Continual Learning via Hessian Aware Low-Rank Perturbation

Jiaqi Li

Department of Computer Science,
Western University

September 19, 2024

Background: Continual Learning

Continual Learning (CL): learn from a sequence of tasks with a single model

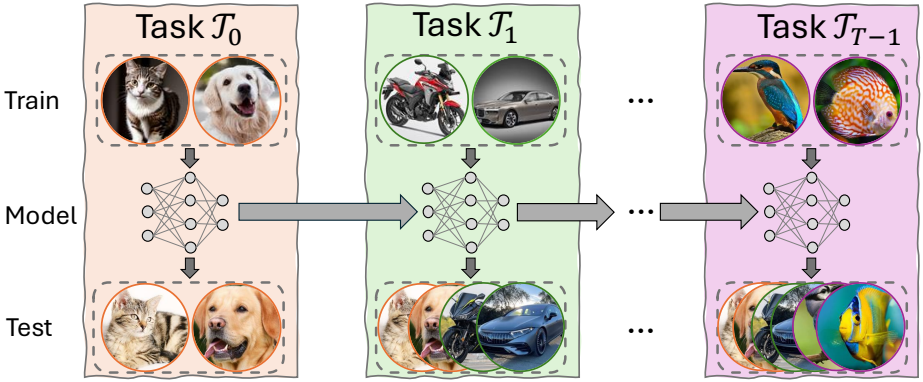
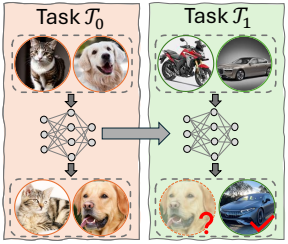
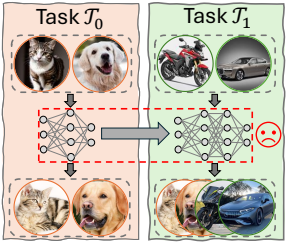


Figure: Continual learning along the tasks $\mathcal{T}_0, \mathcal{T}_1, \cdots, \mathcal{T}_{T-1}$

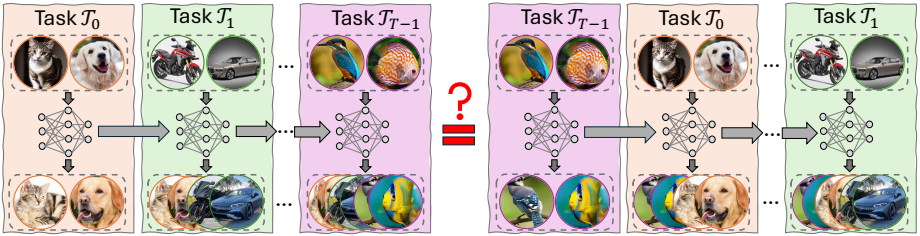
Background: Challenges in Continual Learning



(a) Catastrophic Forgetting



(b) Model Size Explosion



(c) Performance Robustness w.r.t. Different Task Orders

Overview: Hessian Aware Low-Rank Perturbation (HALRP) [1]

- Modeling parameter transition;
- Low-rank approximation on weights;
- Rank selection with Hessian information.

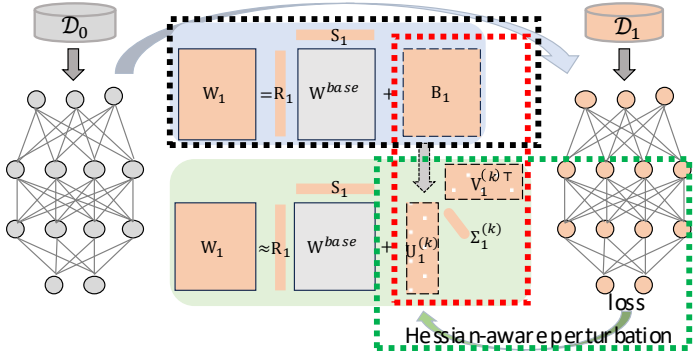


Figure: Framework of HALRP

Methodology: Parameters Transition (Example for $\mathcal{T}_0$ to $\mathcal{T}_1$)

- Learn $\mathcal{T}_0$: $\mathbf{W}^{base} = \arg\min_{\mathbf{W}} \mathcal{L}(\mathbf{W}; \mathcal{D}_0) \in \mathbb{R}^{J \times I}$;
- Learn $\mathcal{T}_1$:

$$\mathbf{W}_1 = \mathbf{R}_1 \mathbf{W}^{base} \mathbf{S}_1 + \mathbf{B}_1 \tag{1}$$

where $\mathbf{B}_1 \in \mathbb{R}^{J \times I}$ is a residual matrix,

$\mathbf{R}_1 = \begin{bmatrix} r_0 & & \\ & \ddots & \\ & & r_J \end{bmatrix}$ and $\mathbf{S}_1 = \begin{bmatrix} s_0 & & \\ & \ddots & \\ & & s_I \end{bmatrix}$ are (diagonal) scaling matrices.

