模式分析与机器智能
工业和信息化部重点实验室
MIIT Key Laboratory of
Pattern Analysis & Machine Intelligence

ParNeC 模式识别与神经计算研究组
PAttern Recognition and NEural Computing

Loss Decoupling for Task-Agnostic Continual Learning

Yan-Shuo Liang and Wu-Jun Li*

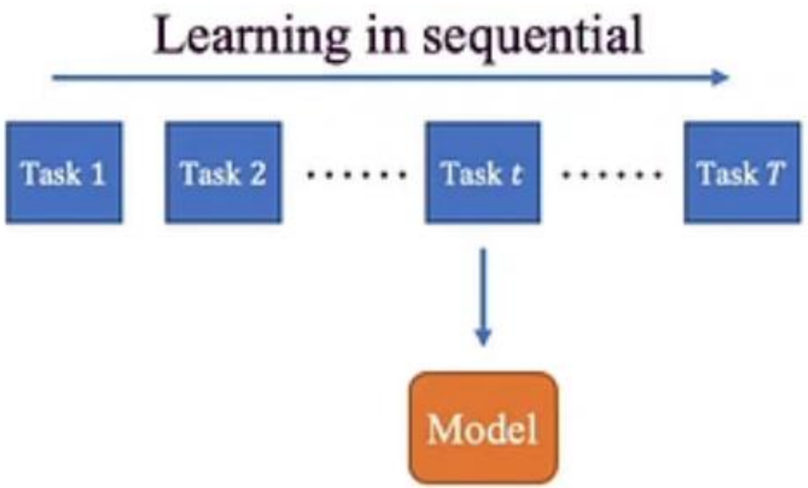
National Key Laboratory for Novel Software Technology,
Department of Computer Science and Technology, Nanjing University, P. R. China

liangys@smail.nju.edu.cn,liwujun@nju.edu.cn

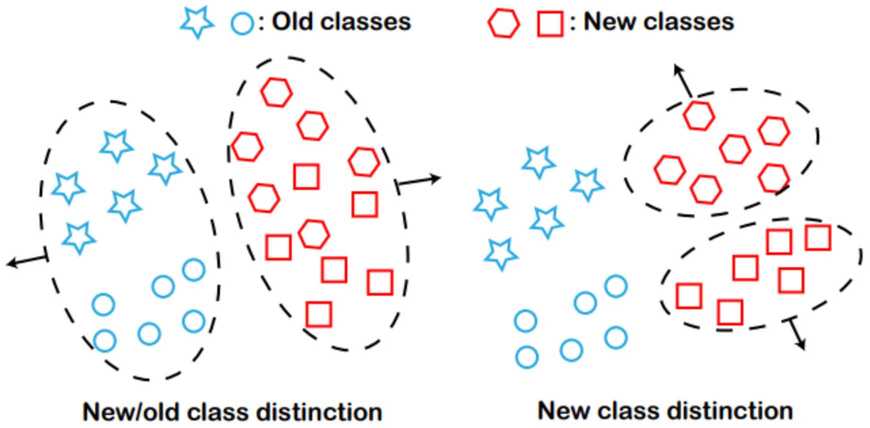
NeurIPS 2023

Background

- Continual learning requires the model to learn multiple task sequentially.



- Task-agnostic problem vs Task-aware problem: Task identities :available or not.



