyu Lee, and Taesup Moon. Uncertainty-based continual learning with adaptive regularization. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 4394–4404, 2019.

Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. Memory aware synapses: Learning what (not) to forget. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 139–154, 2018.

Arslan Chaudhry, Puneet K Dokania, Thalaiyasingam Ajanthan, and Philip HS Torr. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 532–547, 2018.

Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet K Dokania, Philip HS Torr, and Marc'Aurelio Ranzato. On tiny episodic memories in continual learning. *arXiv preprint arXiv:1902.10486*, 2019.

Thomas M Cover and Joy A Thomas. *Elements of information theory*. John Wiley & Sons, 2012.

Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Aleš Leonardis, Gregory Slabaugh, and Tinne Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *arXiv preprint arXiv:1909.08383*, 2019.

Sangwon Jung, Hongjoon Ahn, Sungmin Cha, and Taesup Moon. Adaptive group sparse regularization for continual learning. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pp. 3647–3658. Curran Associates, Inc., 2020.

Diederick P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017. ISSN 0027-8424. doi: 10.1073/pnas.1611835114.

Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(12):2935–2947, 2017.

Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

Gabriel Pereyra, George Tucker, Jan Chorowski, Łukasz Kaiser, and Geoffrey Hinton. Regularizing neural networks by penalizing confident output distributions. *arXiv preprint arXiv:1701.06548*, 2017.

Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Joan Serra, Didac Suris, Marius Miron, and Alexandros Karatzoglou. Overcoming catastrophic forgetting with hard attention to the task. In *International Conference on Machine Learning (ICML)*, pp. 4548–4557. PMLR, 2018.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pp. 2818–2826, 2016.

Zhewei Yao, Amir Gholami, Kurt Keutzer, and Michael Mahoney. Pyhessian: Neural networks through the lens of the hessian. *arXiv preprint arXiv:1912.07145*, 2019.

Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *International Conference on Machine Learning (ICML)*, pp. 3987–3995, 2017.

Ying Zhang, Tao Xiang, Timothy M Hospedales, and Huchuan Lu. Deep mutual learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4320–4328, 2018.