PROL : Rehearsal Free Continual Learning in Streaming Data via Prompt Online Learning

M. Anwar Ma'sum[1,*], Mahardhika Pratama[1], Savitha Ramasamy[2], Lin Liu[1], Habibullah Habibullah[1], and Ryszard Kowalczyk[1,3]

[1]STEM University of South Australia, [2]Institute for Infocomm Research, A*STAR & IPAL, CNRS@CREATE

[3]Systems Research Institute, Polish Academy of Sciences, Warsaw, Poland

Abstract

This document presents supplementary material for PROL: Rehearsal Free Continual Learning in Streaming Data via Prompt Online Learning. This supplementary consists of 3 parts i.e. PROL algorithm that is presented in section A, PROL complexity analysis that is presented in section B, detailed setting as presented in section C, and complete numerical result that is presented in section D.

A. PROL Algorithm

This section presents the PROL training and inference algorithm that is presented in algorithm 1 and 2, respectively.

B. PROL Complexity Analysis

This section presents the complexity analysis of our proposed method. Following algorithm 1, the complexity of PROL can be formulated as:

$$O(PROL) = \sum_{t=1}^{T} \sum_{i=1}^{S^t} O(per - stream) \tag{A1}$$

where S^t is the number of streams in task t. Given a stream s on task t, PROL executes several operations as presented in lines (5) to (22). All the operations are O(1). Thus PROL complexity can be derived into

$$O(PROL) = \sum_{t=1}^{T} \sum_{i=1}^{S^t} O(|s|) \tag{A2}$$

Where $|s|$ represents the stream's size. Since $\sum_{i=1}^{S^t} |s| = |\mathcal{T}^t|$ then the PROL complexity becomes

$$O(PROL) = \sum_{t=1}^{T} O(|\mathcal{T}|) \tag{A3}$$

Since $\sum_{t=1}^{T} |\mathcal{T}| = |\mathcal{T}| = N$ then the PROL complexity summarized as

$$O(PROL) = O(N) \tag{A4}$$

C. Detailed experimental setting

We evaluate our method in 4 of the most popular datasets in CL i.e. CIFAR100[5], ImageNet-R[7], ImageNet-A[6], and CUB[13]. CIFAR100 contains 100 classes images, while the other has 200 classes. We uniformly split the dataset into 10 tasks. We compare our proposed method with the current SOTAs of PTM-based OCL i.e. RanDumb[9], RanPAC[8], and MOS[12], the extension of EASE[18]. We adapt the official code of RanDumb and RanPAC from [9] into our rehearsal free and disjoint setting. We also adapt the rehearsal version (denoted with -R) with 5 samples per class as memory, and the joint version (denoted with -J) for both of them. We adapted MOS from its official code into rehearsal free setting. We add L2P [16], DualPrompt[15], and ConvPrompt[11] as representation of pool-based, task-specific, and growing components prompting, respectively. To ensure fairness, all the consolidated methods are run with the same dataset split, backbone i.e. ViT-B/16 pre-trained on Imagenet-21K. We also compare our method to non-PTM methods i.e. ER [10] DER++[1], ERASE [2, 3], GSA[4], ONPRO [17], and ERSM[14]. The compared result is taken from [14]. The learning rates for all methods are set by grid search from the range of [0.001,0.005,0.01,0.05,0.1]. All the methods are run under the same environment, i.e., a single NVIDIA A100 GPU with 40 GB RAM by three different random seed numbers. We utilize Adam optimizer for PROL, while the rest follow their official implementation. We evaluate final average accuracy (FAA), cumulative average accuracy (CAA), and

