A Continual Deepfake Detection Benchmark: Dataset, Methods, and Essentials (Supplementary Material)

Chuqiao Li[1], Zhiwu Huang[2,*], Danda Pani Paudel[1], Yabin Wang[3,2],
Mohamad Shahbazi[1], Xiaopeng Hong[4], Luc Van Gool[1,5]
[1]ETH Zürich, Switzerland [2]Singapore Management University, Singapore
[3]Xi'an Jiaotong University, China [4]Harbin Institute of Technology, China [5]KU Leuven, Belgium
chuqli@student.ethz.ch, zzhiwu.huang@gmail.com,
{paudel, mshahbazi, vangool}@vision.ee.ethz.ch,
iamwangyabin@stu.xjtu.edu.cn, hongxiaopeng@ieee.org

Abstract

In the supplementary material, we further present (1) illustration of the suggested approaches for the adaptation of class incremental learning (CIL) methods to the continual deepfake detection problem (CDD), (2) training details of the evaluated methods on the proposed continual deepfake detection benchmark (CDDB), (3) study on more essential components of CIL methods on the suggested CDDB, (4) evaluation on the GANFake [8] dataset, (5) experiments on the generalization capability of the suggested CIL methods to unseen domains and corrupted test images, and (6) more overviews of the suggested CDDB evaluations.

1. Illustrations of Adapting CIL to CDD

In the main paper, we study three main adaptations of CIL methods for CDD. Fig.1 illustrates the suggested three approaches that adapts the exiting CIL methods to the context of CDD.

2. Training Settings

We evaluated three types of the state-of-the-art class incremental learning (CIL) methods on the suggested CDDB.
Gradient-based Method. We used the official code of the null space class incremental learning method (NSCIL) [13], which is one of the state-of-the-art gradient-based methods. We followed the official code's setup to tune the hyperparameter λ from the default setup $\{10, 30\}$ to the new settings $\{100, 1\}$, which is used to select the smallest singular values corresponding to the null space. The larger λ leads to larger approximate null space, increasing the plasticity to learn new tasks while decreasing the memorization of old

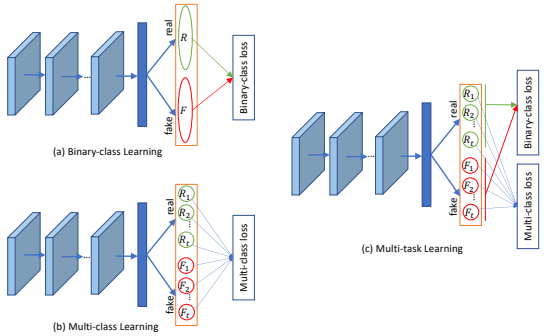

Figure 1. Three main adaptions (a), (b), (c) of multi-class incremental learning for CDD. Here R, F indicate the logits/features for reals and fakes respectively, and t is the number of deepfake tasks.

tasks [13]. We trained the NSCIL model for 12 epochs in total, as we found that training more epochs for NSCIL resulted in performance degradation. The initial learning rate is 0.001 for the first task and 0.0001 for all other tasks and is divided by 2 after 4 and 8 epochs for the EASY evaluation. For batch normalization layers, the learning rate starts from 5×10^{-5}. The other parameters are the same as official NSCIL implementation.

Memory-based Method. We employed latent replay class incremental learning (LRCIL) [10], which is one of the most effective and efficient memory-based methods. LRCIL was trained in the NSCIL's framework, with 20 epochs in total, the initial learning rate is 0.001 for the first task and 0.0001 for all other tasks and is divided by 2 after 10 and 15 epochs for the EASY, HARD, LONG evaluations. For batch normalization layers, the learning rate starts from 5×10^{-5}. For the GANfake evaluation, we used 60 epochs instead and the initial learning rate is divided by 2 after 20 and 40 epoch. Following original LRCIL's implementation, there is no knowledge distillation loss. We also tried to add distillation loss with a factor $\gamma_d = 0.3$. And we chose the second layer as latent layer. We used Adam with the batch

*Corresponding Author

size 32 to train the network.

Distillation-based Method. We utilized five state-of-the-art distillation-based methods, i.e., incremental classifier and representation learning (iCaRL) [11], Learning a Unified Classifier Incrementally via Rebalancing (LUCIR) [6], Dynamic Token Expension (DyTox) [4], as well as Compositional Class Incremental Learning (CCIL) [9], which is additionally evaluated in the Supp. Material. Besides, we also evaluated the multi-task variant of iCaRL (iCaRL-SumLog) [8].

For iCaRL, we trained its model for 30 epochs in total. The learning rate starts from 0.0001 and is divided by $\frac{10}{3}$ after 10 and 20 epochs for the EASY, HARD and LONG evaluations. For GANfake sequences, we used 60 epochs instead and the initial learning rate 0.001 is divided by $\frac{10}{3}$ after 20 and 40 epoch. Following the original iCaRL's implementation, γ_d was set to 1 for knowledge distillation loss $\ell_{distill}$ with temperature as $T = 1$. We used Adam with the batch size 32 to train the network.

