Supplementary Material for

Online Continual Learning on a Contaminated Data Stream with Blurry Task Boundaries

Jihwan Bang[1,2] Hyunseo Koh[2,3] Seulki Park[2,4] Hwanjun Song[1,2] Jung-Woo Ha[1,2] Jonghyun Choi[2,5]
NAVER CLOVA[1] NAVER AI Lab[2] GIST[3] Seoul National Univ.[4] Yonsei Univ.[5]
{jihwan.bang,hwanjun.song,jungwoo.ha}@navercorp.com, hyunseo8157@gm.gist.ac.kr,
seulki.park@snu.ac.kr, jc@yonsei.ac.kr

Note: We use blue color to refer to figures, tables, section numbers and citations **in the main paper** (*e.g.*, [17]). All red or green characters refer to figures, tables, section numbers and citations in this supplementary material.

1. Detailed Algorithm

Algorithm 1 describes overall procedure of PuriDivER. For each task, a model is trained with online data stream S_t via SGD optimizer (Lines 3–5). Since we can see the data at once by definition of online learning, episodic memory is updated every batch. To get the diversified examples in the memory, we consider the examples of memory and a new example from batch as memory candidates, then remove one that has the maximum value of score function (Lines 7–9). After constructing memory, the model trains with memory. Since the memory might have contaminated data by selecting examples considering both diversity and purity, we split the memory as three parts every epoch; clean set $\mathcal{C}$, noisy but high model's confident set $\mathcal{R}$, noisy and low model's confident set $\mathcal{U}$ (Lines 11–13). Finally, we apply to the different loss function of each set that is appropriate for handling noisy labels (Lines 14–15).

Algorithm 1 Purity and Diversity aware Episode Replay

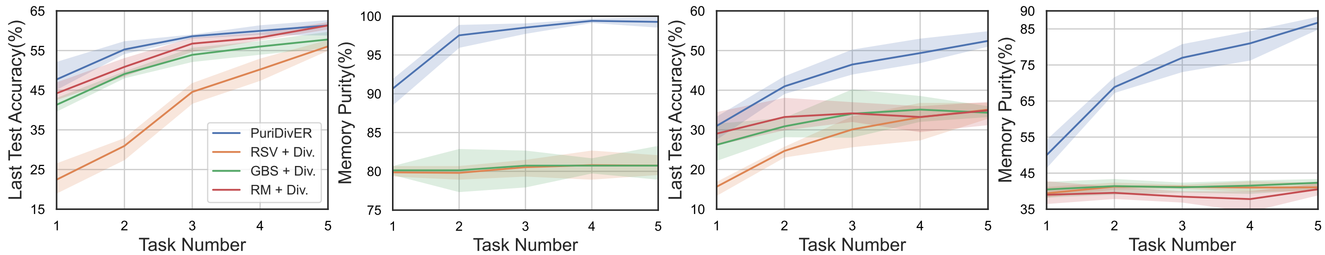
1: **Input:** $\mathcal{S}_t$: stream data at task t, $\mathcal{M}$: exemplars stored in a episodic memory, T: the number of tasks, θ_0: initial model.
2: **for** t = 1 **to** T **do**
3:     **for** each mini-batch $\mathcal{B} \in \mathcal{S}_t$ **do**
4:         /* Online Training */
5:         $\theta \leftarrow \theta - \alpha \nabla \sum_{i \in \mathcal{B}} \ell(\theta(x_i), \tilde{y_i})$
6:         /* Memory Construction */
7:         **for** $x_i, \tilde{y_i}$ in $\mathcal{B}$ **do**
8:             $\mathcal{M} \leftarrow \mathcal{M} \cup (x_i, \tilde{y_i})$
9:             $\mathcal{M}$.remove($\arg\max_{(x,\tilde{y}) \in \mathcal{M}} S(x, \tilde{y})$) by Eq. 2
10:     /* Memory Usage */
11:     **for** $e = 1$ **to** MaxEpoch **do**
12:         Split $\mathcal{C}, \mathcal{N}$ from $\mathcal{M}$ via Eq. 5
13:         Split $\mathcal{R}, \mathcal{U}$ from $\mathcal{N}$ via Eq. 7
14:         **for** each mini-batch $\mathcal{B}_\mathcal{C} \in \mathcal{C}$, $\mathcal{B}_\mathcal{R} \in \mathcal{R}$, $\mathcal{B}_\mathcal{U} \in \mathcal{U}$ **do**
15:             $\theta \leftarrow \theta - \alpha \nabla [\ell_{cls} + \eta \ell_{reg}]$