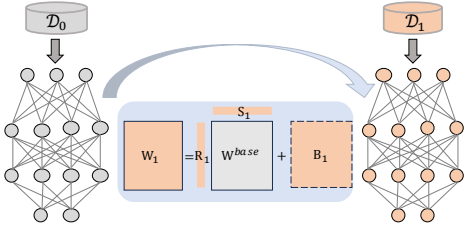


Figure: Parameters Transition from $\mathcal{T}_0$ to $\mathcal{T}_1$

Methodology: Initialization of $\mathbf{R}, \mathbf{S}, \mathbf{B}$

1. Warm-up training on $\mathcal{T}_1$ (*e.g.*, one or two epochs)

$$\mathbf{W}_1^{free} = \arg\max_{\mathbf{W}} \mathcal{L}(\mathbf{W}; \mathcal{D}_1) \tag{2}$$

2. Initialize $\mathbf{R}_1, \mathbf{S}_1, \mathbf{B}_1$ via Least Squares Minimization (LSM):

$$\mathbf{R}_1^{free}, \mathbf{S}_1^{free}, \mathbf{B}_1^{free} = \arg\min_{\mathbf{R},\mathbf{S},\mathbf{B}} \|\mathbf{W}_1^{free} - \mathbf{R}\mathbf{W}^{base}\mathbf{S} - \mathbf{B}\|_F^2 \tag{3}$$

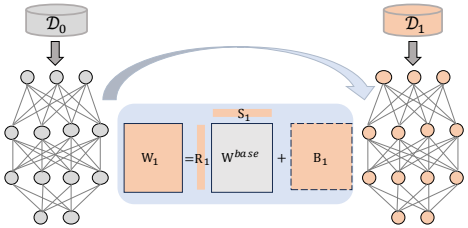


Figure: Parameters Transition from $\mathcal{T}_0$ to $\mathcal{T}_1$

Methodology: Low-Rank Approximation on Residual Weights $\mathbf{B}$

$\Longrightarrow$ Compress $\mathbf{B}_1$ via the low-rank approximation.

$$\mathbf{U}_1, \Sigma_1, \mathbf{V}_1 \leftarrow \mathrm{SVD}(\mathbf{B}_1) \tag{4}$$

Establish the k-rank approximation for $\mathbf{B}_1^{free}$ (then for $\mathbf{W}_1^{free}$)

$$\mathbf{W}_1^{free} \approx \mathbf{W}_1^{(k)free} = \mathbf{R}_1 \mathbf{W}^{base} \mathbf{S}_1 + \mathbf{U}_1^{(k)free} \Sigma_1^{(k)free} (\mathbf{V}_1^{(k)free})^\top, \tag{5}$$

where $\mathbf{B}_1^{free} \approx \mathbf{B}_1^{(k)free} = \mathbf{U}_1^{(k)free} \Sigma_1^{(k)free} (\mathbf{V}_1^{(k)free})^\top$

The approximation error is caused by the perturbation on weights: $\Delta \mathbf{W}_1^{free} = \mathbf{W}_1^{free} - \mathbf{W}_1^{(k)free}$

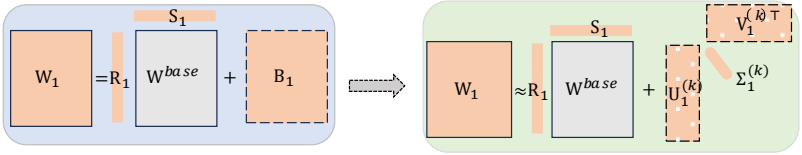


Figure: Low-Rank Approximation on Model Weights.

Theoretical Support: Rank Selection through Hessian Information

Problem: how to choose a proper k_l (i.e., preserved ranks) for each layer l?

