

A. Training Details

A.1. Augmentation

We follow the data augmentation scheme introduced in [2] for representation learning and linear evaluation. We describe the default set of augmentations following the PyTorch [6] notations in what follows.

- **RandomResizedCrop**. We crop Seq-CIFAR-10, Tiny-ImageNet, and R-MNIST datasets with the scale in $[0.2, 1.0]$, $[0.1, 1.0]$, and $[0.7, 1.0]$, respectively. The cropped images are resized to 32×32 for Seq-CIFAR-10, 64×64 for Tiny-ImageNet, and 28×28 for R-MNIST.
- **RandomHorizontalFlip**. Images are flipped horizontally with probability 0.5.
- **ColorJitter**. The maximum strengths of {brightness, contrast, saturation, hue} are {0.4, 0.4, 0.4, 0.1} with probability 0.8.
- **RandomGrayScale**. Images are grayscaled with probability 0.2.
- **GaussianBlur**. For the Tiny-ImageNet dataset, blur augmentation is applied with Gaussian kernel. Kernel size is 7×7 and the standard deviation is randomly drawn from $[0.1, 2.0]$. This operation is randomly applied with probability 0.5.

A.2. Architecture

For Seq-CIFAR-10 and Tiny-ImageNet datasets, we use ResNet-18 (not pretrained) as a base encoder for representation learning followed by a 2-layer projection MLP which maps representations to a 128-dimensional latent space [4]. The hidden layer of projection MLP consists of 512 hidden units.

For R-MNIST, we use two convolutional layers and one fully connected layer for the base encoder. We use 20 and 50 filters with 5×5 kernel and stride 1 for two convolutional layers, respectively. Each feature map is followed by a max pooling operation with stride 2. The base encoder for R-MNIST is also followed by a 2-layer projection MLP for representation learning. The output dimensions of the last fully connected layer of encoder and the following 2-layer MLP’s all hidden/output neuron sizes are equally 500.

A.3. Hyperparameter

The hyperparameters for Section 5 are selected by performing a grid search using the validation set consisting of randomly drawn 10% of the training samples, and chosen hyperparameters are given in Table 4. We consider following hyperparameters for Co²L: learning rate (η), batch size (bsz), temperature for asymmetric supervised contrastive learning loss (τ), temperatures for instance-wise relation distillation loss (κ, κ^*), and the number of epochs of t -th task (E_t). The hyperparameter search space for Co²L on benchmark

Method	Buffer	Parameters
R-MNIST		
ER	200	η : 0.1
	500	η : 0.1
GEM	200	η : 0.1, γ : 0.5
	500	η : 0.3, γ : 0.5
A-GEM	200	η : 0.1
	500	η : 0.1
FDR	200	η : 0.1, α : 1.0
	500	η : 0.2, α : 0.3
GSS	200	η : 0.2, gmb s: 128, nb : 1
	500	η : 0.2, gmb s: 128, nb : 1
HAL	200	η : 0.03, λ : 0.1, β : 0.3, α : 0.1
	500	η : 0.03, λ : 0.1, β : 0.5, α : 0.1
DER	200	η : 0.1, α : 0.5
	500	η : 0.1, α : 0.5
DER++	200	η : 0.1, α : 1.0, β : 0.5
	500	η : 0.2, α : 1.0, β : 1.0
Co ² L	200	η : 0.01, τ : 0.1, κ : 0.2, κ^* : 0.01, $epoch$: 20
	500	
Seq-CIFAR-10		
Co ² L	200	η : 0.5, τ : 0.5, κ : 0.2, κ^* : 0.01, $epoch$: 100
	500	
Seq-Tiny-ImageNet		
Co ² L	200	η : 0.1, τ : 0.5, κ : 0.1, κ^* : 0.1, $epoch$: 50
	500	

Table 4. Hyperparameters chosen in our experiments

Dataset	Parameter	Values
Seq-CIFAR-10	η	{0.1, 0.5, 1.0}
	τ	{0.1, 0.5, 1.0}
	κ	{0.1, 0.2}
	κ^*	{0.01, 0.05, 0.1}
	E_0	{500}
	$E_{t>0}$	{50, 100}
	bsz	{256, 512, 1024}
Seq-Tiny-ImageNet	η	{0.1, 0.5, 1.0}
	τ	{0.1, 0.5, 1.0}
	κ	{0.1, 0.2}
	κ^*	{0.01, 0.05, 0.1}
	E_0	{500}
	$E_{t>0}$	{50, 100}
	bsz	{256, 512, 1024}
R-MNIST	η	{0.01, 0.05, 0.1}
	τ	{0.1, 0.5, 1.0}
	κ	{0.1, 0.2}
	κ^*	{0.01, 0.05, 0.1}
	E_0	{100}
	$E_{t>0}$	{10, 20}
	bsz	{256, 512, 1024}

