Contrastive Continuity on Augmentation Stability Rehearsal for Continual Self-Supervised Learning

Haoyang Cheng, Haitao Wen, Xiaoliang Zhang, Heqian Qiu *,
Lanxiao Wang, Hongliang Li *
University of Electronic Science and Technology of China, Chengdu, China
chenghaoyang@std.uestc.edu.cn, haitaowen@std.uestc.edu.cn, xlzhang@std.uestc.edu.cn,
hqqiu@uestc.edu.cn, lanxiao.wang@std.uestc.edu.cn, hlli@uestc.edu.cn

A. Proofs of the relation between C^{2}ASR and the Information Bottleneck (IB) principle

We show that $\mathcal{L}_\tau$ is an upper bound of the IB principle, as follows:

$$\mathcal{L}_\tau = -\log \frac{\exp \sum_{B_{t-1}^\tau} \log \frac{p\left(z_{t|\tau}|z_{\tau|\tau}\right)}{p\left(z_{t|\tau}\right)}}{\exp \sum_{\mathcal{D}_t} \log \frac{p\left(z_{t|t}|z_{\tau|t}\right)}{p\left(z_{t|t}\right)} + \exp \sum_{B_{t-1}^\tau} \log \frac{p\left(z_{t|\tau}|z_{\tau|\tau}\right)}{p\left(z_{t|\tau}\right)}} \tag{1}$$

$$= \log \frac{\exp \sum_{\mathcal{D}_t} \log \frac{p\left(z_{t|t}|z_{\tau|t}\right)}{p\left(z_{t|t}\right)} + \exp \sum_{B_{t-1}^\tau} \log \frac{p\left(z_{t|\tau}|z_{\tau|\tau}\right)}{p\left(z_{t|\tau}\right)}}{\exp \sum_{B_{t-1}^\tau} \log \frac{p\left(z_{t|\tau}|z_{\tau|\tau}\right)}{p\left(z_{t|\tau}\right)}} \tag{2}$$

$$= \log \left(1 + \frac{\exp \sum_{\mathcal{D}_t} \log \frac{p\left(z_{t|t}|z_{\tau|t}\right)}{p\left(z_{t|t}\right)}}{\exp \sum_{B_{t-1}^\tau} \log \frac{p\left(z_{t|\tau}|z_{\tau|\tau}\right)}{p\left(z_{t|\tau}\right)}} \right) \tag{3}$$

$$\geqslant \log \frac{\exp \sum_{\mathcal{D}_t} \log \frac{p\left(z_{t|t}|z_{\tau|t}\right)}{p\left(z_{t|t}\right)}}{\exp \sum_{B_{t-1}^\tau} \log \frac{p\left(z_{t|\tau}|z_{\tau|\tau}\right)}{p\left(z_{t|\tau}\right)}} \tag{4}$$

$$= \sum_{\mathcal{D}_t} \log \frac{p\left(z_{t|t}|z_{\tau|t}\right)}{p\left(z_{t|t}\right)} - \sum_{B_{t-1}^\tau} \log \frac{p\left(z_{t|\tau}|z_{\tau|\tau}\right)}{p\left(z_{t|\tau}\right)} \tag{5}$$

$$= \frac{\mathop{\mathbb{E}}_{\mathcal{D}_t} \left[\log \frac{p\left(z_{t|t}|z_{\tau|t}\right)}{p\left(z_{t|t}\right)} \right]}{\frac{1}{|\mathcal{D}_t|}} - \frac{\mathop{\mathbb{E}}_{B_{t-1}^\tau} \left[\log \frac{p\left(z_{t|\tau}|z_{\tau|\tau}\right)}{p\left(z_{t|\tau}\right)} \right]}{\frac{1}{|B_{t-1}^\tau|}} \tag{6}$$

$$= |\mathcal{D}_t| \, I\left(Z_{t|t}; Z_{\tau|t}\right) - \left|B_{t-1}^\tau\right| I\left(Z_{t|\tau}; Z_{\tau|\tau}\right) \tag{7}$$

$$\geqslant I\left(Z_{t|t}; Z_{\tau|t}\right) - \frac{\left|B_{t-1}^\tau\right|}{|\mathcal{D}_t|} I\left(Z_{t|\tau}; Z_{\tau|\tau}\right) \tag{8}$$

