MVP Lab

Continual Forgetting for Pre-trained Vision Models

Hongbo Zhao1,3* Bolin Ni1,3* Junsong Fan2 Haochen Wang1,3 Yuxi Wang2Fei Zhu2 Yuntao Chen2 Gaofeng Meng1,2,3† Zhaoxiang Zhang1,2,3,4†

Motivation

Challenges:

(i) For unwanted knowledge, efficient and effective deleting is crucial.

(ii) For remaining knowledge, the impact brought by the forgetting procedure should be minimal.

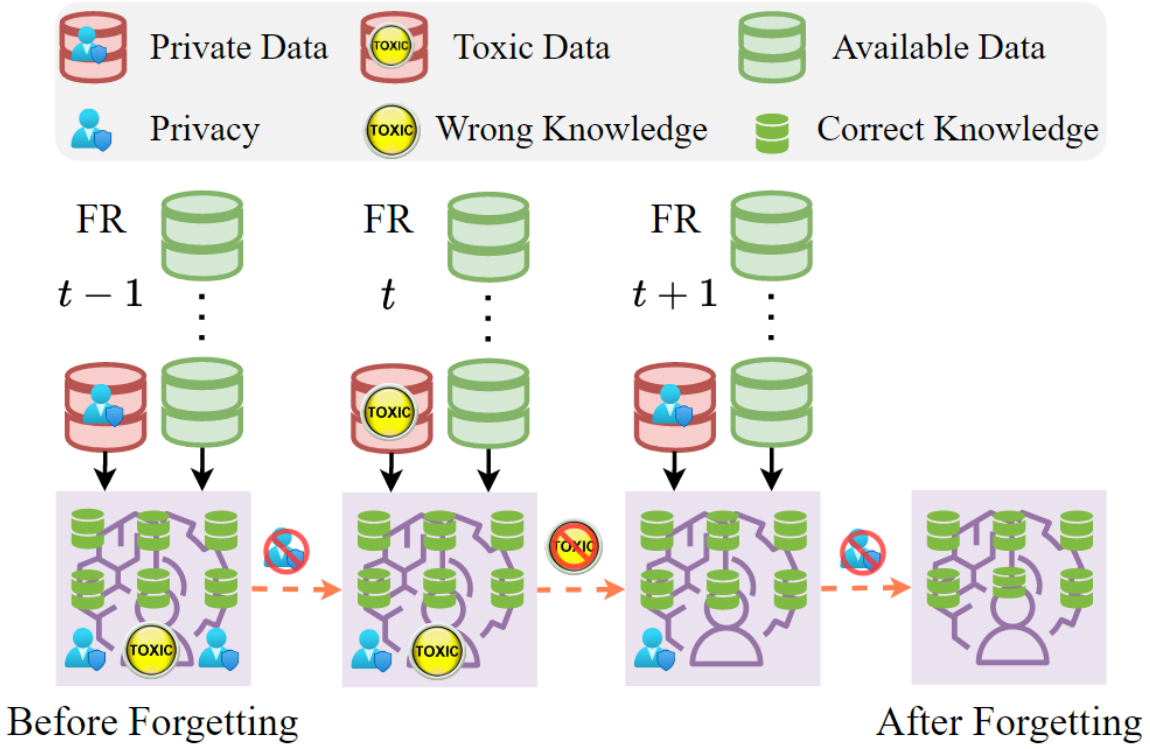
solutions:

(i) this paper use LoRA modules to fine-tune the FFN layers in Transformer blocks for each forgetting task independently, and towards