Table 5. Hyperparameter space for Co²L

datasets are provided in Table 5. In a combined grid search for Class-IL and Task-IL, we select the best hyperparameters that achieve the highest final accuracy averaged over both settings. The average Class-IL test accuracies with batch

Dataset	Buffer	batch size (bsz)			temperature (τ)		
		256	512	1024	0.1	0.5	1
Seq-CIFAR-10	200	55.20 \pm 2.51	65.57	58.29 \pm 1.26	64.11 \pm 2.79	65.57	62.14 \pm 3.11
	500	52.51 \pm 2.63	74.26	74.09 \pm 0.31	73.94 \pm 0.91	74.26	72.66 \pm 0.34
Seq-Tiny-ImageNet	200	13.18 \pm 0.69	13.88	13.81 \pm 0.57	13.09 \pm 0.38	13.88	12.75 \pm 0.18
	500	16.90 \pm 0.43	20.12	19.74 \pm 0.37	17.98 \pm 0.39	20.12	18.22 \pm 0.28

Table 6. Test accuracies of Co²L with various batch size and temperature hyperparameters (averaged over ten independent trials).

size and temperature hyperparameters are given in Table 6. For R-MNIST, we conduct a grid search for all baselines since the architecture for R-MNIST changes. We follow the hyperparameter search space (and its notations) for R-MNIST given in [1]. For all experiments of Co²L, we use distillation power (λ in eq. 9) as 1.0.

A.4. Training Details for Co²L

For representation learning, we use a linear warmup for the first 10 epochs and decay the learning rate with the cosine decay schedule [5]. The learning rate scheduling is restarted at every task is introduced. We use SGD with momentum 0.9 and weight decay 0.0001 for all experiments.

For linear evaluation, we train a linear classifier for 100 epochs using SGD with momentum 0.9 and no weight decay. We decay the learning rate exponentially at 60, 75, and 90 epoch with decay rate 0.2. We use {1.0, 0.1, 1.0} learning rate for {Seq-CIFAR-10, Seq-Tiny-ImageNet, R-MNIST}.

B. Experiments on IRD Alternatives

We propose Co²L that learns representations and preserves learned representations using $\mathcal{L}_{\text{sup}}^{\text{asym}}$ and \mathcal{L}^{IRD} , respectively. In this section, we explore alternatives for IRD loss to preserve learned representations, and verify its effectiveness. More specifically, we consider following baselines.

Embedding distillation. IRD can be viewed as distilling the representations from the past self, similar to how SEED [3] distills the representation from the teacher model to the student model. However, there is a slight difference: IRD distills the instance-wise similarity of the outputs from the joint encoder-projector, where the projector is introduced for contrastive learning. SEED directly distills the output of the encoder. Specifically, for each sample \tilde{x}_i in a batch \mathcal{B} , the similarity score with respect to an encoder f_ϑ is defined as:

$$\mathbf{p}(\tilde{\mathbf{x}}_i; \vartheta, \gamma) = [p_{i,1}, \dots, p_{i,2N}], \quad (14)$$

where $p_{i,j}$ denotes the normalized similarity

$$p_{i,j} = \frac{\exp(\mathbf{z}_i \cdot \mathbf{z}_j / \gamma)}{\sum_{k \neq i}^{2N} \exp(\mathbf{z}_i \cdot \mathbf{z}_k / \gamma)}, \quad (15)$$

and \mathbf{z}_i denotes the normalized feature vector representations of \tilde{x}_i from the encoder f_ϑ , *i.e.*, $\mathbf{z}_i = f_\vartheta(\tilde{x}_i) / \|f_\vartheta(\tilde{x}_i)\|_2$. Denoting the parameters of the teacher/student encoder and temperature as ϑ^T, ϑ^S and γ^T, γ^S , the SEED loss is defined as

$$\mathcal{L}^{\text{SEED}} = \sum_{i=1}^{2N} -\mathbf{p}(\tilde{\mathbf{x}}_i; \vartheta^T, \gamma^T) \cdot \log \mathbf{p}(\tilde{\mathbf{x}}_i; \vartheta^S, \gamma^S) \quad (16)$$