*Corresponding: masmy039@mymail.unisa.edu.au

Algorithm 1 PROL Training

Input: A sequence of tasks $\mathcal{T}^1$, $\mathcal{T}^2$,...,$\mathcal{T}^T$, a frozen pre-trained ViT $f_{\theta(.)}$,
Initiate trainable generator (G_K, G_V) and MLP head g_ϕ
for $t = 1 : T$ **do**
  stream data $s = \{(x_i, yi)\}$ from $\mathcal{T}^t$
  initiate (K^c, a^c, b^c) for any all $c \in s$ if not exist before
  if $t > 1$ **then**
    Freeze (G_K, G_V)
  end if
  Generate prompt (P_K, P_V) following eq. (2)
  Compute PTM only and PTMP+prompt prototypes: $f_\theta(x)$, and $f_{\theta;P}(x)$
  Compute logits: $g_\phi(f_{\theta;P}(x))$
  Compute $\mathcal{L}_{intra}$ following eq. (4)
  Compute $\mathcal{L}_{inter}$ following eq. (5)
  Compute $\mathcal{L}_{sim}$ following eq. (6)
  Compute $\mathcal{L}_{ort}$ following eq. (7)
  Generate M and Compute $\mathcal{L}_{gen}$ as in eq. (8)
  Compute $\mathcal{L}_{total}$ following eq. (9)
  if $t = 1$ **then**
    Update (G_K, G_V) parameters
  end if
  update (K^c, a^c, b^c) and ϕ
  Clamp (a^c, b^c) following eq.(1)
end for
Output: Optimum generator (G_K, G_V), parameters (K^c, a^c, b^c) and ϕ

Algorithm 2 PROL Inference

Input: An input x, a frozen pre-trained ViT $f_{\theta(.)}$, optimized generator (G_K, G_V) parameters (K^c, a^c, b^c) and ϕ
Find top-1 K^c where $c \in \mathcal{T}$
Generate prompt (P_K, P_V) following eq. (2)
Compute logits by forwarding the input i.e. $logits = g_\phi(f_{\theta;P}(x))$
Compute Pedicted label $\hat{y} = argmax(logits)$
Output: Predicted label $\hat{y}$

final forgetting measure (FM); please see the supplementary document for the details.

The λ_1, λ_3 and λ_4 are set to 1.0, while λ_2 is set to 0.001 for CIFAR100, and 0.01 for CUB and 0.03 for ImageNet-R and ImageNet-A. The $\mathcal{L}_{thres}$ is set to 0.3 for CIFAR100 and CUB, and 0.8 for ImageNet-R and ImageNet-A. The cosine annealing scheduler is set to maxT=20 and min-lr=0.005. All the consolidated methods are run under the same machine and computing environment i.e. single NVIDIA A100 GPU with 40 GB memory, python 3.9 and Pytorch 2.2.0.

Performance Metrics: Adapted from HidePrompt, we measure both accuracy and forgetting of the methods. Suppose that $A_{i,t}$ denotes the accuracy on the t-th task after learning the t-th task. The average accuracy of all learned task is defined as $AA_t = (1/t)\Sigma_{i=1}^{t} A_{i,t}$. Suppose that T is the number of all tasks, we measure final average accuracy (FAA), cumulative average accuracy(CAA), and final forgetting measure (FM)

$$FAA = AA_T \tag{A5}$$

$$CAA = \frac{1}{T}\Sigma_{t=1}^{T} AA_t \tag{A6}$$

$$FFM = \frac{1}{T-1}\Sigma_{i=1}^{T-1} max_{t\in 1,...,T-1}(AA_{i,t} - AA_{i,T}) \tag{A7}$$

D. Detailed numerical performance

This section presents detailed numerical results of the consolidated algorithms both accuracy and forgetting, throughput, training time, and inference time as presented in tables A1 to A11