For CCIL, we used its official code[1]. The initial learning rate, total epoches and learning rate decay are the same as iCaRL. We tried the method with ($\gamma_d = 0.3$) or without ($\gamma_d = 0$) knowledge distillation $\ell_{distill}$. Also, we tried it with or without applying mixup and label smoothing techniques to the whole training process which is different from the original implementation (only for the first training phase), and we followed the relevant parameters (e.g, mixup weight). We used Adam with the batch size 32 to train the network. The other parameters are the same as the official implementation of CCIL.

For LUCIR, the learning rate starts from 0.0001 and is divided by 10 after 16 and 33 epochs (50 epochs in total) for the LONG evaluation. For the EASY, HARD and GANfake evaluations, we used 40 epochs and the learning rate is divided by 10 after 13 and 26 epoch. γ_d is set to 0.5 for knowledge distillation loss $\ell_{distill}$. γ_m was set to 0.1 for supplementary loss ℓ_{supp}. Within ℓ_{supp}, the number of closest class embeddings J was set to 2 and the threshold τ was set to 0.2 for all experiments. The other parameters are the same as official LUCIR implementation. We used Adam with the batch size 32 to train the LUCIR network.

For DyTox [4], we used its official implementation[2] with the default settings on its hyperparameters. DyTox [4] originally suggests training its backbone from scratch. However, this case's performance (about 56%) is much lower than that of training from the ImageNet pre-trained base-ConViT[3] on CDDB-Hard500. Therefore, we apply the pre-trained base-ConViT for DyTox throughout the paper.

For a fair comparison, except for DyTox that is based on an ImageNet pre-trained transformer ConViT [5], we used the state-of-the-art deepfake CNN detector (CNNDet) [14], which applies a ResNet-50 pretrained on ImageNet [3] and ProGAN [7] as the backbone for all the remaining methods over the proposed CDDB. For the multi-task (MT) learning variants of the evaluated CIL methods, the study on the trade-off hyperparameter λ is presented in Table 1. In Table 1, we aim to compare all the colorized triplets (LRCIL, iCaRL, LUCIR)'s performances to see which triplets consistently perform well. Each triplet should be of the same MT variant and the same λ value (e.g., LRCIL-SumLogit-0.3, iCaRL-SumLogit-0.3, LUCIR-SumLogit-0.3). By comparing these triplets in terms of their average performances on (LRCIL, iCaRL, LUCIR), we can see that the SumLogit case ($\lambda = 0.3$) works the most promisingly on the CDDB-EASY1500 evaluation. Furthermore, it is not feasible to tune the hyperparameters on the test data in the real-world scenario. Accordingly, we used the same hyperparameter $\lambda = 0.3$ for all the MT variants of LRCIL, iCaRL and LUCIR throughout the main paper.

3. Study on More Essentials

Table 2 studies more essentials in CIL on CDDB. They are *cosine normalization* on linear fully connected (FC) layer, *label smoothing* and *mixup* techniques.

Linear FC (LinFC). The main paper finds that performing cosine normalization on FC (CosFC) layers result in LUCIR [6] and CCIL [9] is clearly inferior performance than using LinFC. This is because CosFC is originally designed to address the *class imbalance issue*, which however is much less serious in the CDD context. We additionally apply LinFC to CCIL* [9] that further drops knowledge distillation. We can see that the LinFC version of CCIL* also performs better than its corresponding CosFC version, enhancing the conclusion that CosFC is too hard for class imbalance based normalization so that it really hurts the CDD performance.

Label Smoothing (LabelSM). We also study *LabelSM* that is commonly used to improve generalization and reduce overconfidence of classification models [12, 9]. This technique affects the cross-entropy loss for classification by interpolating the one-hot labels with a uniform distribution over the possible classes [9]. From the results in Table 2, we can discover that LabelSM slightly improves the AA scores in the cases of iCaRL [11] and LUCIR [6], while hurting the AA scores in the case of LRCIL [10] and CCIL [9].

Mixup. We further employ *Mixup* that is used a form of data augmentation for better clarification in general. The mixup technique is to generate training samples by linearly combining pairs of training samples (i.e., images and labels). Following [9], we applied the mixup data to the training data and merely used them (no original training data)

[1] https://github.com/sud0301/essentials_for_CIL

[2] https://github.com/arthurdouillard/dytox

[3] We empirically find that the performance (AA=96.14%) of ImageNet pre-trained ConViT is very comparable to that (AA=96.29%) of the one that is further fine-tuned on ProGAN for the EASY1500 evaluation. Therefore, for the transformer-based networks, we suggest only using ImageNet pre-trained models without the further warm-up step on ProGAN.