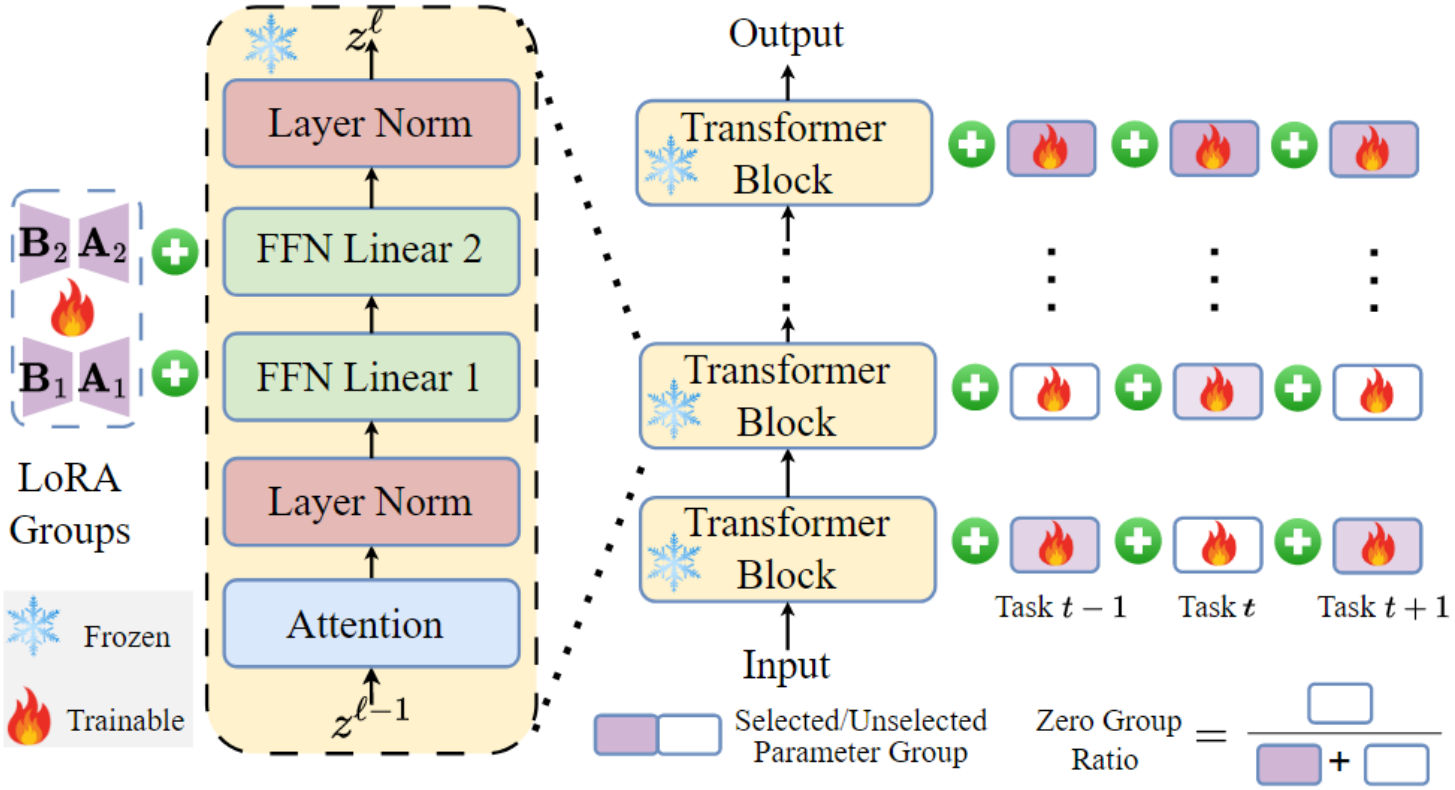
(ii) a simple group sparse regularization is adopted, enabling automatic selection of specific LoRA groups and zeroing out the others.

Existing machine unlearning methods:

- For typical ML algorithms:
  - Limited application scenarios
- For deep learning methods:
  - Computationally heavy
  - Effective on small-scale problems only
  - Specific designs in the pre-training process



Overview



This paper incorporates a set of LoRA modules in each continual forgetting task and adopt a sparse structure selection strategy to achieve accurate and few modifications. All LoRA modules are added in the Linear layers of FFN in the Transformer blocks and we regard the LoRA modules in a Transformer block as one group. We use group sparse regularization to automatically select LoRA groups. The purple groups are selected to modify and the white groups are neglected. The pretrained model (including Transformer blocks and other parts) is frozen and only LoRA groups are trainable.

LoRA Based Model Tuning

To reduce the learnable parameters, we incorporate a set of LoRA modules to the FFN in each Transformer block and only make these LoRA modules trainable. Suppose x is the input of the ℓ-th FFN module, the mathematical form can be expressed as:

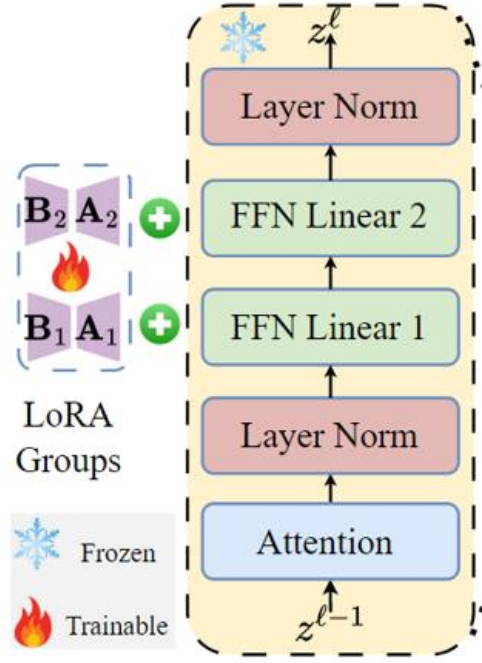
$$\mathcal{FFN}^{(\ell)}(\mathbf{x}) = \max\left(\mathbf{0}, \mathbf{x}\mathbf{W}_1^{(\ell)} + \mathbf{b}_1^{(\ell)}\right)\mathbf{W}_2^{(\ell)} + \mathbf{b}_2^{(\ell)}$$

where W(ℓ) 1 , W(ℓ) 2 , b(ℓ) 1 , b(ℓ) 2 are the weights and biases of two fully connected layers from the pre-trained model, respectively. We use LoRA to only fine-tune the weights of FFN modules:

$$\mathbf{W}_t^{(\ell)} = \begin{bmatrix} \mathbf{W}_{1t}^{(\ell)} \\ \mathbf{W}_{2t}^{(\ell)} \end{bmatrix} = \begin{bmatrix} \mathbf{W}_1^{(\ell)} \\ \mathbf{W}_2^{(\ell)} \end{bmatrix} + \sum_{i=1}^{t} \mathbf{B}_i^{(\ell)} \mathbf{A}_i^{(\ell)},$$

$$\mathbf{B}_i^{(\ell)} = \begin{bmatrix} \mathbf{B}_{1i}^{(\ell)} & \mathbf{O} \\ \mathbf{O} & \mathbf{B}_{2i}^{(\ell)} \end{bmatrix}, \quad \mathbf{A}_i^{(\ell)} = \begin{bmatrix} \mathbf{A}_{1i}^{(\ell)} \\ \mathbf{A}_{2i}^{(\ell)} \end{bmatrix},$$

i = 1, 2, · · · , t refer to the corresponding LoRA matrices in task Ti



Group Sparsity Selection

To mitigate catastrophic forgetting and achieve precise modifications automatically, this paper introduce a group sparsity selection strategy that enables the selection of fewer Transformer blocks. Suppose LoRA matrices added to the ℓ-th Transformer block in task Tt areB1(ℓ)t , A1(ℓ)t , B2(ℓ)t , A2(ℓ)t . Then the optimization goal with group sparse regularization can be expressed as follows:

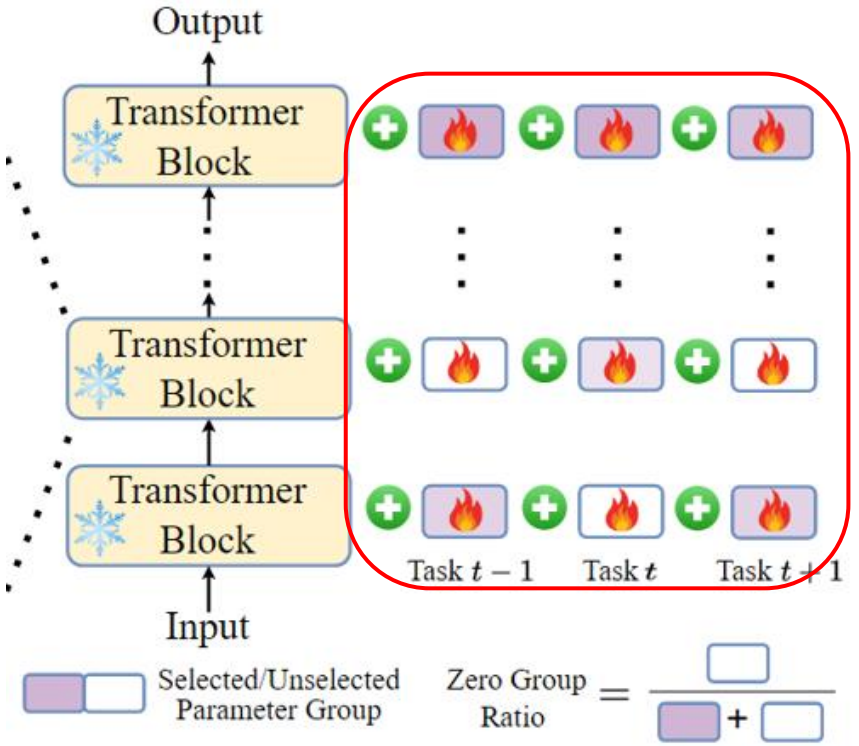
$$\mathcal{L}_{total} = \mathcal{L}_{data} + \alpha \mathcal{L}_{strcuture}.$$

Ldata denotes the loss on data, Lstructure is the group sparse loss, and α serves as a hyperparameter to regulate the sparse intensity.

The group sparse loss on a set of weights can be represented as:

$$\mathcal{L}_{structure} = \sum_{\ell=1}^{G} \mathcal{L}_{gs}^{(\ell)} \qquad \mathcal{L}_{gs}^{(l)} = \|\mathbf{B}_t^{(\ell)}\|_F + \|\mathbf{A}_t^{(\ell)}\|_F$$

where G is the number of groups, L(l)gs is the group sparse loss of the ℓ-th group.



Single-step Experiments face recognition task

this paper constructed a subdataset called CASIA-Face100 which collects 100 face IDs from the CASIA-WebFace [84] dataset. this paper use a Face Transformer [90] pre-trained on the CASIA-Face100 dataset. For the object detection task, this paper use a deformable DETR [96] pre-trained on the COCO 2017 [44] dataset.