B. Ablation study

B.1. The accuracy maps across the task streams

In this part, we report the accuracy maps across the task streams on Split CIFAR-10 in Table 1. Specifically, the accuracy map includes the knn accuracies on all seen tasks after training each task, where Tr_iTe_j corresponds to the value of row i and column j in each accuracy map. Compared with FINETUNE, LUMP and CaSSLe† have alleviated catastrophic forgetting. However, LUMP suffers from the overfitting effect, whose F_1, F_2, F_3 and F_4 still reach 2.49%, 4.84%, 3.12% and 3.17% respectively. Compared with LUMP which utilizes random sampling strategy for rehearsal, the proposed ASR which samples the most representative and discriminative samples by estimating the augmentation stability for rehearsal makes greater improvements, whose F_1, F_2, F_3 and F_4 outperform LUMP by 0.04%, 0.80%, 1.80% and 0.94% respectively. In addition, the average accuracy of ASR outperforms LUMP by 0.69%. CaSSLe† acquires better anti-forgetting performance, whose average forgetting F outperforms LUMP by 0.78%. However, CaSSLe† has a phenomena that it prevents the model from learning new task streams. Specifically, the Tr_2Te_2, Tr_3Te_3, Tr_4Te_4 and Tr_5Te_5 of CaSSLe† are lower than FINETUNE by 3.06%, 2.89%, 0.05% and 1.00% respectively. Compared with CaSSLe†, the proposed C^{2}ASR preserves as much information shared among seen task streams as possible to prevent catastrophic forgetting, whose average forgetting F outperforms CaSSLe† by 0.11%, and dismisses the redundant information between previous task streams and current task stream to free up the ability to encode fresh information, whose Tr_2Te_2, Tr_3Te_3 and Tr_5Te_5 outperform CaSSLe† by 1.38%, 1.65% and 1.29% respectively. Therefore, the average accuracy of C^{2}ASR outperforms CaSSLe† by 0.94%.

B.2. The results based on existing popular self-supervised learning methods

We give the average accuracy and average forgetting of the proposed C^{2}ASR based on existing popular self-supervised learning methods on Split CIFAR-10, e.g.,

*Corresponding authors.

Table 1: The accuracy maps across the task streams on Split CIFAR-10. CaSSLe† is the improved reproduced version by incorporating with the replay strategy in LUMP [6] to make a fair comparison.

FINETUNE					LUMP [6]					CaSSLe† [4]				
1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
96.97	89.17	89.16	88.84	93.32	96.82	95.62	95.16	94.68	94.33	96.94	95.95	95.72	94.72	94.43
-	88.94	83.83	82.70	80.45	-	86.61	83.32	81.81	81.77	-	85.88	84.87	83.49	82.31
-	-	94.68	90.31	87.57	-	-	91.53	89.74	88.41	-	-	91.79	91.44	90.71
-	-	-	97.51	93.60	-	-	-	97.60	94.43	-	-	-	97.46	94.13
-	-	-	-	97.25	-	-	-	-	96.41	-	-	-	-	96.25
FINETUNE					ASR(Ours)					C^{2}ASR(Ours)				
1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
96.97	89.17	89.16	88.84	93.32	96.84	95.92	94.68	94.43	94.39	96.92	95.96	95.54	94.79	94.48
-	88.94	83.83	82.70	80.45	-	86.65	84.83	83.03	82.61	-	87.26	85.07	84.52	84.21
-	-	94.68	90.31	87.57	-	-	91.06	90.09	89.74	-	-	93.44	92.38	91.43
-	-	-	97.51	93.60	-	-	-	97.21	94.98	-	-	-	97.43	94.87
-	-	-	-	97.25	-	-	-	-	97.06	-	-	-	-	97.54

Table 2: The results (Average Accuracy and Average Forgetting) of collaboration with existing popular self-supervised learning methods on Split CIFAR-10. All methods are pre-trained with Resnet-18 as backbone for 200 epoches on Split CIFAR-10 and evaluated with KNN classifier [7]. CaSSLe† is the improved reproduced version by incorporating with the replay strategy in LUMP [6] to make a fair comparison. All the performances are measured by calculating the mean and standard deviation across three trials. The Top-2 results are highlighted in bold and underlined respectively.