Table A1. Detailed Accuracy performance of consolidated methods in CIFAR100

Method	1	2	3	4	5	6	7	8	9	10	AVG
RanPAC-J	97.40	93.32	89.10	86.91	84.70	83.40	82.18	81.17	80.94	79.94	85.91
RanDumb-J	97.67	92.97	89.47	87.08	85.03	83.45	82.57	81.52	81.32	80.39	86.15
RanPAC-R	97.40	78.42	71.13	70.51	65.19	64.07	63.59	60.92	62.59	59.66	69.35
RanDumb-R	97.90	72.42	63.62	60.08	56.03	55.61	56.48	53.80	54.55	50.93	62.14
RanPAC	97.37	47.57	31.01	23.58	18.75	15.74	13.28	11.75	10.64	9.52	27.92
RanDumb	97.67	47.60	30.86	23.63	18.85	15.53	13.30	11.78	10.62	9.56	27.94
L2P	97.10	92.30	88.87	87.40	86.00	84.10	83.93	83.75	82.14	80.99	86.66
DualPrompt	98.50	94.55	91.93	88.78	86.66	85.40	84.40	84.04	83.80	82.53	88.06
ConvPrompt	98.50	92.50	90.27	88.28	87.18	85.27	84.83	84.81	85.44	84.79	88.19
MOS	98.00	96.05	92.97	90.98	88.98	88.15	87.60	84.51	84.06	83.99	89.53
PROL	99.30	96.20	94.40	92.73	91.64	88.92	88.07	88.00	87.90	86.32	91.35

Table A2. Detailed Forgetting of consolidated methods in CIFAR100

Method	1	2	3	4	5	6	7	8	9	10	AVG
RanPAC-J	-	3.27	3.90	4.36	5.03	5.57	5.87	6.01	6.14	6.43	5.17
RanDumb-J	-	4.00	3.83	4.52	5.15	5.83	5.64	5.80	6.01	6.25	5.23
RanPAC-R	-	33.00	32.73	29.93	32.44	32.61	31.57	33.62	32.03	34.30	32.47
RanDumb-R	-	47.93	47.08	46.13	47.87	46.06	43.27	45.34	43.94	47.35	46.11
RanPAC	-	97.37	96.25	95.18	94.96	94.71	94.67	94.42	94.37	94.52	95.16
RanDumb	-	97.63	96.42	95.13	94.94	94.81	94.54	94.37	94.38	94.51	95.19
L2P	-	4.50	5.25	3.70	4.35	4.76	4.58	4.77	6.35	6.71	5.00
DualPrompt	-	2.00	2.05	2.50	4.30	3.78	3.95	3.76	3.54	4.53	3.38
ConvPrompt	-	0.00	1.50	1.33	1.98	2.10	2.73	2.61	2.19	2.38	1.87
MOS	-	1.20	5.75	6.60	8.23	8.50	8.95	11.66	11.60	11.34	8.20
PROL	-	2.00	2.75	3.97	4.60	4.74	4.83	4.73	5.04	6.34	4.33

Table A3. Detailed Accuracy of consolidated methods in ImageNet-R

Method	1	2	3	4	5	6	7	8	9	10	AVG
RanPAC-J	66.72	60.66	59.82	56.40	55.21	53.29	52.59	52.26	53.08	52.09	56.21
RanDumb-J	65.12	60.11	59.61	57.06	55.02	53.27	52.79	52.37	53.31	52.63	56.13
RanPAC-R	66.67	42.74	36.69	33.63	31.01	27.22	29.12	28.64	29.82	28.16	35.37
RanDumb-R	64.68	39.34	29.03	28.37	25.63	21.99	23.13	22.02	22.89	21.91	29.90
RanPAC	66.72	34.36	15.67	16.01	13.73	9.59	8.83	7.09	8.88	7.24	18.81
RanDumb	65.12	34.53	15.62	16.80	13.95	9.68	8.96	7.07	8.91	7.32	18.80
L2P	67.01	61.92	58.23	54.58	54.35	53.13	52.93	52.44	52.35	52.08	55.90
DualPrompt	68.02	66.26	63.33	62.17	62.23	62.57	60.92	60.54	60.08	61.17	62.73
ConvPrompt	85.90	81.19	74.99	73.56	73.27	72.54	72.14	71.05	71.04	70.92	74.66
MOS	86.50	80.76	75.22	73.65	68.09	64.32	62.90	58.81	57.33	52.13	67.97
PROL	86.48	83.27	80.35	78.41	77.35	76.60	76.26	74.85	74.25	73.50	78.13