2. Detailed Configuration for Online Continual Learning

To split dataset into several tasks, we follow [5] to make *blurry-CL*. *Blurry-CL* contains two types of classes, major and minor classes. We set $L = 0.1$, so the number of minor classes are 10% of the entire of classes at each task. Since some labels might be incorrect, real class distribution of each task might be different. We configure CIFAR-10/100, WebVision and Food-101N as 5, 5, 10 and 5 tasks, respectively. Since Food-101N has 101 classes, the number of major classes is 21 at the last task. Furthermore, we consider online-CL which the incoming samples are presented to a model only once except for the examples from the memory.

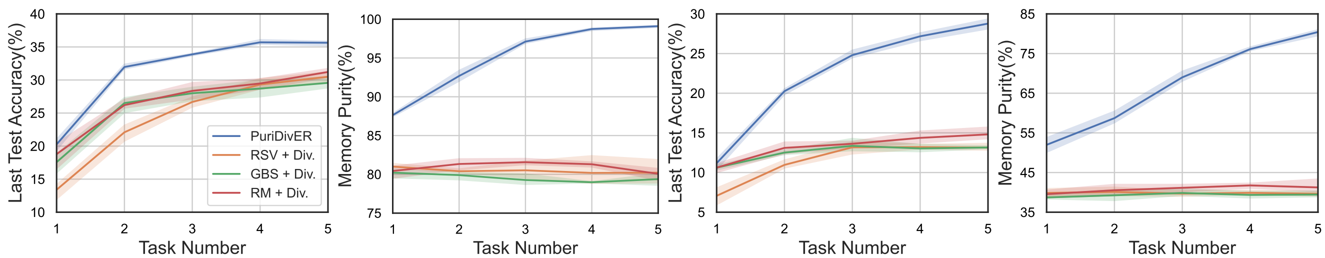
3. Accuracy and memory purity by various noise ratios

We add experimental results of the last accuracy and memory purity of PuriDivER on CIFAR-10/100 with the symmetric noise 20% and 60% in Fig. 1 and 2. We can find consistent results as in Fig. 2. On different scenarios, PuriDivER outperforms the other baselines for both accuracy and memory purity over the entire task stream. Especially, in memory purity, PuriDivER greatly outperforms the other baselines, and it shows that our method can effectively find clean labels from corrupted data for the online continual learning setup. Since we believe that memory sampling and robust learning from PuriDivER are the relationship to help each other, so the last accuracy and memory purity increases and have larger gap as the task number increases.



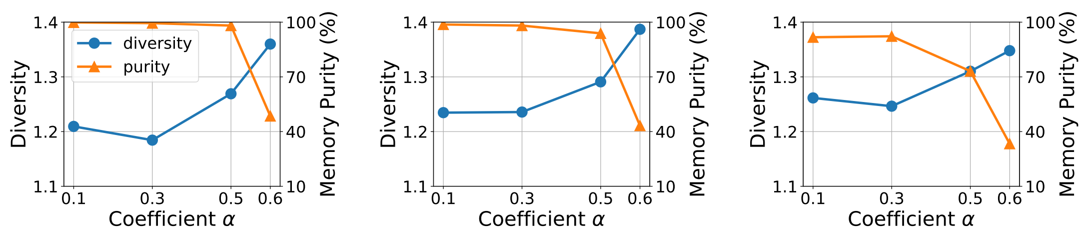
(a) CIFAR-10 with SYM-20%   (b) CIFAR-10 with SYM-60%

Figure 1. Illustration of last accuracy and memory purity changes as the task number increases on CIFAR-10 with SYM-$\{20\%, 60\%\}$.