	MoCo v2 [2]		BYOL [5]	
	Accuracy	Forgetting	Accuracy	Forgetting
FINETUNE	89.27(±0.51)	4.70(±0.81)	88.49(±0.52)	4.93(±0.77)
LUMP [6]	91.56(±0.25)	2.24(±0.29)	91.14(±0.48)	2.61(±0.37)
CaSSLe† [4]	<u>91.74</u>(±0.36)	<u>1.82</u>(±0.47)	<u>91.65</u>(±0.52)	<u>2.51</u>(±0.49)
C^{2}ASR(Ours)	**92.07**(±0.28)	**1.73**(±0.34)	**92.43**(±0.40)	**2.25**(±0.46)
	SimSiam [3]		BarlowTwins [8]	
	Accuracy	Forgetting	Accuracy	Forgetting
FINETUNE	90.11(±0.12)	5.42(±0.08)	87.72(±0.32)	4.08(±0.56)
LUMP [6]	91.00(±0.40)	2.92(±0.53)	90.31(±0.30)	1.13(±0.18)
CaSSLe† [4]	<u>91.51</u>(±0.38)	<u>2.77</u>(±0.54)	<u>90.97</u>(±0.35)	<u>1.09</u>(±0.41)
C^{2}ASR(Ours)	**92.47**(±0.41)	**2.59**(±0.58)	**91.34**(±0.26)	**0.94**(±0.22)

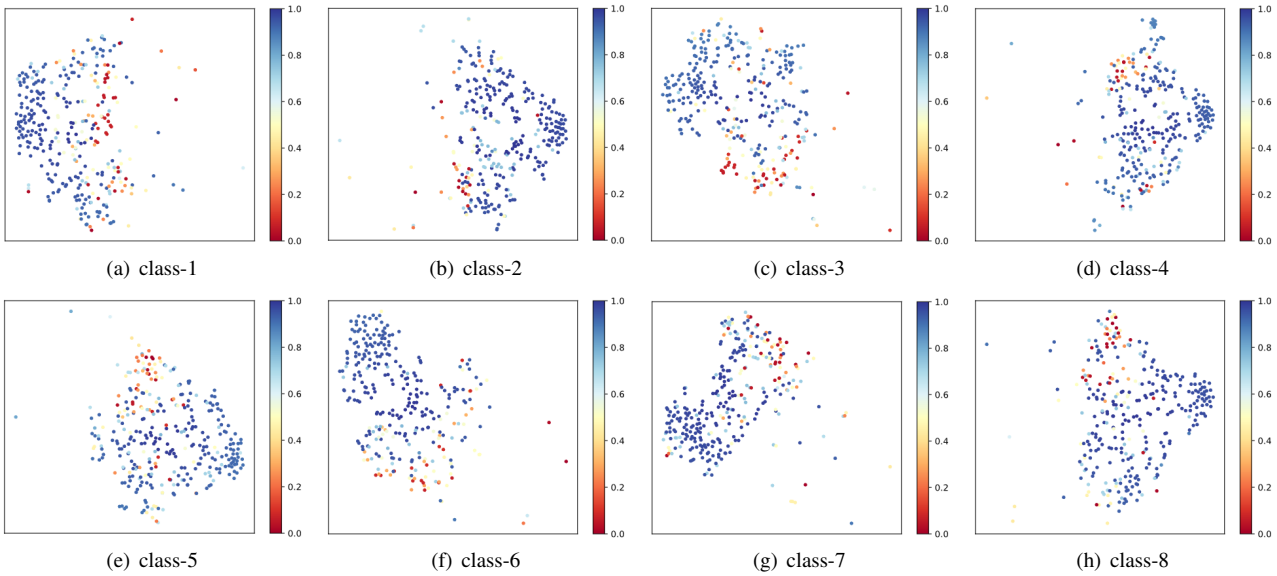
MoCo v2 [2], BYOL [5], BarlowTwins [8], as shown in Table 2. Our C^{2}ASR always achieves better results than existing continual self-supervised learning methods, which shows C^{2}ASR can be well integrated with other self-supervised learning methods and obtain better performance.

B.3. The computational cost with different previous task stream interval

As shown in (10), the length of previous task stream interval plays an important role in $\mathcal{L}^t_{C^2ASR}$, where a large previous task stream interval brings huge computational cost and a small previous task stream interval leads to the infor-

Table 3: **The computational cost with different previous task stream interval**. We report the average accuracy, average forgetting, training efficiency (the training time per epoch) and GPU memory requirements (the peak memory per GPU) of C^{2}ASR with different previous task stream interval on Split CIFAR-100. The performances are measured by calculating the mean and standard deviation across three trials.