Methods	Tunable Ratio ↓	100-5			100-10			100-50			100-90		
		H ↑	Acc_r ↑	Acc_f ↓	H ↑	Acc_r ↑	Acc_f ↓	H ↑	Acc_r ↑	Acc_f ↓	H ↑	Acc_r ↑	Acc_f ↓
Pre-train	-	-	70.2	74.5	-	74.4	73.8	-	74.8	74.0	-	73.8	74.6
L2*	99.73%	67.7	67.8	2.6	67.0	65.0	4.5	63.4	55.3	0.6	53.8	42.2	0.2
EWC* [37]	99.73%	69.0	68.8	1.0	69.2	67.4	2.6	60.9	51.4	0.1	46.3	33.6	0.2
MAS* [2]	99.73%	68.5	69.2	2.4	68.5	66.7	3.4	59.9	50.0	0.1	42.8	30.0	0.1
LwF [20]	99.73%	67.0	67.0	3.1	68.2	65.1	2.1	64.0	56.1	0.3	50.6	38.4	0.2
DER [8]	99.73%	66.4	67.4	4.8	67.9	67.2	5.1	61.3	52.0	0.3	54.0	42.5	0.2
DER++ [8]	99.73%	67.1	66.9	2.9	68.6	67.3	3.9	63.0	54.8	0.6	64.3	57.0	0.6
FDR [4]	99.73%	67.2	69.5	5.0	68.5	67.4	4.2	65.9	59.3	0.5	55.8	44.9	0.5
SCRUB [38]	99.73%	67.0	65.5	1.7	69.2	66.5	1.7	0.0	0.0	0.0	18.2	10.4	**0.0**
SCRUB-S [38]	99.73%	68.6	**71.8**	4.5	68.9	**71.9**	7.7	54.8	63.1	26.4	19.0	10.9	**0.0**
LIRF* [83]	50.66%	28.7	62.6	51.6	26.3	63.3	57.2	46.1	54.2	34.7	46.9	34.9	2.7
Retrain	100.00%	13.2	7.3	**0.0**	16.2	9.1	**0.7**	13.2	7.3	**0.0**	9.5	5.1	**0.0**
GS-LoRA	**1.28%**	**69.3**	70.5	1.9	**71.4**	71.1	2.0	**71.9**	**69.9**	0.8	**72.2**	**70.5**	0.5

Table 1. **Single-step forgetting results for face recognition.** Acc_r and Acc_f are the accuracies of remaining and forgotten classes. * denotes the original methods with a rehearsal buffer. Note that "retrain" represents retraining the model using replay data and *the training epoch is the same as other methods to ensure a fair comparison.* Pre-train denotes the results before forgetting. All setting is in the form of 100-Y, which means all experiments start from a pre-trained model (100 classes originally) and forget Y classes.

Single-step Experiments object detection task

For the object detection task, we use a deformable DETR [96] pre-trained on the COCO 2017 [44] dataset.

Methods	Tunable Ratio ↓	80-1			80-5			80-40			80-70		
		H ↑	AP_r ↑	AP_f ↓	H ↑	AP_r ↑	AP_f ↓	H ↑	AP_r ↑	AP_f ↓	H ↑	AP_r ↑	AP_f ↓
Pre-train	-	-	44.3	57.1	-	44.8	41.3	-	44.8	44.6	-	45.0	44.6
L2*	99.61%	25.6	35.6	37.1	27.7	34.9	18.4	27.9	32.3	20.0	29.9	34.9	18.4
EWC* [37]	99.61%	37.6	32.6	12.5	31.7	33.4	11.2	33.0	31.7	10.2	33.6	29.5	5.7
MAS* [2]	99.61%	39.4	32.5	6.9	27.9	30.4	15.4	31.1	30.4	12.8	31.6	28.6	9.3
Retrain	100.00%	46.6	39.3	**0.0**	40.3	39.6	**0.2**	40.5	39.2	**2.6**	37.8	39.7	8.6
GS-LoRA	**0.62%**	**49.9**	**44.5**	0.4	**42.4**	**45.0**	1.2	**41.6**	**42.8**	4.1	**43.7**	**43.6**	**0.9**

Table 2. **Single-step forgetting results for object detection** on the COCO dataset. AP_r and AP_f denotes the AP of remaining classes and forgotten classes. All setting is in the form of 80-Y, which means all experiments start from a pre-trained model and forget Y classes.

Tabs. 1 and 2 show the performance comparisons with the aforementioned baselines for single-task forgetting, the degraded scenario in continual forgetting. The proposed GS-LoRA performs poorly in forgotten classes while retaining approximately the original performance in preserved classes. It is effective whether forgetting a small number of classes (e.g., 1 class), or a large number of classes (e.g., 90% of all the classes).