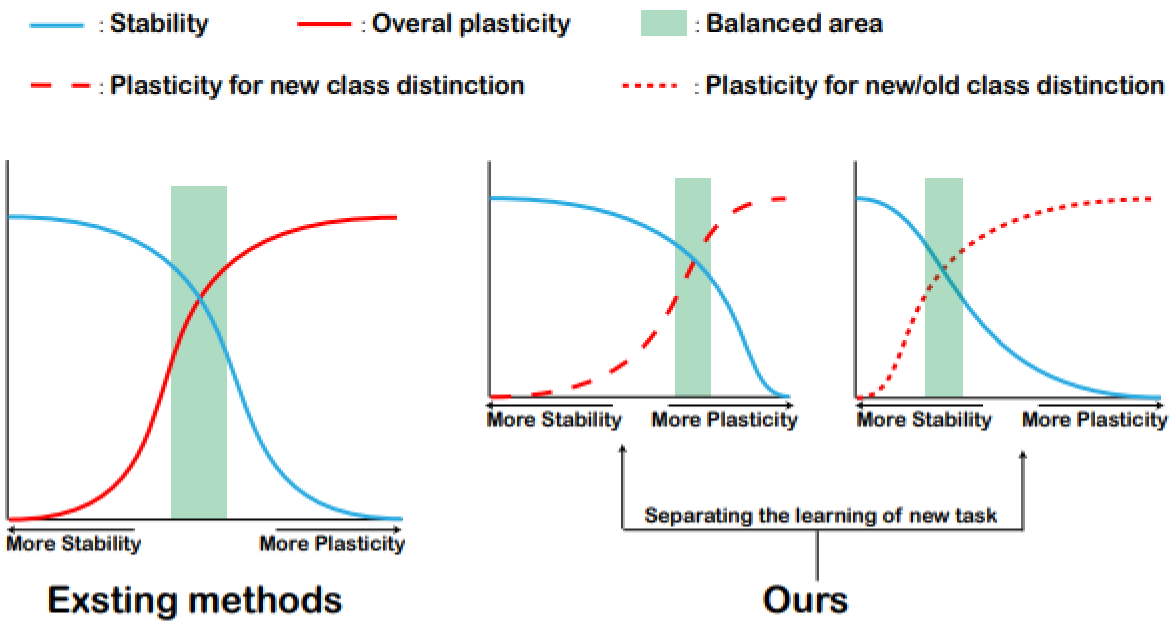
(a) Different learning objectives

Background

PArNeC 模式识别与神经计算研究组 PAttern Recognition and NEural Computing

- Different learning objectives may cause different degrees of forgetting in CL.

- If a new learning objective leads to more forgetting, a good continual learner should pay more attention to the model's stability. Otherwise, a good continual learner should pay more attention to the model's plasticity.



(b) Stability-plasticity trade-off in different methods

Method

- Learning objective for replay-based methods:

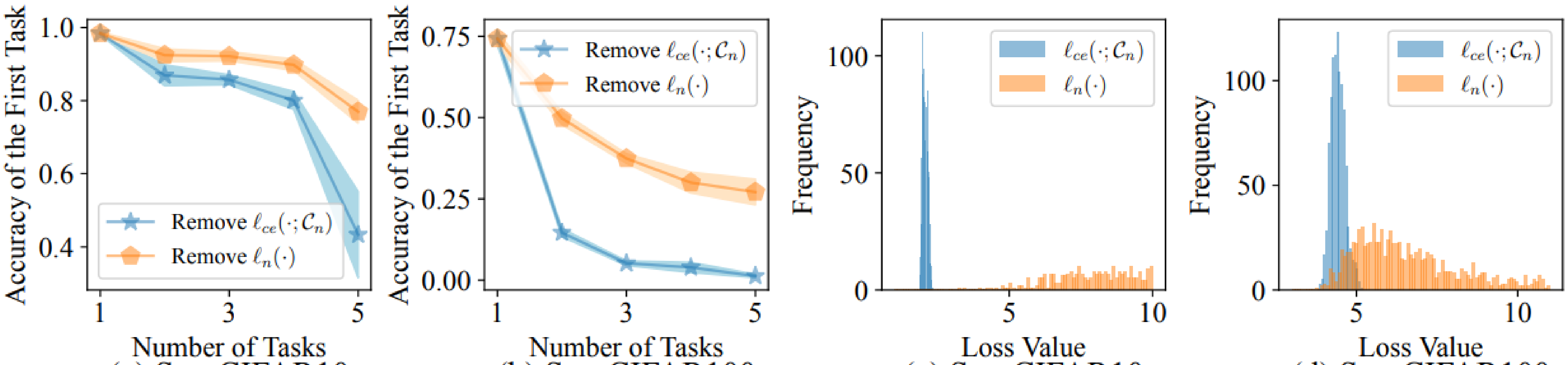
  - $$\mathcal{L} = \frac{1}{|\mathcal{B}_t|}\sum_{i=1}^{|\mathcal{B}_t|}\mathcal{L}_{new}(f_{\Theta}(\boldsymbol{x}_i^t), y_i^t) + \frac{1}{|\mathcal{B}_{\mathcal{M}}|}\sum_{i=1}^{|\mathcal{B}_{\mathcal{M}}|}\mathcal{L}_{rep}(f_{\Theta}(\boldsymbol{x}_i^{\mathcal{M}}), y_i^{\mathcal{M}}). \quad (1)$$

  - We assume that $\mathcal{L}_{new}$ is cross-entropy (CE) loss:

  - $$\mathcal{L}_{new}(f_{\Theta}(\boldsymbol{x}), y) = \ell_{ce}(f_{\Theta}(\boldsymbol{x}), y) = -\log\left(\frac{\exp(o_y)}{\sum_{i=1}^{m+n}\exp(o_i)}\right). \quad (2)$$