Previous task stream interval	1	2	3	4	5
Accuracy (%)	82.36(±0.73)	83.12(±0.92)	83.18(±0.83)	83.20(±0.97)	83.15(±1.06)
Forgetting (%)	2.68(±1.71)	2.22(±1.48)	2.16(±1.38)	2.20(±1.51)	2.18(±1.59)
Training efficiency (s/epoch)	41.4	51.9	62.8	73.4	83.3
GPU memory requirements (MB/GPU)	6271	6828	6842	6859	6888



(a) class-1 (b) class-2 (c) class-3 (d) class-4

(e) class-5 (f) class-6 (g) class-7 (h) class-8

Figure 1: The t-SNE visualization of the pre-trained representations combined with corresponding augmentation stability on Split CIFAR-10. The color bar on the right corresponds the value of the augmentation stability. Samples which are located at the boundary of corresponding category distribution usually have lower augmentation stability scores, i.e., red points. samples within the corresponding category distribution usually have higher augmentation stability scores.

mation leakage problem. In this part, we discuss about the computational cost with different previous task stream interval to find a appropriate previous task stream interval to trade off the computational cost and the information leakage problem.

We report the average accuracy, average forgetting, training efficiency and GPU memory requirements of proposed C^{2}ASR with different previous task stream interval on Split CIFAR-100 in Table 3. We choose the training time per epoch and the peak memory per GPU as the metric of training efficiency and GPU memory requirements, which are commonly used to measure the computational cost in self-supervised learning [1]. Except for the previous task stream interval, we keep all other settings consistent with those in 4.1. We report the training efficiency and GPU memory requirements in a server with an Intel Xeon E5-2620 v4 and an TITAN Xp (4 workers).

From the Table, we can see C^{2}ASR obtains a considerable performance promotion from previous 1 task stream interval to 2, where the information leakage problem is solved, but the training time per epoch and the peak memory per GPU increase 10.5s and 557MB respectively. From previous 2 task stream interval to 5, C^{2}ASR achieves trivial improvements, but the training time per epoch keeps increasing. In the final implementation, we chose previous 2 task stream interval to trade off the computational cost and information leakage.

B.4. Visualization

In this part, we give the t-SNE visualization of the pre-trained representations combined with corresponding augmentation stability on Split CIFAR-10, as shown in Figure

1. We only show the categories in the first four tasks, i.e., task 1 (Figure 1(a), 1(b)), task 2 (Figure 1(c), 1(d)), task 3 (Figure 1(e), 1(f)) and task 4 (Figure 1(g), 1(h)), since the last task doesn't need to be replayed. We can see that the samples located in the center of category distribution often have a large augmentation stability value, while the samples located in the boundary of category distribution are low. This phenomenon is common in all tasks, and becomes the initial motivation of our ASR. The underlying mechanism is that self-supervised learning can learn semantically informative representations by encouraging augmentation invariance, even without manual annotations. Thus, this augmentation stability distribution is also encoded into the feature space by self-supervised models.

References

[1] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In *Advances in Neural Information Processing Systems*, pages 9912–9924, Virtual, December 2020. 3

[2] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv:2003.04297*, 2020. 2

[3] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15750–15758, Virtual, June 2021. 2

[4] Enrico Fini, Victor G Turrisi da Costa, Xavier Alameda-Pineda, Elisa Ricci, Karteek Alahari, and Julien Mairal. Self-supervised models are continual learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9621–9630, Virtual, June 2022. 2

[5] Jean Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. In *Advances in Neural Information Processing Systems*, pages 21271–21284, Virtual, December 2020. 2

[6] Divyam Madaan, Jaehong Yoon, Yuanchun Li, Yunxin Liu, and Sung Ju Hwang. Representational continuity for unsupervised continual learning. In *Proceedings the International Conference on Learning Representations*, Virtual, April 2022. 2

[7] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3733–3742, Salt Lake City, UT, USA, June 2018. 2

[8] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *Proceedings of the International Conference on Machine Learning*, pages 12310–12320, Virtual, July 2021. 2