continual Experiments face recognition task

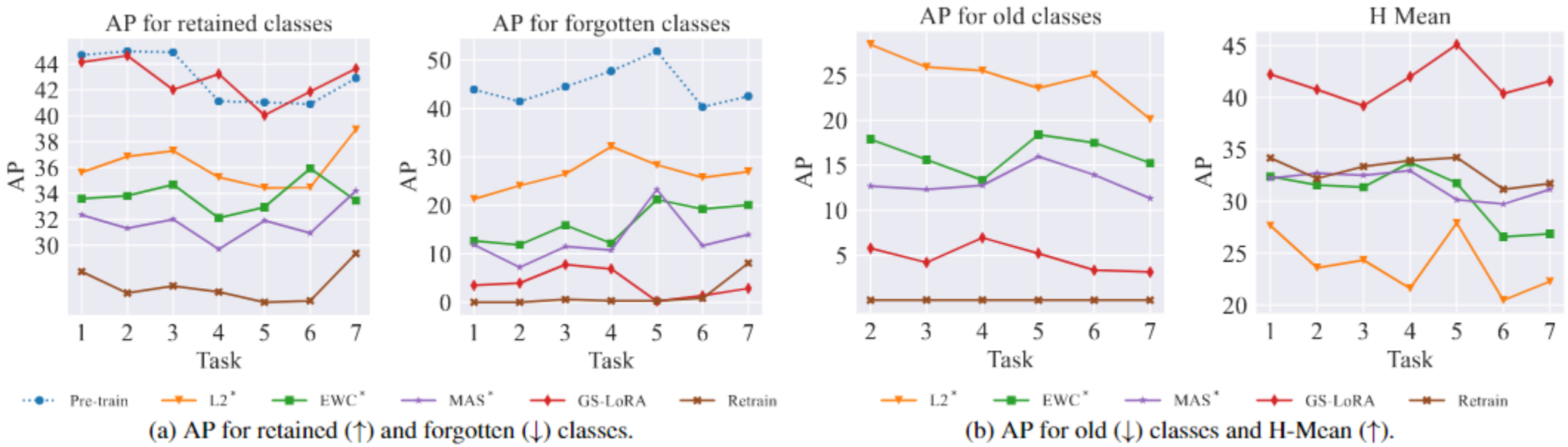
Methods	100-20			80-20				60-20				40-20			
	$H\uparrow$	$Acc_r\uparrow$	$Acc_f\downarrow$	$H\uparrow$	$Acc_r\uparrow$	$Acc_f\downarrow$	$Acc_o\downarrow$	$H\uparrow$	$Acc_r\uparrow$	$Acc_f\downarrow$	$Acc_o\downarrow$	$H\uparrow$	$Acc_r\uparrow$	$Acc_f\downarrow$	$Acc_o\downarrow$
Pre-train	-	74.6	74.6	-	72.9	70.9	-	-	71.9	69.7	-	-	72.7	71.3	-
L2*	66.7	61.9	2.3	63.2	60.9	5.1	9.4	63.8	60.3	2.2	10.1	62.3	56.7	2.2	6.8
EWC* [37]	66.9	61.0	0.4	66.0	62.9	1.5	**0.0**	66.2	63.9	1.2	**0.0**	64.8	59.7	0.5	**0.0**
MAS*[2]	66.6	60.7	0.7	65.4	61.8	1.6	**0.0**	66.1	63.5	0.8	**0.0**	64.2	58.6	**0.3**	**0.0**
LwF [20]	66.2	60.9	2.1	64.6	60.8	2.1	0.5	64.9	61.4	1.4	**0.0**	65.0	60.7	1.4	**0.0**
DER [8]	66.7	62.7	3.4	63.3	59.8	3.7	**0.0**	63.8	60.2	2.0	**0.0**	62.7	57.2	2.0	**0.0**
DER++ [8]	66.1	62.8	4.8	63.8	61.7	5.0	**0.0**	64.4	61.6	2.4	**0.0**	65.0	61.8	2.7	**0.0**
FDR [4]	64.4	59.3	4.1	62.2	58.0	3.9	**0.0**	65.0	62.8	2.4	**0.0**	65.7	62.6	2.3	**0.0**
SCRUB [38]	67.8	63.1	1.3	66.3	64.4	2.6	**0.0**	66.7	64.5	0.8	**0.0**	68.4	66.9	1.5	**0.0**
SCRUB-S [38]	71.2	69.6	1.7	**69.1**	70.4	3.0	9.0	**70.4**	71.9	0.8	6.4	70.1	70.1	1.1	2.3
LIRF* [83]	28.6	60.1	55.8	28.3	58.5	52.2	43.4	35.8	56.1	43.5	33.5	36.8	59.8	44.7	22.1
Retrain	18.4	10.5	**0.3**	16.0	9.1	0.8	**0.0**	16.9	9.6	**0.0**	**0.0**	23.4	14.0	0.5	**0.0**
GS-LoRA	**71.6**	**72.1**	3.5	68.4	**71.1**	4.9	**0.0**	69.7	**72.0**	2.2	**0.0**	**70.2**	**71.0**	1.8	**0.0**

Table 3. **Continual forgetting results for face recognition.** Acc_o is the accuracy of old tasks, *i.e.*, the accuracy on all previously forgotten classes in task $\mathcal{T}_1, \mathcal{T}_2, \cdots, \mathcal{T}_{t-1}$. There are 4 tasks in total and 20 classes are forgotten in each task.