Table A4. Detailed Forgetting of consolidated methods in ImageNet-R

Method	1	2	3	4	5	6	7	8	9	10	AVG
RanPAC-J	-	7.03	4.39	4.82	5.90	6.34	6.27	6.08	6.19	6.63	5.96
RanDumb-J	-	5.91	3.89	3.62	5.63	6.00	5.87	5.84	5.95	6.04	5.42
RanPAC-R	-	47.34	36.84	39.78	40.11	40.58	37.82	37.83	38.98	39.81	39.90
RanDumb-R	-	53.68	48.11	48.68	48.87	49.28	47.14	47.05	48.40	48.81	48.89
RanPAC	-	66.72	67.31	66.86	66.16	65.90	65.47	65.55	66.05	67.27	66.36
RanDumb	-	64.87	66.55	66.46	66.47	66.33	66.02	66.05	66.51	67.67	66.33
L2P	-	7.99	5.30	6.18	5.99	5.26	5.14	5.04	6.04	5.67	5.85
DualPrompt	-	1.74	1.82	2.11	2.14	1.74	2.56	2.70	4.76	3.75	2.59
ConvPrompt	-	2.33	1.59	1.96	1.93	1.92	2.22	3.38	3.73	3.39	2.49
MOS	-	9.36	14.39	15.96	20.99	24.98	25.47	30.44	30.67	35.72	23.11
PROL	-	3.20	2.03	2.93	3.11	3.45	3.65	3.90	4.89	4.82	3.55

References

[1] Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and Simone Calderara. Dark experience for general continual learning: a strong, simple baseline. *Advances in neural information processing systems*, 33:15920–15930, 2020. 1

[2] Lucas Caccia, Rahaf Aljundi, Nader Asadi, Tinne Tuyte-

Table A5. Detailed Accuracy of consolidated methods in ImageNet-A

Method	1	2	3	4	5	6	7	8	9	10	AVG
RanPAC-J	67.00	62.80	56.63	54.42	53.27	51.29	48.93	47.74	45.86	45.03	53.30
RanDumb-J	67.25	61.77	55.72	53.04	52.04	49.74	48.03	46.43	44.91	44.07	52.30
RanPAC-R	68.32	40.54	35.54	28.85	32.36	31.09	30.07	29.35	26.96	28.54	35.16
RanDumb-R	67.37	35.76	29.98	22.64	27.95	27.71	25.83	25.24	20.81	24.86	30.82
RanPAC	67.37	31.56	17.67	10.36	5.09	2.24	3.77	3.24	3.47	3.77	14.86
RanDumb	67.25	31.00	16.92	10.12	4.80	2.03	3.45	3.11	3.18	3.69	14.56
L2P	36.85	32.67	26.11	21.14	20.47	17.88	17.19	16.62	15.22	14.64	21.88
DualPrompt	0.87	25.83	28.50	29.59	26.65	22.94	21.45	20.30	19.49	20.05	21.57
ConvPrompt	-	-	-	-	-	-	-	-	-	-	-
MOS	72.41	66.67	62.30	57.17	52.45	51.09	49.60	48.39	46.53	44.79	55.14
PROL	78.04	69.09	63.70	61.82	58.80	55.99	53.80	51.57	49.03	47.72	58.96