Logit matching. Buzzega *et al.* [1] shows matching the logit, *i.e.*, pre-softmax outputs, of the past and current model is effective for mitigating forgetting. Similarly, we replace IRD loss with the one that directly matches representation maps. Specifically, for each sample \tilde{x}_i in batch \mathcal{B} , two types of matching loss are defined as

$$\mathcal{L}_{\text{embedding}}^{\text{MSE}} = \frac{1}{2N} \sum_{i=1}^{2N} (f_\vartheta(\tilde{x}_i) - f_{\vartheta^*}(\tilde{x}_i))^2 \quad (17)$$

$$\mathcal{L}_{\text{projection}}^{\text{MSE}} = \frac{1}{2N} \sum_{i=1}^{2N} ((g \circ f)_\psi(\tilde{x}_i) - (g \circ f)_{\psi^*}(\tilde{x}_i))^2 \quad (18)$$

where f_ϑ is encoder and $(g \circ f)_\psi$ is the feature map which maps augmented batch to an unnormalized d -dimensional Euclidean sphere. The difference between $\mathcal{L}_{\text{embedding}}^{\text{MSE}}$ and $\mathcal{L}_{\text{projection}}^{\text{MSE}}$ is the choice of representation maps to be matched; one defined on embedding space and the other defined on the projection space.

As shown in Table 7, we find that distilling the projector output (and thereby applying both \mathcal{L}^{IRD} and $\mathcal{L}_{\text{asym}}^{\text{sup}}$ at the same layer) significantly outperforms distilling at the encoder output ($\mathcal{L}^{\text{SEED}}, \mathcal{L}_{\text{embedding}}^{\text{MSE}}$) and the projection output ($\mathcal{L}_{\text{projection}}^{\text{MSE}}$). Since we learn representations continually in contrastive learning schemes, where similarity is defined on a unit d -dimensional Euclidean sphere, regulating the relation drifts in the projection space can be more effective to preserve learned representations than other alternatives.

Buffer	Objective	Space	Seq-CIFAR-10		Seq-Tiny-ImageNet	
			Class-IL	Task-IL	Class-IL	Task-IL
200	$\mathcal{L}_{sup}^{asym} + \mathcal{L}^{SEED}$ [3]	Embedding	53.42 \pm 1.07	85.79 \pm 0.91	9.23 \pm 0.65	27.02 \pm 1.70
	$\mathcal{L}_{sup}^{asym} + \mathcal{L}^{MSE}$	Embedding	56.31 \pm 2.30	86.12 \pm 0.94	11.03 \pm 0.26	33.15 \pm 0.55
	$\mathcal{L}_{sup}^{asym} + \mathcal{L}^{MSE}$	Projection	53.10 \pm 1.52	85.05 \pm 0.95	11.45 \pm 0.33	34.38 \pm 0.66
	$\mathcal{L}_{sup}^{asym} + \mathcal{L}^{IRD}$ (ours)	Projection	65.57\pm1.37	93.43\pm0.78	13.88\pm0.40	42.37\pm0.74
500	$\mathcal{L}_{sup}^{asym} + \mathcal{L}^{SEED}$ [3]	Embedding	61.65 \pm 3.24	88.40 \pm 2.44	12.04 \pm 0.40	34.91 \pm 0.57
	$\mathcal{L}_{sup}^{asym} + \mathcal{L}^{MSE}$	Embedding	62.83 \pm 2.92	88.63 \pm 2.05	14.89 \pm 0.40	42.25 \pm 0.51
	$\mathcal{L}_{sup}^{asym} + \mathcal{L}^{MSE}$	Projection	57.47 \pm 1.07	86.29 \pm 0.31	14.73 \pm 0.39	41.85 \pm 1.22
	$\mathcal{L}_{sup}^{asym} + \mathcal{L}^{IRD}$ (ours)	Projection	74.26\pm0.77	95.90\pm0.26	20.12\pm0.42	53.04\pm0.69

Table 7. Classification accuracies for Seq-CIFAR-10 and Seq-Tiny-ImageNet on our algorithm and three alternatives. All results are averaged over ten independent trials.

References

- [1] Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and Simone Calderara. Dark experience for general continual learning: a strong, simple baseline. In *Advances in Neural Information Processing Systems*, 2020.
- [2] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *Proceedings of the International Conference on Machine Learning*, 2020.
- [3] Zhiyuan Fang, Jianfeng Wang, Lijuan Wang, Lei Zhang, Yezhou Yang, and Zicheng Liu. SEED: Self-supervised distillation for visual representation. In *International Conference on Learning Representations*, 2021.
- [4] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. In *Advances in Neural Information Processing Systems*, 2020.
- [5] Ilya Loshchilov and Frank Hutter. SGDR: stochastic gradient descent with warm restarts. In *International Conference on Learning Representations*, 2017.
- [6] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*.