H-Mean evaluates the overall performance after learning task Tt

$$Drop^{(t)} = Acc_f^{(t-1)} - Acc_f^{(t)} \qquad H\text{-}Mean^{(t)} = \frac{2Acc_r^{(t)} \cdot Drop^{(t)}}{Acc_r^{(t)} + Drop^{(t)}}.$$

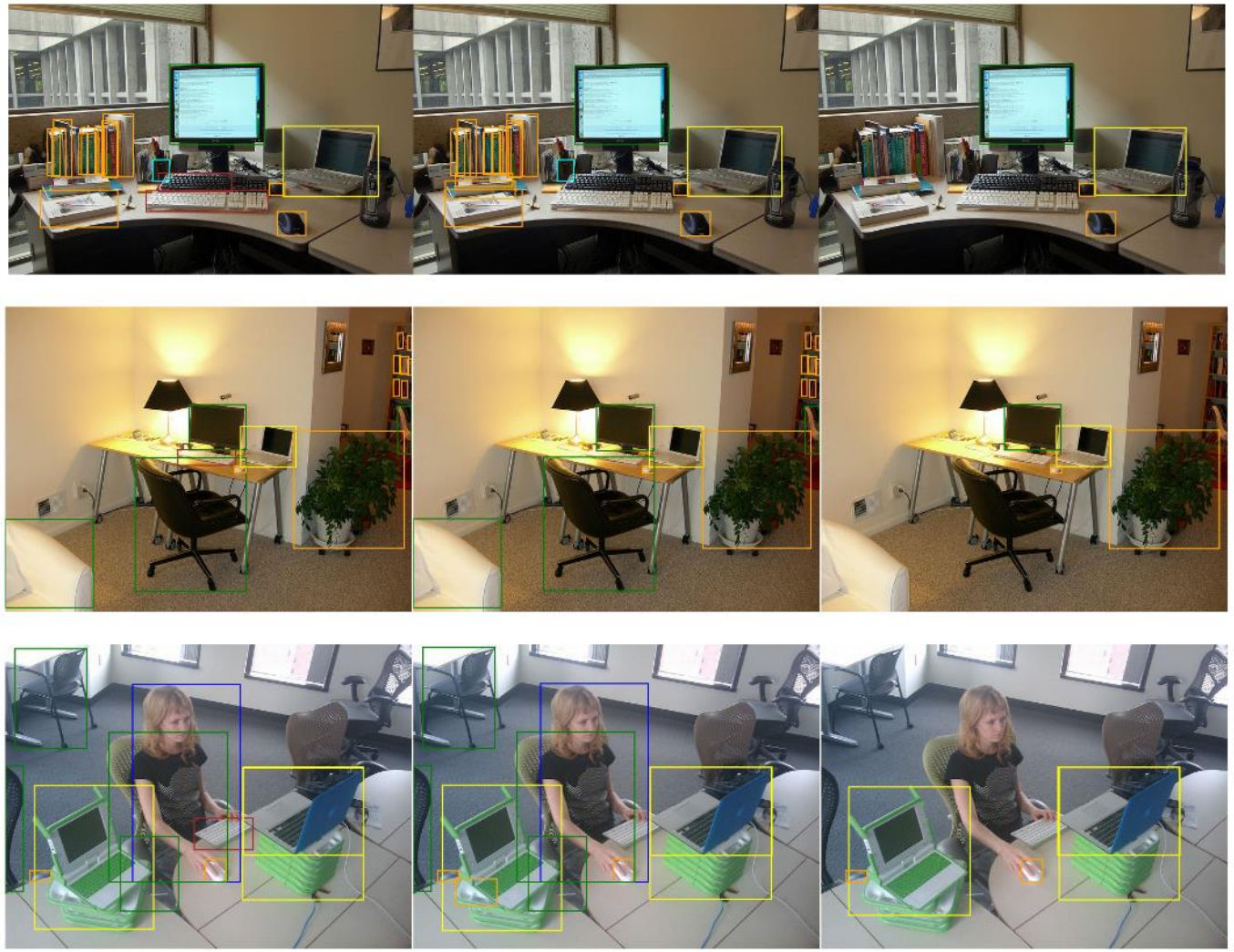
continual Experiments object detection task



(a) AP for retained (↑) and forgotten (↓) classes.

(b) AP for old (↓) classes and H-Mean (↑).

Fig. 3 and Tab. 3 show the results for continual forgetting. For the object detection tasks, 10 classes are forgotten per task (7 tasks in total), while for the classification task, 20 classes are forgotten per task (4 tasks in total). GS-LoRA works the best among the listed methods, especially on object detection tasks. Besides, we can observe that in such a fast modification setting, severe underfitting occurs when using the retraining method.



Visualization for continual forgetting for object detection task. The left column shows the results from the pre-trained model. The middle column shows the results when "keyboard" (red bounding boxes in the left column) is erased. The left column shows the results when more objects (e.g., person, book, chair) are erased.