Theorem 1 (Relationship between Loss and Weights Perturbation)

Assume that a neural network of L layers with vectorized weights $(\boldsymbol{\omega}_1^\star, \ldots, \boldsymbol{\omega}_L^\star)$ that have converged to local optima, such that the first and second order optimally conditions are satisfied, i.e., the gradient is zero, and the Hessian is positive semi-definite. Suppose a perturbation $\Delta\boldsymbol{\omega}_1^\star$ applied to the first layer weights, then we have the loss change

$$|\mathcal{L}(\boldsymbol{\omega}_1^\star - \Delta\boldsymbol{\omega}_1^\star, \ldots, \boldsymbol{\omega}_L^\star) - \mathcal{L}(\boldsymbol{\omega}_1^\star, \ldots, \boldsymbol{\omega}_L^\star)| \leq \frac{1}{2}\|\mathbf{H}_1\|_F \cdot \|\Delta\boldsymbol{\omega}_1^\star\|_F^2 + o\left(\|\Delta\boldsymbol{\omega}_1^\star\|_F^2\right), \quad (6)$$

where $\mathbf{H}_1 = \nabla^2\mathcal{L}(\boldsymbol{\omega}_1^\star)$ is the Hessian matrix at only the variables of the first layer weights.

Take-way:

Risk change by weight perturbation is relevant to Hessian information!

Methodology: Rank Selection through Hessian Information

- Apply Theorem 1 to the proposed low-rank approximation:

$$|\mathcal{L}(\mathbf{W}_1^{(k)free}) - \mathcal{L}(\mathbf{W}_1^{free})| \leq \frac{1}{2}\|\mathbf{H}_1\|_F \left(\sum_{i=k+1}^{r} \sigma_i^2\right) + o\left(\sum_{i=k+1}^{r} \sigma_i^2\right). \tag{7}$$

less approximation error $\leftarrow$ **keep more ranks**

- Measure the contribution of a rank k for the layer l:
  (**Notes**: According to Kunstner et al. (2019), $\|\mathbf{H}_1\|_F$ can be approximated by negative empirical Fisher information with $\|\mathbf{g}_1\|_F^2$, where $\mathbf{g}_1 = \frac{\partial \mathcal{L}}{\partial \mathbf{W}_1} |_{\mathbf{W}_1 = \mathbf{W}_1^{free}}$.)

$$\|\mathbf{g}_l\|_F^2 \sigma_{l,k}^2 \tag{8}$$

  with
  - $\mathbf{g}_l$: the gradient for the layer-l;
  - $\sigma_{l,k}$: the k-th singular value of the layer-l;

Methodology: Rank Selection through Hessian Information

For a given approximation rate α (*e.g.*, 0.9), the preserved ranks k_l for layer l is determined by

$$\begin{aligned} \min_{k_1,\ldots,k_L} \quad & \sum_{l=1}^{L}\sum_{i=1}^{k_l} \|\mathbf{g}_l\|_F^2 \sigma_{l,i}^2 \\ \text{s.t.} \quad & \sum_{l=1}^{L}\sum_{i=1}^{k_l} \|\mathbf{g}_l\|_F^2 \sigma_{l,i}^2 \geq \alpha \left(\sum_{l=1}^{L}\sum_{i=1}^{r_l} \|\mathbf{g}_l\|_F^2 \sigma_{l,i}^2 \right) \end{aligned} \tag{9}$$

where L is total number of layers, r_l (with $k_l \leq r_l$) is the full rank of layer l.

Remark:

Eq. 9 provides a *global perspective* for the trade-off between the approximation error and computational efficiency.

Methodology: Rank Selection Implementation (An Example)