Analyzing Learning Objectives by Decoupling Loss

$$\mathcal{L}_{new}(f_{\mathbf{\Theta}}(\boldsymbol{x}), y) = -\log\left(\frac{\exp(o_y)}{\sum_{i=m+1}^{m+n}\exp(o_i)}\right) - \log\left(\frac{\sum_{i=m+1}^{m+n}\exp(o_i)}{\sum_{i=1}^{m+n}\exp(o_i)}\right) \quad (3)$$
$$= \ell_{ce}(f_{\mathbf{\Theta}}(\boldsymbol{x}), y; \mathcal{C}_n) + \ell_n(f_{\mathbf{\Theta}}(\boldsymbol{x})).$$



(a) Seq-CIFAR10 (b) Seq-CIFAR100 (c) Seq-CIFAR10 (d) Seq-CIFAR100

Method

Loss Decoupling for Continual Learning

- $$\mathcal{L} = \frac{1}{|\mathcal{B}_t|}\sum_{i=1}^{|\mathcal{B}_t|}\left(\beta_1 \ell_{ce}(f_{\mathbf{\Theta}}(\boldsymbol{x}_i^t), y_i^t; \mathcal{C}_n) + \beta_2 \ell_n(f_{\mathbf{\Theta}}(\boldsymbol{x}_i^t), y_i^t)\right) + \frac{1}{|\mathcal{B}_{\mathcal{M}}|}\sum_{i=1}^{|\mathcal{B}_{\mathcal{M}}|}\mathcal{L}_{rep}(f_{\mathbf{\Theta}}(\boldsymbol{x}_i^{\mathcal{M}}), y_i^{\mathcal{M}}). \quad (4)$$
- $$\beta_1 = C, \quad \beta_2 = \rho\frac{|\mathcal{C}_n|}{|\mathcal{C}_o|}.$$

- Combining LODE with ER and DER++: The combinations of LODE with these two methods are direct.

- Combining LODE with ESMER：

$$\mathcal{L} = \frac{1}{|\mathcal{B}_t|}\sum_{i=1}^{|\mathcal{B}_t|} w_i\left(\beta_1 \ell_{ce}(f_{\mathbf{\Theta}}(\boldsymbol{x}_i^t), y_i^t; \mathcal{C}_n) + \beta_2 \ell_n(f_{\mathbf{\Theta}}(\boldsymbol{x}_i^t), y_i^t)\right) + \frac{1}{|\mathcal{B}_{\mathcal{M}}|}\sum_{i=1}^{|\mathcal{B}_{\mathcal{M}}|}\mathcal{L}_{rep}(f_{\mathbf{\Theta}}(\boldsymbol{x}_i^{\mathcal{M}}), y_i^{\mathcal{M}}). \quad (6)$$

Relation with Existing Methods

- Let $\beta_1 = 1$ and $\beta_2 = 0$, we can get the of experience replay with asymmetric cross entropy(ER-ACE):

$$\mathcal{L} = \frac{1}{|\mathcal{B}_t|}\sum_{i=1}^{|\mathcal{B}_t|}\mathcal{L}_{ce}(f_\Theta(\boldsymbol{x}_i^t), y_i^t; \mathcal{C}_n) + \frac{1}{|\mathcal{B}_\mathcal{M}|}\sum_{i=1}^{|\mathcal{B}_\mathcal{M}|}\mathcal{L}_{rep}(f_\Theta(\boldsymbol{x}_i^\mathcal{M}), y_i^\mathcal{M}). \tag{7}$$

Algorithm 1 Loss Decoupling (LODE) for Continual Learning

1: **Input:** a sequence of tasks with datasets $\{\mathcal{D}_1, ..., \mathcal{D}_T\}$, a neural network model $f_\Theta(\cdot)$.
2: **Output:** a learned neural network model $f_\Theta(\cdot)$.
3: **while** Get a mini-batch of samples $\mathcal{B}_t$ from a task t **do**
4: Sample a mini-batch $\mathcal{B}_\mathcal{M}$ from memory $\mathcal{M}$;
5: Specify the weights for the two different learning objectives by (5);
6: Get the losses for learning objective through (3);
7: Compute the final loss through (4).
8: Perform backward propagation and update the model $f_\Theta(\cdot)$ through SGD;
9: Update memory $\mathcal{M}$ with $\mathcal{B}_t$ through some memory update methods;
10: **end while**

Experiment

Table 1: Classification results which are averaged across 5 runs.