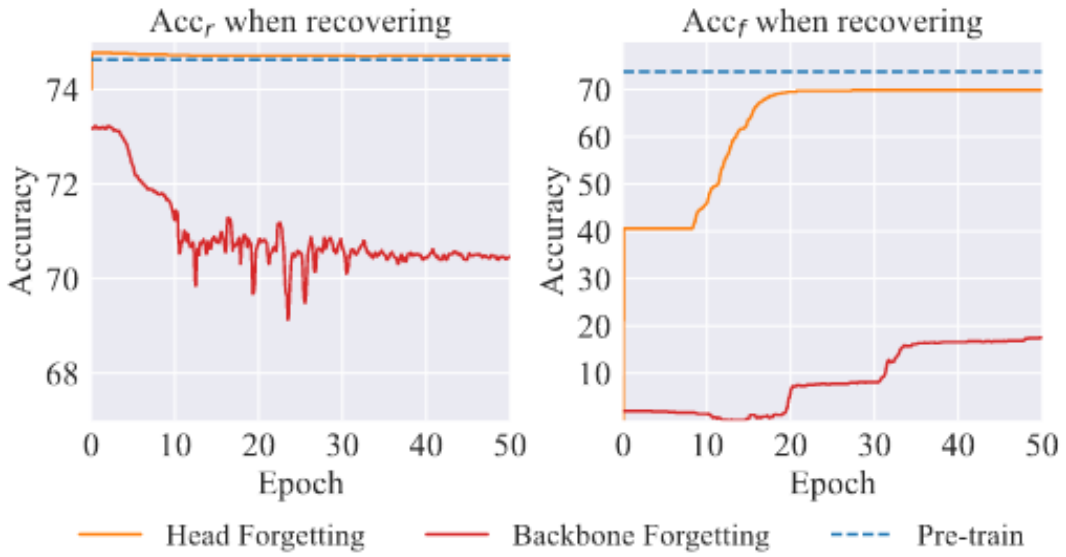


Figure 6. **Accuracy on forgotten classes and retained classes when recovering.** The blue line (Pre-train) is the result before forgetting. The orange line (Head Forgetting) is the trivial masking method. The red line (Backbone Forgetting) is the GS-LoRA.

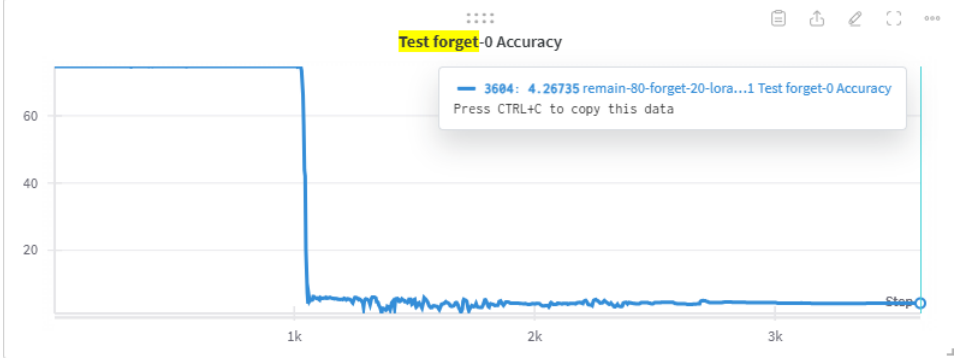
Figure shows the classification accuracy curve with epoch when recovering. Compared to head forgetting, we can find that the model after forgetting via GS-LoRA can only be recovered to approximately 17% on forgotten classes, significantly lower than 70% achieved in head forgetting.

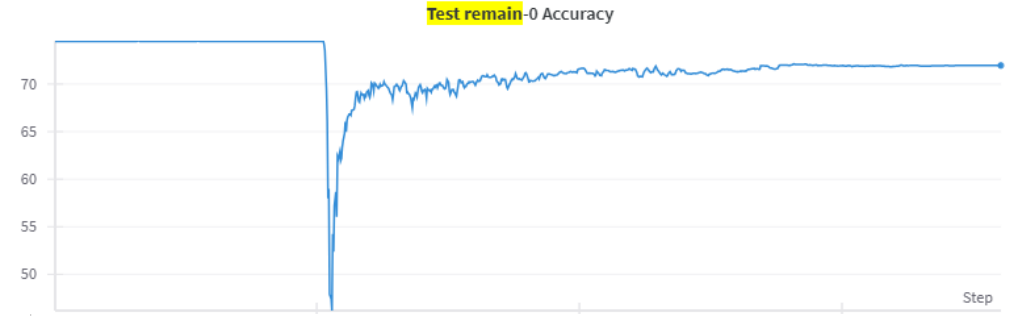
That's why this paper freezes the Transformer blocks, especially the output FFN layers. so, we can ensure forgetting occurs in the backbone and is difficult to recover.