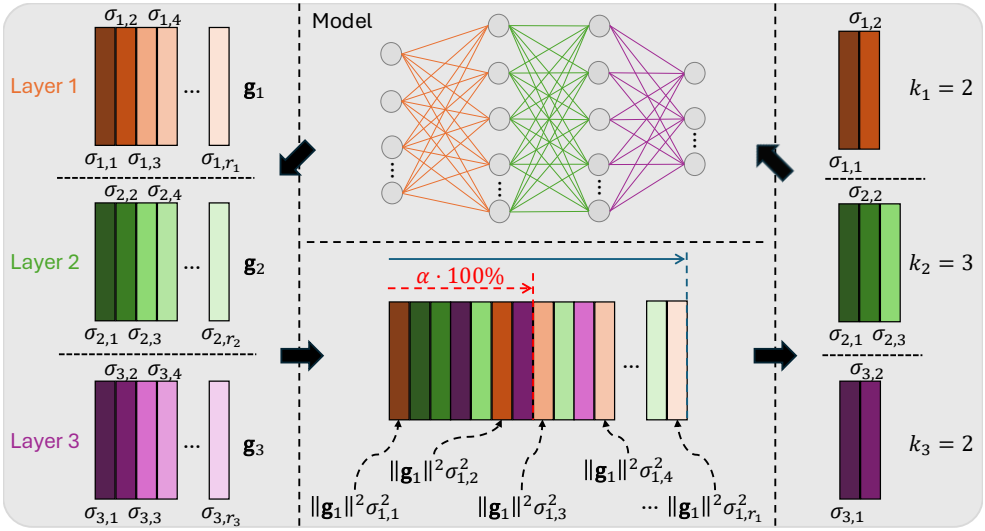


Figure: Hessian-Aware Rank Selection for a Neural Network with $L = 3$.

Methodology: Fine-tuning after Rank Selection

- Re-initialize model with $\{\mathbf{W}^{base}, \mathbf{R}_1^{free}, \mathbf{S}_1^{free}, \mathbf{U}^{(k)free}, \Sigma^{(k)free}, \mathbf{V}^{(k)free}\}$
- Fine-tune the model through

$$\min_{\mathbf{W}} \mathcal{L}(\mathbf{W}; \mathcal{D}_t) + \mathcal{L}_{\text{reg}}(\mathbf{W}; \lambda_0, \lambda_1) \tag{10}$$

  where $\mathcal{L}_{\text{reg}}(\mathbf{W}_t; \lambda_0, \lambda_1)$ is a regularization loss with L_1-regularizer and L_2-regularizer on $\{\mathbf{R}, \mathbf{S}, \mathbf{U}^{(k)}, \mathbf{V}^{(k)}\}$ with coefficients λ_0 and λ_1, respectively.
- (Optional) Weight pruning if model increment ratio $>$ a threshold p:
  - by an absolute value threshold;
  - by a percentile;
  - or a combination of the above two.

Methodology: Algorithm Description[1]

Algorithm Hessian Aware Low-Rank Perturbation (HALRP) for Continual Learning

Require: Task data $\{\mathcal{D}_t\}_{i=0}^{T-1}$; total epochs for one task n; rank estimation epochs n_w; parameter increments limitation ratio p, approximation rate α.

Ensure: Base weights $\mathbf{W}^{\text{base}}$ and $\{\mathbf{W}_t^\star\}_{t=1}^{T-1}$ for each task.

1: Obtain $\mathbf{W}^{\text{base}} = \arg\min_{\mathbf{W}} \mathcal{L}(\mathbf{W}; \mathcal{D}_0)$ on task $\mathcal{T}_0$.
2: **for** $t = 1, \cdots, T-1$ **do**
3: Warm-up pre-training on task $\mathcal{T}_t$ for n_w epochs: $\mathbf{W}_t^{\text{free}} = \arg\min_{\mathbf{W}} L(\mathbf{W}; \mathcal{D}_t)$.
4: Decomposition for all layers: $\mathbf{W}_t^{\text{free}} = \mathbf{R}_t^{\text{free}} \mathbf{W}^{\text{base}} \mathbf{S}_t^{\text{free}} + \mathbf{B}_t^{\text{free}}$.
5: Apply SVD on $\mathbf{B}_t^{\text{free}}$.
6: Select the ranks k_l for each layer l.
7: Re-initialize the task $\mathcal{T}_t$ parameters approximated weights $\{\mathbf{R}_t^{\text{free}}, \mathbf{S}_t^{\text{free}}, \mathbf{U}_t^{(k)\text{free}}, \Sigma_t^{(k)\text{free}}, \mathbf{V}_t^{(k)\text{free}}\}$.
8: Fine-tuning on $\mathcal{T}_t$ for $(n - n_w)$ epochs with $\mathbf{W}_t^\star = \arg\min_{\mathbf{W}} \mathcal{L}(\mathbf{W}; \mathcal{D}_t) + \mathcal{L}_{\text{reg}}(\mathbf{W})$
9: If the model increment ratio is larger than a threshold p, apply pruning.
10: **end for**
11: **return** $\mathbf{W}^{\text{base}}$ and $\{\mathbf{W}_t^\star\}_{t=1}^{T-1}$.