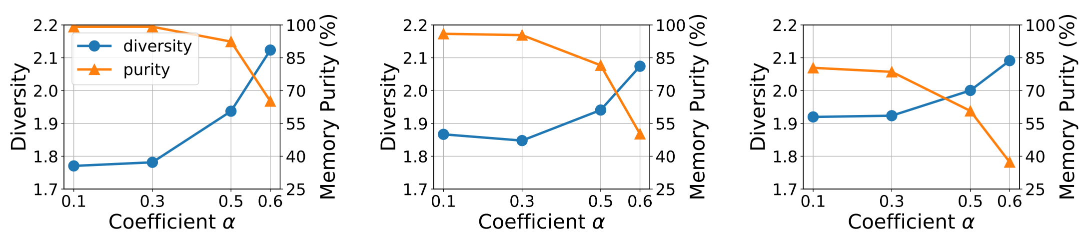
(a) CIFAR-100 with SYM-20%   (b) CIFAR-100 with SYM-60%

Figure 2. Illustration of last accuracy and memory purity changes as the task number increases on CIFAR-100 with SYM-$\{20\%, 60\%\}$.



(a)   (b)   (c)

Figure 3. Illustration of diversity and purity in the memory as the coefficient α changes on CIFAR-10 with (a) SYM-20%, (b) SYM-40%, and (c) SYM-60%.



(a)   (b)   (c)

Figure 4. Illustration of diversity and purity in the memory as the coefficient α changes on CIFAR-100 with (a) SYM-20%, (b) SYM-40%, and (c) SYM-60%.

4. Diversity vs. Purity

To analyze how diversity and purity in a memory change according to coefficient α, we plot the diversity and purity metrics (defined in Eq. 12) in Fig. 3 and 4 for various noise ratios. As hyper-parameter α increases, the diversity score increases while the purity score decreases in all figures. When α is increased beyond a certain value, it can be seen that purity is sacrificed for the sample diversity. If there are too many noisy labels in a memory, a trained model would inevitably learn the noisy labels, which can lead to performance degradation. Therefore, the best strategy is to set the α before the abrupt decrease in the purity score. For example, in Fig. 3a and 3b, memory purity is over 95% until

Table 1. Comparison of our method and SPR in disjoint setup.

Methods	CIFAR-10					WebVision
	Sym.			Asym.		
	20	40	60	20	40	Real
Self-Centered filter [17]	36.5	35.7	32.5	37.1	36.9	33.0
Self-Replay [17]	40.1	31.4	22.4	44.1	43.2	48.0
SPR [17]	43.9	43.0	40.0	44.5	43.9	40.0
PuriDivER (Ours)	**61.2**	**60.9**	**56.0**	**62.4**	**46.4**	**51.8**

$\alpha = 0.5$, and then falls to less than 50% when α is set to 0.6. Thus, the best strategy for balancing the diversity and memory purity can be obtained at $\alpha = 0.5$. Meanwhile, in the case of Fig. 3c, memory purity drops to 70% at $\alpha = 0.5$, so it can be expected that the best performance is obtained at $\alpha = 0.3$, before memory purity is too much degraded. This is because memory purity becomes more important as the ratio of noisy labels contained in memory increases. These results are consistent with the results in Tab. 4.

5. Comparison to SPR in Disjoint Tasks

As SPR [17] is proposed for similar task setup, we compare our PuriDivER to SPR [17] in the disjoint setup for which the SPR is proposed, and summarize the results in Tab. 1. Following their experimental setup, we configure five tasks with randomly paired classes for CIFAR-10, and use the top 14 largest classes in seven tasks with randomly paired classes for WebVision.

SPR uses two different memories; a delayed buffer, which the incoming data is saved temporarily, and a purified buffer which maintains purified data, and a fixed memory size of 500 or 1,000 for CIFAR-10 or WebVision, respectively. As the delayed buffer is only utilized to auxiliary model (called Self-Replay Model) to use only purified examples for model training, their actual memory size is $2\times$ of the episodic memory. Hence, we set memory size as 1,000 and 2,000 in CIFAR-10 and WebVision for fair comparison.

Interestingly, PuriDivER outperforms SPR by 3%-18% in all experiments (Tab. 1). We believe it is because the SPR always purifies incoming contaminated data stream before training the model, but it has a high risk of overfitting by false model predictions.