Table A6. Detailed Forgetting of consolidated methods in ImageNet-A

Method	1	2	3	4	5	6	7	8	9	10	AVG
RanPAC-J	-	5.02	5.45	5.21	4.64	4.61	6.14	6.02	6.97	6.92	6.92
RanDumb-J	-	5.25	4.46	4.57	3.58	3.37	3.78	4.10	4.94	4.98	4.34
RanPAC-R	-	50.70	42.50	42.19	34.60	33.36	34.42	33.63	35.74	32.05	37.69
RanDumb-R	-	58.44	49.84	48.92	39.01	36.06	37.21	36.10	40.42	34.79	42.31
RanPAC	-	67.37	69.85	65.26	66.32	63.98	60.11	59.23	58.44	57.76	63.15
RanDumb	-	66.34	68.77	63.56	64.72	62.20	57.82	56.62	56.07	55.15	61.25
L2P	-	12.53	7.67	8.86	7.07	5.69	5.74	5.04	5.35	6.48	62.25
DualPrompt	-	0.50	0.12	1.79	1.69	1.40	2.10	1.96	1.75	1.91	1.47
ConvPrompt	-	-	-	-	-	-	-	-	-	-	-
MOS	-	5.05	6.29	11.71	14.50	15.02	14.32	13.84	14.32	15.50	12.28
PROL	-	5.34	5.19	4.11	3.05	2.55	2.41	2.57	2.78	3.29	3.48

Table A7. Detailed Accuracy of consolidated methods in CUB

Method	1	2	3	4	5	6	7	8	9	10	AVG
RanPAC-J	93.11	87.48	87.10	85.17	86.54	85.32	84.81	83.84	84.51	85.20	86.31
RanDumb-J	93.40	88.38	87.43	85.70	87.12	86.23	85.62	84.60	85.19	85.82	86.84
RanPAC-R	92.75	83.03	78.25	78.59	79.39	77.49	77.05	75.64	75.87	75.63	79.37
RanDumb-R	93.46	82.66	79.08	79.17	78.58	75.99	75.74	74.27	74.20	73.81	78.70
RanPAC	93.11	44.63	33.10	22.23	19.80	14.78	13.07	10.05	10.32	9.52	27.06
RanDumb	93.40	45.54	32.80	22.37	19.83	14.89	13.21	10.15	10.40	9.53	27.21
L2P	92.43	80.68	74.77	72.85	72.12	69.67	67.45	62.26	60.43	61.98	71.46
DualPrompt	0.00	38.00	50.90	54.64	59.81	60.73	59.95	59.78	57.58	58.06	49.95
ConvPrompt	93.79	85.60	78.23	76.19	77.97	76.36	74.33	71.80	70.38	70.12	77.48
MOS	96.63	91.27	85.47	82.57	76.95	72.55	71.25	66.70	65.19	61.74	77.03
PROL	94.17	87.43	82.03	78.74	80.10	78.83	77.33	74.75	73.05	72.51	79.89

Table A8. Detailed Forgetting of consolidated methods in CUB

Method	1	2	3	4	5	6	7	8	9	10	AVG
RanPAC-J	-	0.39	2.23	2.97	2.67	3.63	3.30	3.04	2.74	2.63	2.62
RanDumb-J	-	1.10	2.77	3.26	2.85	3.44	3.31	2.94	2.69	2.52	2.82
RanPAC-R	-	11.26	16.03	12.82	12.65	14.14	14.01	14.10	14.03	14.65	13.74
RanDumb-R	-	14.50	17.57	13.93	15.34	17.55	16.91	16.66	16.76	17.45	16.30
RanPAC	-	93.11	88.86	90.45	89.38	90.66	90.48	90.13	88.75	88.98	90.09
RanDumb	-	93.27	90.14	91.02	90.08	91.29	91.08	90.75	89.44	89.69	90.75
L2P	-	2.72	12.68	10.66	11.81	12.45	11.97	11.60	10.22	9.81	10.44
DualPrompt	-	0.00	6.78	5.66	6.59	6.41	5.61	4.93	4.63	6.23	5.20
ConvPrompt	-	1.94	13.98	9.92	8.97	8.13	6.98	6.18	5.42	7.21	7.64
MOS	-	9.11	16.05	18.59	23.86	28.27	27.81	32.31	33.91	37.38	25.25
PROL	-	1.55	11.98	8.77	8.43	8.98	8.97	8.13	7.22	8.92	8.11

laars, Joelle Pineau, and Eugene Belilovsky. New insights on reducing abrupt representation change in online continual learning. *arXiv preprint arXiv:2104.05025*, 2021. 1