[1] Source code: https://github.com/lijiaqi/HALRP

Experiments: Evaluation Metrics

Define $A_{t,i}$: the accuracy on task i after learning task t (with $i \leq t$),

- **Average Accuracy** (ACC ↑):

$$\text{Acc} = \frac{1}{T}\sum_{i=0}^{T-1} A_{T-1,i},$$

- **Backward Transfer** (BWT ↓):

$$\text{BWT} = \frac{1}{T}\sum_{i=0}^{T-1}(A_{i,i} - A_{T-1,i})$$

- **Order-normalized Performance Disparity** (OPD ↓): task i performance under R different task orders $A^1_{T-1,i}, ..., A^R_{T-1,i}$ (i.e., $R = 5$)

$$\begin{aligned} \text{OPD}_i &\triangleq \max\{A^1_{T-1,i}, ..., A^R_{T-1,i}\} - \min\{A^1_{T-1,i}, ..., A^R_{T-1,i}\} \\ \text{MOPD} &\triangleq \max\{\text{OPD}_0, ..., \text{OPD}_{T-1}\} \\ \text{AOPD} &\triangleq \frac{1}{T}\sum_{i=0}^{T-1} \text{OPD}_i \end{aligned} \tag{11}$$

Experiments: Performances with Different Amounts of Data

Method	CIFAR100-Split (with LeNet)				CIFAR100-SuperClass (with LeNet)			
	5%	25%	50%	100%	5%	25%	50%	100%
STL	45.13 ± 0.04	59.04 ± 0.03	64.38 ± 0.06	69.55 ± 0.06	43.76 ± 0.68	56.09 ± 0.07	60.06 ± 0.06	64.47 ± 0.05
MTL	44.95 ± 0.11	60.21 ± 0.28	65.65 ± 0.20	69.70 ± 0.28	40.43 ± 0.15	49.88 ± 0.27	53.83 ± 0.27	55.62 ± 0.41
L2	37.15 ± 0.21	48.86 ± 0.28	53.35 ± 0.34	58.09 ± 0.43	34.03 ± 0.08	43.40 ± 0.27	46.10 ± 0.28	48.75 ± 0.24
EWC	37.76 ± 0.20	50.09 ± 0.38	55.65 ± 0.40	60.53 ± 0.26	33.70 ± 0.32	44.02 ± 0.39	47.35 ± 0.47	49.97 ± 0.39
BN	37.60 ± 0.17	50.70 ± 0.28	54.79 ± 0.28	60.34 ± 0.40	36.76 ± 0.14	48.20 ± 0.16	51.43 ± 0.16	55.44 ± 0.19
BE	37.63 ± 0.15	51.13 ± 0.3	55.37 ± 0.28	61.09 ± 0.33	37.05 ± 0.20	48.48 ± 0.14	51.78 ± 0.17	55.97 ± 0.17
APD	36.60 ± 0.14	54.59 ± 0.07	59.71 ± 0.03	66.54 ± 0.03	32.81 ± 0.29	49.00 ± 0.06	52.64 ± 0.19	60.54 ± 0.23
APDfix	35.66 ± 0.33	54.62 ± 0.11	59.86 ± 0.24	66.64 ± 0.14	24.27 ± 0.22	48.71 ± 0.11	53.42 ± 0.12	61.47 ± 0.16
IBWPF	38.35 ± 0.26	47.87 ± 0.25	53.46 ± 0.13	57.13 ±0.15	33.09 ± 0.50	51.32 ± 0.27	52.52 ± 0.26	55.98 ± 0.33
GPM	32.86 ± 0.35	51.61 ± 0.22	57.60 ± 0.19	64.49 ± 0.10	34.88 ± 0.30	47.31 ± 0.51	51.23 ± 0.55	57.91 ± 0.28
WSN	37.01 ± 0.63	55.21 ± 0.59	61.56 ± 0.42	66.56 ± 0.49	36.89 ± 0.49	52.42 ± 0.62	58.23 ± 0.38	61.81 ± 0.54
BMKP	42.36 ± 0.90	56.81 ± 1.05	62.87 ± 0.60	66.95 ± 0.53	37.26 ± 0.87	53.62 ± 0.59	57.76 ± 0.66	61.97 ± 0.19
CLR	36.46 ± 0.29	51.44 ± 0.35	57.00 ± 0.43	61.83 ± 0.60	37.93 ± 0.25	49.82 ± 0.56	53.86 ± 0.63	57.07 ± 0.60
PRD	31.58 ± 0.29	56.07 ± 0.22	59.61 ± 0.33	62.74 ± 0.45	33.34 ± 0.48	52.99 ± 0.28	55.85 ± 0.45	57.80 ± 0.50
HALRP	**45.09 ± 0.05**	**58.94 ± 0.09**	**63.61 ± 0.08**	**67.92 ± 0.17**	**43.84 ± 0.04**	**54.93 ± 0.04**	**58.68 ± 0.11**	**62.56 ± 0.30**

Table: Accuracy (↑) on CIFAR100-Splits(10 tasks) and -Superclass(20 tasks) datasets with different percentages of training data.