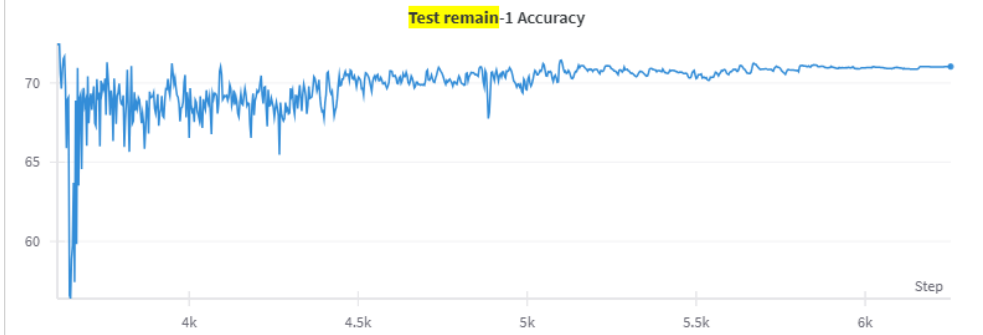
Although it is possible to recover the accuracy of forgotten classes to a very low level in backbone forgetting, such recovery adversely impacts the accuracy of the remaining classes. Additionally, the recovery process for GS-LoRA needs more epochs while head forgetting can be recovered within 20 training epochs.

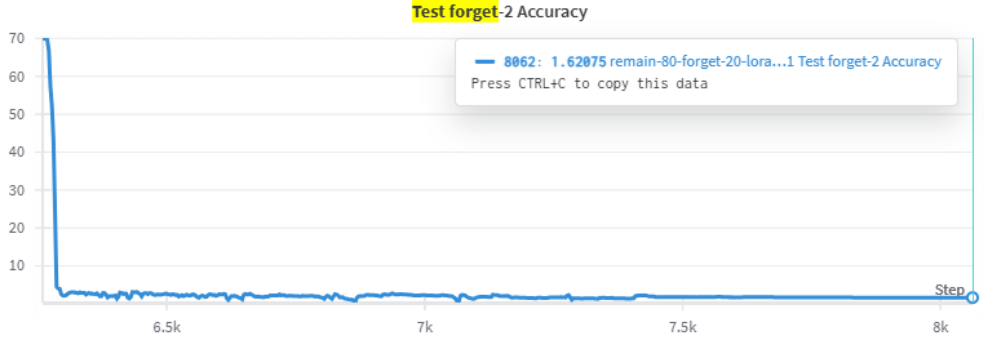
复现

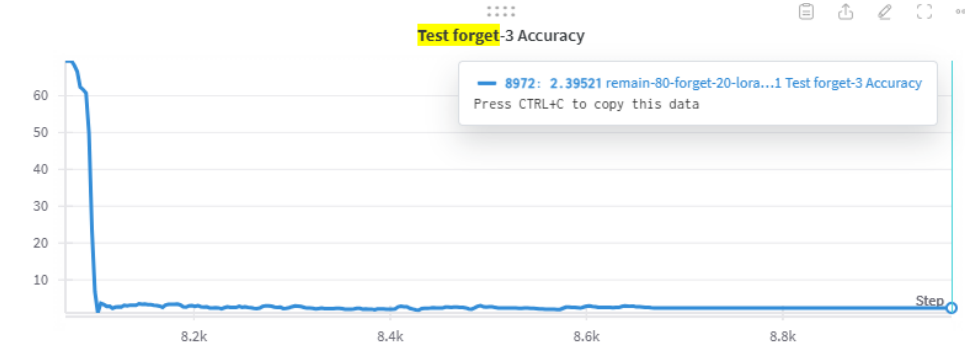
连续遗忘
从100类中
忘20，4次











	4步连续															
	100-20			80-20				60-20				40-20				
	H	Accr	Accf	H	Accr	Accf	Acco	H	Accr	Accf	Acco	H	Accr	Accf	Acco	
per-train		74.6	74.6		72.9	70.9			71.9	69.7			72.7	71.3		
gslora	**71.13078**	**71.95**	**4.27**	**68.33121**	**71.03**	**5.07**	**0**	**69.70049**	**71.4**	**1.62**	**0.03**	**70.03112**	**71.2**	**2.40**	**0**	
ewc	66.44092	61.16	1.88	64.75491	61.31	2.29	0	66.56245	64.11	0.49	0.03	64.98798	60.29	0.82	0.02	
l2	66.32385	62.05	3.37	63.33221	60.28	4.19	11.26	62.64289	58.01	1.62	8.6	61.76135	55.06	0.98	5.96	
der	65.90196	62.65	5.09	63.61608	61.59	5.12	0	64.29833	61.9	2.81	0	63.30399	58.27	2.01	0	
lwf	66.32352	61.31	2.37	64.64219	60.88	2	0.67	65.31789	62.41	1.19	0.03	65.1771	61.19	1.58	0	
GS-LoRA	**71.6**	**72.1**	3.5	68.4	**71.1**	4.9	**0.0**	69.7	**72.0**	2.2	**0.0**	**70.2**	**71.0**	1.8	**0.0**	

	100-5			100-10				100-20		
	H	Accr	Accf	H	Accr	Accf		H	Accr	Accf
per-train		70.2	74.5		74.4	73.8			74.6	74.6
gslora	**70.78624**	**73.74**	**6.44**	**69.04748**	**74.1**	**6.26**		**71.13078**	**71.95**	**4.27**
SCRUB								70.82603	70.3	3.24
LIRF								26.85631	65.97	57.74
l2								8.788504	74.42	69.93