[3] Yiduo Guo, Bing Liu, and Dongyan Zhao. Online continual learning through mutual information maximization. In *International conference on machine learning*, pages 8109–8126. PMLR, 2022. 1

Table A9. Detailed Throughput of consolidated methods in ImageNet-R

Method	1	2	3	4	5	6	7	8	9	10	AVG
RanPAC-J	56.51	82.89	72.54	80.14	84.26	88.76	99.13	90.00	89.29	85.67	82.92
RanDumb-J	55.28	76.34	87.05	92.96	73.46	82.19	71.25	81.82	73.53	76.41	77.03
RanPAC-R	66.92	82.89	91.63	89.38	73.46	85.35	65.14	72.00	119.05	148.79	89.46
RanDumb-R	79.47	107.44	145.08	110.67	114.60	130.53	126.67	112.50	119.05	122.91	116.89
RanPAC	77.06	90.66	87.05	83.00	84.26	96.48	81.43	78.26	92.59	97.48	86.83
RanDumb	51.90	69.07	108.81	80.14	81.86	110.95	81.43	105.88	83.33	85.67	85.90
L2P	56.51	54.74	36.27	43.85	36.27	36.38	36.19	36.73	36.23	36.24	40.94
DualPrompt	42.38	52.75	54.41	50.52	51.16	51.60	54.29	35.29	36.76	35.78	46.50
ConvPrompt	18.04	16.67	15.01	14.71	12.91	11.68	8.64	7.41	6.78	5.89	11.77
MOS	14.37	13.25	9.67	11.56	13.77	10.88	13.49	15.65	29.07	28.85	16.06
PROL	32.60	31.53	37.04	35.21	37.70	34.14	38.00	40.91	30.86	36.24	35.42

Table A10. Detailed Training Time of consolidated methods in ImageNet-R

Method	1	2	3	4	5	6	7	8	9	10	AVG
RanPAC-J	45.00	35.00	24.00	29.00	34.00	25.00	23.00	20.00	28.00	33.00	29.60
RanDumb-J	46.00	38.00	20.00	25.00	39.00	27.00	32.00	22.00	34.00	37.00	32.00
RanPAC-R	38.00	35.00	19.00	26.00	39.00	26.00	35.00	25.00	21.00	19.00	28.30
RanDumb-R	32.00	27.00	12.00	21.00	25.00	17.00	18.00	16.00	21.00	23.00	21.20
RanPAC	33.00	32.00	20.00	28.00	34.00	23.00	28.00	23.00	27.00	29.00	27.70
RanDumb	49.00	42.00	16.00	29.00	35.00	20.00	28.00	17.00	30.00	33.00	29.90
L2P	45.00	53.00	48.00	53.00	79.00	61.00	63.00	49.00	69.00	78.00	59.80
DualPrompt	60.00	55.00	32.00	46.00	56.00	43.00	42.00	51.00	68.00	79.00	53.20
ConvPrompt	141.00	174.00	116.00	158.00	222.00	190.00	264.00	243.00	369.00	480.00	235.70
MOS	177.00	219.00	180.00	201.00	208.00	204.00	169.00	115.00	86.00	98.00	165.70
PROL	78.00	92.00	47.00	66.00	76.00	65.00	60.00	44.00	81.00	78.00	68.70