Our proposed HALRP performs well under limited-data scenarios.

Experiments: Diverse Datasets/Task Order Robustness

Method	P-MNIST LeNet Acc.↑	MOPD↓	AOPD↓
STL	98.24±0.01	0.15	0.09
MTL	96.70±0.07	1.58	0.81
L2	79.14±0.70	29.66	18.94
EWC	81.69±0.86	21.51	12.16
BN	81.04±0.15	19.77	8.18
BE	83.80±0.08	16.76	6.88
APD	97.94±0.02	0.25	0.16
APDfix	97.99±0.01	0.10	0.11
GPM	96.69±0.02	0.45	0.27
WSN	97.91±0.02	0.36	0.22
BMKP	97.08±0.01	3.21	1.04
CLR	88.55±0.20	14.97	7.88
PRD	83.16±0.17	7.91	6.16
HALRP	**98.10±0.03**	0.47	0.24

Method	Five-dataset AlexNet Acc.↑	MOPD↓	AOPD↓	ResNet-18 Acc.↑	MOPD↓	AOPD↓
STL	89.32±0.06	0.74	0.30	94.24±0.05	0.67	0.24
MTL	88.02±0.18	2.08	0.68	93.82±0.06	0.67	0.30
L2	78.24±2.00	35.11	15.35	85.94±2.79	37.03	12.72
EWC	78.44±2.20	33.29	13.18	86.32±2.80	33.84	11.80
BN	82.35±2.45	34.83	12.56	88.36±2.21	30.22	9.55
BE	82.91±2.37	33.91	11.63	88.75±2.14	29.22	8.97
APD	83.70±0.90	4.80	3.45	92.18±0.28	3.50	1.54
APDfix	84.03±1.24	5.50	3.66	91.91±0.48	6.74	1.98
GPM	87.27±0.61	4.54	1.88	88.52±0.28	6.97	2.82
WSN	86.74±0.40	8.54	2.89	92.58±0.39	4.62	1.21
BMKP	84.03±0.55	9.32	3.07	92.57±0.65	9.08	2.13
CLR	86.68±1.41	19.78	7.08	90.04±1.04	14.05	4.51
PRD	74.74±0.69	17.53	9.23	88.45±0.93	14.17	5.37
HALRP	**88.81±0.31**	4.28	1.31	**93.39±0.30**	4.39	1.27

Method	Omniglot-Rotation LeNet Acc.↑	MOPD↓	AOPD↓
STL	80.93±0.18	20.83	3.42
MTL	93.95±0.11	6.25	2.11
L2	69.86±1.23	17.23	6.96
EWC	69.75±1.28	21.39	7.02
BN	77.08±0.86	14.41	5.58
BE	78.24±0.69	17.17	5.53
APD(⋆)	81.60±0.53	8.19	3.78
APDfix	78.14±0.12	6.53	2.63
GPM	80.41±0.16	28.33	13.02
WSN	82.55±0.44	17.09	7.57
BMKP	81.12±2.71	26.53	16.17
CLR	72.75±1.41	24.30	12.27
PRD	74.49±2.78	49.17	18.44
HALRP	**83.08±0.73**	10.36	3.91