Keeping Extra Model		**Seq-CIFAR10**		**Seq-CIFAR100**		**Seq-TinyImageNet**	
no	*joint*	91.86±0.26		70.10±0.60		59.82±0.31	
	finetune	19.65±0.03		17.41±0.09		8.13±0.04	
	Buffer Size	500	5120	500	5120	500	5120
no	SCR [27]	57.95±1.57	82.47±0.44	23.06±0.22	45.02±0.67	8.37±0.26	18.20±0.48
	PCR [25]	65.74±3.29	82.58±0.42	28.38±0.46	52.51±1.61	11.88±1.61	26.39±1.64
	MIR [3]	63.93±0.39	83.73±0.97	27.80±0.52	53.73±0.82	11.22±0.43	30.60±0.40
	ER-ACE [8]	68.45±1.78	83.49±0.40	40.67±0.06	58.56±0.91	17.73±0.56	37.99±0.17
	ER [9]	61.78±0.72	83.64±0.95	27.69±0.58	53.86±0.57	10.36±0.11	27.54±0.30
	LODE (ER)	68.87±0.71	83.73±0.48	41.52±1.22	58.59±0.48	17.77±1.03	38.34±0.04
	DER++ [7]	73.29±0.96	85.66±0.14	42.08±1.71	62.73±0.58	19.28±0.61	39.72±0.47
	LODE (DER++)	**75.45**±**0.90**	**85.78**±**0.40**	**46.31**±**1.01**	**64.00**±**0.48**	**21.15**±**0.68**	**40.31**±**0.03**
yes	CLS-ER [5]	70.73±0.54	**85.73**±**0.29**	51.21±0.84	60.17±0.38	29.44±1.66	45.66±0.47
	TAMiL [6]	74.25±0.31	84.82±1.77	50.62±0.23	63.77±0.43	27.83±0.41	43.00±0.56
	iCaRL [33]	61.60±2.03	72.01±0.62	49.59±0.95	54.23±0.28	20.01±0.50	30.34±0.18
	BIC [43]	52.63±2.46	79.98±1.49	37.06±0.60	60.43±0.61	29.82±0.88	37.60±0.23
	SSIL [1]	64.31±0.89	71.72±1.47	41.61±0.37	57.53±0.52	16.80±0.71	40.06±0.58
	ESMER [36]	71.48±0.98	79.19±0.68	52.37±0.87	63.99±0.13	30.97±1.12	44.07±0.52
	LODE (ESMER)	**74.53**±**0.95**	85.34±0.41	**55.06**±**0.35**	**65.69**±**0.33**	**32.15**±**0.17**	**46.40**±**0.46**

Experiment

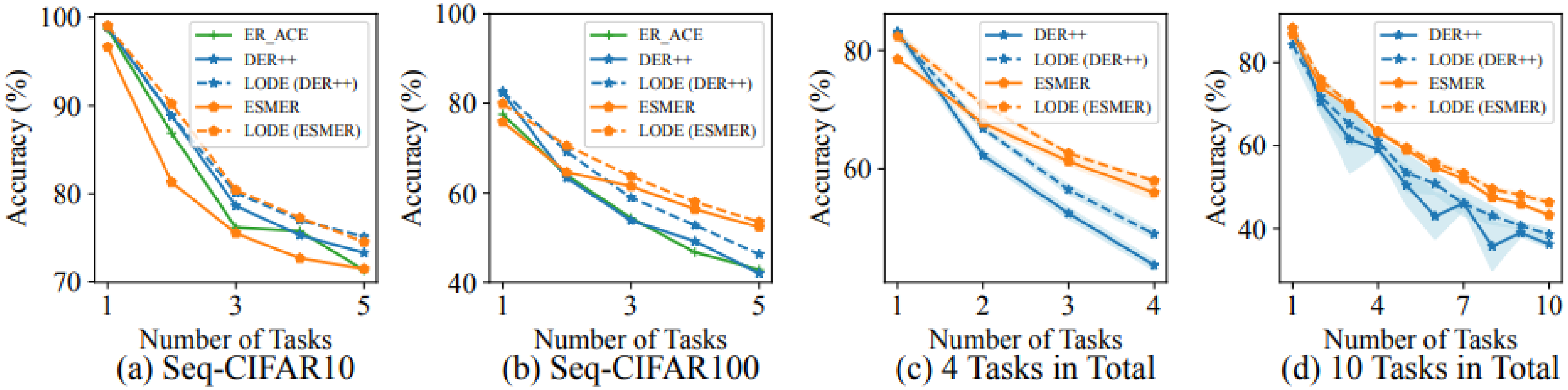


Figure 3: (a) and (b) show the variation of the accuracy for different methods on Seq-CIFAR10 and Seq-CIFAR100. (c) and (d) show the variation of accuracy on Seq-CIFAR100 with different number of tasks.