Table A11. Detailed Inferfence Time of consolidated methods in ImageNet-R

Method	1	2	3	4	5	6	7	8	9	10	AVG
RanPAC-J	10.29	21.00	27.37	44.49	53.27	53.38	75.35	85.34	74.61	107.30	55.24
RanDumb-J	9.70	15.80	28.27	44.34	45.73	59.51	69.59	75.77	90.66	98.53	53.79
RanPAC-R	10.34	40.23	36.92	65.13	78.80	87.39	126.79	113.13	131.69	151.57	84.20
RanDumb-R	7.74	14.80	25.75	31.27	36.84	44.17	59.77	64.45	68.29	80.57	43.36
RanPAC	10.89	22.25	27.39	28.72	44.23	55.75	53.36	64.28	60.94	84.30	45.21
RanDumb	9.97	18.95	25.77	42.52	47.30	50.34	58.10	65.96	74.19	75.05	46.81
L2P	7.00	16.00	22.00	42.00	53.00	62.00	72.00	80.00	90.00	101.00	54.50
DualPrompt	9.00	18.00	27.00	32.00	43.00	48.00	74.00	84.00	98.00	93.00	52.60
ConvPrompt	9.27	22.18	31.75	47.56	65.13	93.09	122.91	174.15	183.34	289.80	103.92
MOS	29.00	78.00	247.00	281.00	232.00	503.00	453.00	383.00	464.00	552.00	322.20
PROL	11.46	23.07	37.61	37.77	51.12	65.09	64.43	72.62	92.26	98.52	55.39

[4] Yiduo Guo, Bing Liu, and Dongyan Zhao. Dealing with cross-task class discrimination in online continual learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11878–11887, 2023. 1

[5] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, Dawn Song, Jacob Steinhardt, and Justin Gilmer. The many faces of robustness: A critical analysis of out-of-distribution generalization. *ICCV*, 2021. 1

[6] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. *CVPR*, 2021. 1

[7] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 1

[8] Mark D McDonnell, Dong Gong, Amin Parvaneh, Ehsan Abbasnejad, and Anton Van den Hengel. Ranpac: Random projections and pre-trained models for continual learning. *Advances in Neural Information Processing Systems*, 36:12022–12053, 2023. 1

[9] Ameya Prabhu, Shiven Sinha, Ponnurangam Kumaraguru, Philip Torr, Ozan Sener, and Puneet Dokania. Randumb: Random representations outperform online continually learned representations. *Advances in Neural Information Processing Systems*, 37:37988–38006, 2024. 1

[10] Matthew Riemer, Ignacio Cases, Robert Ajemian, Miao Liu, Irina Rish, Yuhai Tu, and Gerald Tesauro. Learning to learn without forgetting by maximizing transfer and minimizing interference. *arXiv preprint arXiv:1810.11910*, 2018. 1

[11] Anurag Roy, Riddhiman Moulick, Vinay K Verma, Saptarshi Ghosh, and Abir Das. Convolutional prompting meets language models for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23616–23626, 2024. 1

[12] Hai-Long Sun, Da-Wei Zhou, Hanbin Zhao, Le Gan, De-Chuan Zhan, and Han-Jia Ye. Mos: Model surgery for pre-trained model-based class-incremental learning. *arXiv preprint arXiv:2412.09441*, 2024. 1

[13] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011. 1

[14] Maorong Wang, Nicolas Michel, Jiafeng Mao, and Toshihiko Yamasaki. Dealing with synthetic data contamination in online continual learning. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. 1

[15] Zifeng Wang, Zizhao Zhang, Sayna Ebrahimi, Ruoxi Sun, Han Zhang, Chen-Yu Lee, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, et al. Dualprompt: Complementary prompting for rehearsal-free continual learning. In *European Conference on Computer Vision*, pages 631–648. Springer, 2022. 1

[16] Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. Learning to prompt for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 139–149, 2022. 1

[17] Yujie Wei, Jiaxin Ye, Zhizhong Huang, Junping Zhang, and Hongming Shan. Online prototype learning for online continual learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 18764–18774, 2023. 1

[18] Da-Wei Zhou, Hai-Long Sun, Han-Jia Ye, and De-Chuan Zhan. Expandable subspace ensemble for pre-trained model-based class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23554–23564, 2024. 1