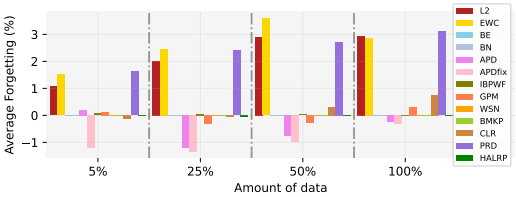
Method	CIFAR100-Split (with LeNet) 5% MOPD↓	AOPD↓	25% MOPD↓	AOPD↓	50% MOPD↓	AOPD↓	100% MOPD↓	AOPD↓	CIFAR100-SuperClass (with LeNet) 5% MOPD↓	AOPD↓	25% MOPD↓	AOPD↓	50% MOPD↓	AOPD↓	100% MOPD↓	AOPD↓
STL	1.96	1.38	3.44	2.41	3.68	2.57	4.38	3.13	2.52	1.48	3.64	1.44	2.52	1.48	2.64	1.45
MTL	1.44	0.93	1.34	0.87	1.66	0.71	1.54	0.84	6.96	2.66	12.04	4.89	13.96	5.74	13.48	6.29
L2	11.66	6.13	13.14	6.96	15.02	7.50	15.18	7.15	9.92	4.55	13.24	6.43	13.32	7.04	14.44	6.82
EWC	11.92	6.07	13.20	6.85	13.78	6.96	13.44	6.48	13.24	5.60	10.92	6.53	13.48	8.03	21.60	8.32
BN	11.70	5.95	13.28	7.37	12.40	6.91	13.92	7.90	8.52	3.35	11.20	3.99	11.64	4.05	11.40	4.25
BE	12.12	5.47	12.92	6.13	9.28	5.56	8.50	5.35	9.56	3.46	11.88	4.00	11.80	3.88	10.84	4.03
APD	11.42	7.21	6.48	3.77	8.00	4.10	8.88	4.07	26.56	12.98	8.64	4.39	8.28	4.48	6.72	3.26
APDfix	6.64	4.39	8.32	4.94	8.92	5.54	7.40	4.21	9.04	4.90	10.28	5.68	8.56	5.32	6.04	2.70
IBPWF	4.30	2.84	4.60	2.73	5.40	3.07	3.68	2.68	14.84	7.45	4.44	2.45	5.36	2.86	5.52	3.38
GPM	11.14	6.37	6.32	4.22	6.32	3.69	2.28	1.348	9.32	4.46	10.00	5.53	8.84	5.89	7.68	4.47
WSN	4.16	2.57	4.42	2.71	4.2	2.62	3.56	2.39	5.08	3.14	4.76	3.192	4.00	2.16	3.76	2.36
BMKP	12.98	7.63	13.22	6.50	6.52	4.30	8.34	3.35	11.72	6.03	13.32	5.27	11.08	4.43	3.72	1.99
CLR	13.50	7.14	12.77	6.13	8.90	5.82	9.37	5.34	10.00	4.01	7.67	3.84	6.60	3.71	9.00	4.06
PRD	5.80	3.80	4.60	2.66	4.16	2.63	5.20	3.16	7.47	4.28	8.33	3.95	5.33	3.29	5.00	2.86
HALRP	2.56	1.34	2.58	1.44	2.34	1.71	3.90	2.56	2.96	1.65	3.40	1.91	4.48	1.65	4.34	1.96

Experiments: Challenging Dataset - TinyImageNet

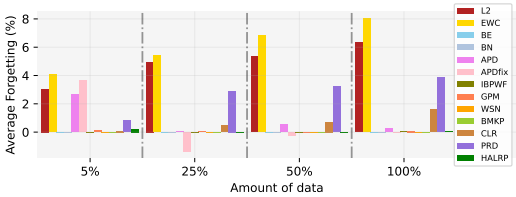
	TinyImageNet 20-split						TinyImageNet 40-split					
	AlexNet			ResNet18			AlexNet			ResNet18		
Method	Acc.↑	MODP↓	AODP↓	Acc.↑	MODP↓	AODP↓	Acc.↑	MODP↓	AODP↓	Acc.↑	MODP↓	AODP↓
STL	66.78±0.18	6.20	3.43	67.00±0.30	5.87	2.87	74.65±0.17	8.16	4.36	74.08±0.25	8.13	4.08
MTL	71.23±0.54	6.87	3.42	73.64±0.46	6.94	3.31	78.80±0.25	7.74	3.92	80.05±0.22	9.07	4.04
L2	56.33±0.22	11.93	5.72	60.80±0.56	8.27	4.23	63.47±1.35	16.27	7.42	65.59±1.03	14.80	7.00
EWC	56.55±0.22	10.93	5.53	60.88±0.56	7.54	4.63	64.11±1.36	17.87	6.91	66.54±0.94	14.67	6.99
BN	57.20±0.11	12.07	5.37	61.03±0.65	9.73	4.24	64.04±1.15	19.07	7.23	66.17±1.75	15.33	6.92
BE	57.62±0.41	11.80	4.98	61.52±0.68	7.00	4.09	64.70±1.08	23.47	6.97	66.77±1.61	16.94	7.07
APD	**67.26±0.38**	12.00	5.43	68.76±0.58	9.86	4.33	73.88±0.46	8.53	5.04	73.85±0.40	11.46	5.57
APDfix	63.59±0.25	5.40	3.68	67.69±0.41	6.54	3.64	67.91±1.33	23.47	8.95	58.11±2.11	26.93	11.70
GPM	60.84±0.32	11.00	4.44	48.09±1.07	9.27	4.80	70.14±0.38	10.90	4.52	43.40±7.68	48.80	38.80
WSN	65.73±0.21	5.06	3.31	68.27±0.47	8.40	4.52	73.46±1.27	8.00	4.23	75.29±0.46	12.27	4.75
BMKP	65.01±0.41	5.87	3.37	67.45±0.59	10.60	6.69	73.57±0.21	9.60	4.31	74.84±0.63	12.90	4.95
CLR	57.29±0.39	8.80	4.42	61.77±0.47	10.27	5.42	65.22±0.49	9.86	5.74	67.59±0.92	16.54	9.30
PRD	46.49±0.30	15.40	8.33	49.65±0.57	19.33	11.59	53.54±0.45	19.74	10.02	63.78±0.59	20.27	7.93
HALRP	66.68±0.20	5.60	3.43	**70.09±0.29**	4.67	3.20	**74.01±0.25**	7.47	4.12	**75.53±0.50**	7.87	4.48