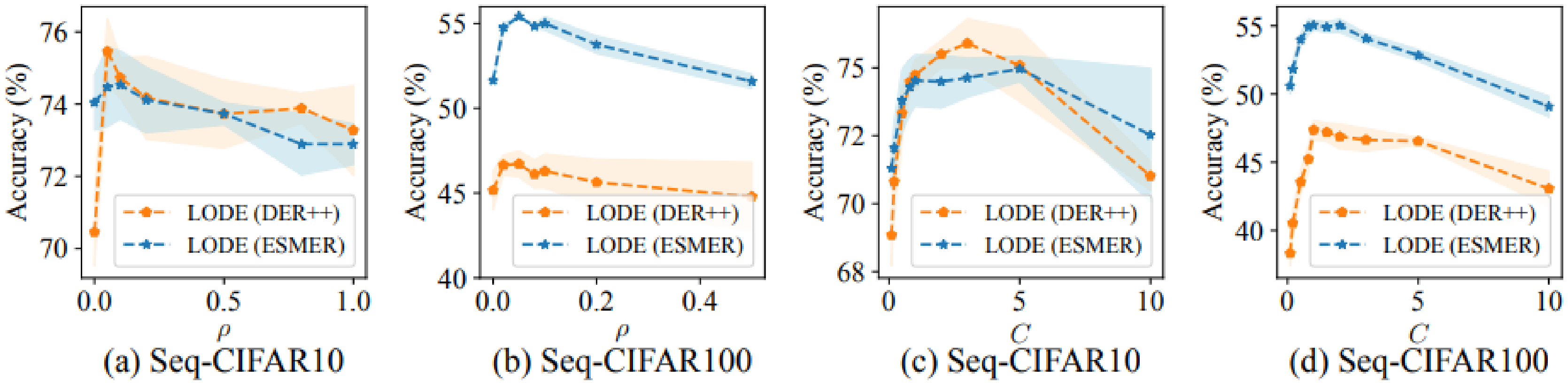


Figure 4: (a) and (b) show the variation of the accuracy for different ρ. (c) and (d) show the variation of the accuracy for different C.

Experiment

Table 2: Ablation study on Seq-CIFAR10 and Seq-CIFAR100.

	LODE (DER++)		LODE (ESMER)	
	Seq-CIFAR10	Seq-CIFAR100	Seq-CIFAR10	Seq-CIFAR100
$\beta_1 = C, \beta_2 = \rho\frac{\lvert\mathcal{C}_n\rvert}{\lvert\mathcal{C}_o\rvert}$ (Ours)	**75.45**$\pm$**0.90**	**46.31**$\pm$**1.01**	**74.53**$\pm$**0.95**	**55.06**$\pm$**0.35**
$\beta_1 = \beta_2 = \rho\frac{\lvert\mathcal{C}_n\rvert}{\lvert\mathcal{C}_o\rvert}$	71.18$\pm$0.80	37.49$\pm$1.79	73.41$\pm$0.40	45.64$\pm$0.87
$\beta_1 = \beta_2 = C$	73.80$\pm$0.72	42.08$\pm$1.71	73.08$\pm$0.81	52.37$\pm$0.87
$\beta_1 = \rho\frac{\lvert\mathcal{C}_n\rvert}{\lvert\mathcal{C}_o\rvert}, \beta_2 = C$	73.19$\pm$0.15	40.79$\pm$0.12	72.38$\pm$0.24	51.86$\pm$0.35

Table 3: Classification results which are averaged across 5 runs in the online continual learning setting.

Keeping Extra Model		Seq-CIFAR10	Seq-CIFAR100	Seq-TinyImageNet
no	SCR [27]	69.49$\pm$3.02	36.09$\pm$0.82	20.04$\pm$1.24
	PCR [25]	73.28$\pm$1.83	34.89$\pm$0.67	23.84$\pm$0.60
	ER-ACE [8]	69.17$\pm$1.64	35.24$\pm$0.51	23.42$\pm$0.34
	MIR [3]	71.10$\pm$1.59	35.08$\pm$1.32	20.64$\pm$1.17
	ER [9]	67.93$\pm$2.04	34.40$\pm$1.13	21.14$\pm$0.72
	LODE (ER)	69.63$\pm$1.41	36.91$\pm$1.38	24.31$\pm$0.82
	DER++ [7]	72.30$\pm$0.99	34.72$\pm$1.51	20.40$\pm$1.02
	LODE (DER++)	**74.00**$\pm$**0.08**	**37.82**$\pm$**1.16**	**25.30**$\pm$**1.80**

Thanks