Table: Accuracy (↑) and task order robustness (MOPD ↓ and AOPD ↓) on TinyImageNet 20-split and 40-split.

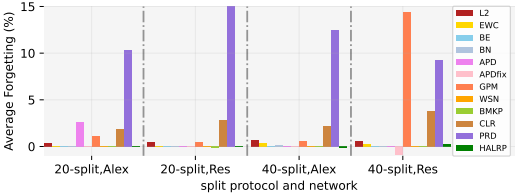
Experiments: Forgetting/Model Increment/Efficiency



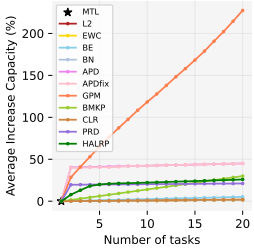
(a) Forgetting on CIFAR100-Splits (10 tasks)



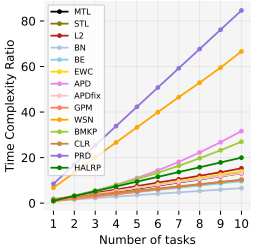
(b) Forgetting on CIFAR100-SuperClass (20 tasks)



(c) Forgetting on Tiny-ImageNet.

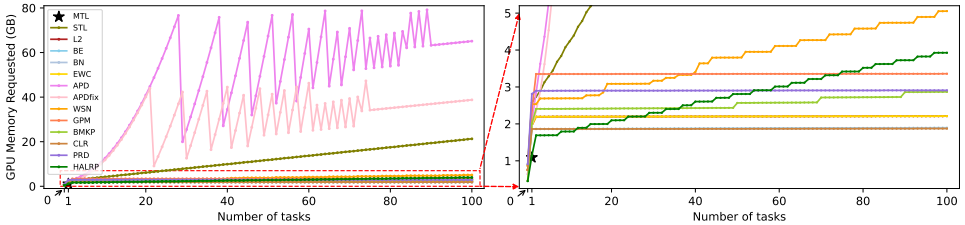


(d) Model Increment.

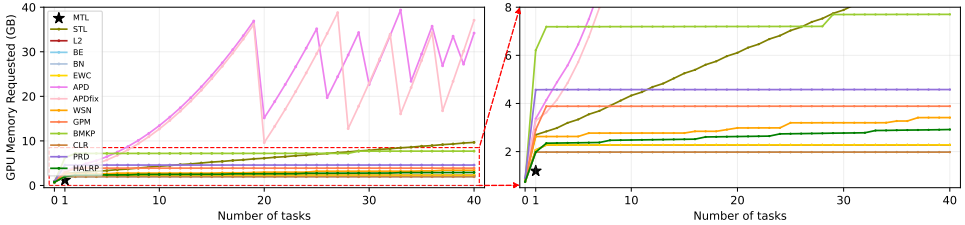


(e) Time Complexity

Experiments: Memory Overhead Comparison



(a) GPU memory tracking for OmniglotRotation+LeNet (100 tasks)



(b) GPU memory tracking for TinyImageNet+AlexNet (40 tasks)

References:

[1] J. Li et al. "Hessian Aware Low-Rank Perturbation for Order-Robust Continual Learning". In: *IEEE Transactions on Knowledge and Data Engineering* (2024), pp. 1–12. ISSN: 1558-2191. DOI: 10.1109/TKDE.2024.3419449.

Thank you!