Learning System	Evaluated Method	CDDB-EASY1500 0.1	0.3	0.5	0.7	0.9
Multi-task (MT) learning	LRCIL[10]-SumLog[8]	**89.54**	88.76	86.51	87.08	87.20
	iCaRL[11]-SumLog[8]	88.97	87.05	87.51	87.39	86.29
	LUCIR[6]-SumLog[8]	91.65	91.56	**92.05**	91.39	91.08
	LRCIL[10]-**SumLogit**	89.38	89.33	88.08	88.52	87.71
	iCaRL[11]-**SumLogit**	88.52	90.43	90.37	87.88	87.57
	LUCIR[6]-**SumLogit**	91.48	92.00	91.90	91.11	91.67
	LRCIL[10]-**SumFeat**	88.23	87.81	88.33	87.68	86.27
	iCaRL[11]-**SumFeat**	87.37	87.74	86.26	86.52	85.46
	LUCIR[6]-**SumFeat**	91.75	91.81	91.03	91.28	91.48
	LRCIL[10]-**Max**	89.17	89.16	87.83	88.27	85.66
	iCaRL[11]-**Max**	90.11	89.92	89.76	**90.78**	87.11
	LUCIR[6]-**Max**	91.30	91.21	91.20	91.35	91.04

Table 1. Empirical study on different multi-task learning trade-off parameter λ values for the suggested CDDB' EASY1500 evaluation. The reported results are Average Accuracies (AA) for continual deepfake detection. **Bold**: best **green/blue/red** results, Underline: second/third best green/blue/red results. The best/second-best/third-best LRCIL, iCaRL, LUCIR results are in green, blue, and red respectively. *Note that we aim at comparing all the colorized triplets (LRCIL, iCaRL, LUCIR)'s performances to see which triplets consistently perform well. Each triplet should be of the same MT variant and the same λ value.*

Learning System	Evaluated Method	CDDB-EASY1500 Task1	Task2	Task3	Task4	Task5	Task6	Task7	AA	AF	AA-M
Multi-class (MC) learning	LRCIL[10]	83.50	77.88	90.84	98.90	84.75	98.86	65.92	85.81	-5.88	67.11
	iCaRL[11]	77.50	71.38	91.22	99.57	95.66	99.92	78.28	87.65	-9.41	65.39
	LUCIR(CosFC)[6]	81.05	88.25	94.47	99.73	90.30	99.73	57.15	87.24	-6.32	63.27
	CCIL(CosFC)[9]	55.65	56.88	67.18	89.34	75.88	89.38	64.32	71.23	-22.01	39.53
	CCIL*(CosFC)[9]	58.35	60.00	71.37	86.25	90.48	86.21	71.01	74.81	-22.56	48.66
	LUCIR[6]-**LinFC**	91.60	89.12	92.56	99.76	94.45	99.80	71.21	91.21	-2.88	74.62
	CCIL[9]-**LinFC**	60.50	66.12	81.49	96.16	73.38	98.55	60.45	76.66	-16.50	41.85
	CCIL*[9]-**LinFC**	62.70	70.75	82.63	98.51	90.76	99.80	75.42	82.94	-13.24	61.95
	LRCIL[10]-**LabelSM**	72.70	74.13	89.89	99.45	92.14	99.57	72.52	85.77	-5.93	68.17
	iCaRL[11]-**LabelSM**	79.55	80.62	90.46	99.84	89.37	99.84	80.17	88.55	-8.93	71.82
	LUCIR[6]-**LabelSM**	84.15	85.75	94.47	100.00	93.25	100.00	82.36	91.42	-5.31	77.35
	CCIL*[9]-**LabelSM**	58.60	65.25	61.83	98.43	88.54	98.43	72.52	77.66	-18.16	53.85
	LRCIL[10]-**Mixup**	52.30	52.13	62.40	85.54	78.65	96.94	70.00	71.14	-25.91	35.43
	iCaRL[11]-**Mixup**	78.05	84.88	94.47	100.00	94.09	100.00	80.47	90.28	-6.95	75.42
	LUCIR[6]-**Mixup**	82.31	87.22	90.70	96.32	92.98	93.21	70.59	87.62	-6.54	70.93
	CCIL*[9]-**Mixup**	57.80	66.25	80.34	98.63	72.27	98.9	73.92	78.30	-18.98	55.16

Table 2. Benchmarking results on essentials of CIS methods on CDDB's *EASY* evaluation. CCIL* indicates the CCIL method without using knowledge distillation loss. AA: Average Accuracy for deepfake detection, AF: Average Forgetting degree, AA-M: Average Accuracy for deepfake recognition. The AA results of LRCIL, iCaRL, LUCIR, CCIL are in green, blue, red, and brown respectively.

Learning System	Evaluated Method	GANfake [8] 1024 AA	1024 AF	512 AA	512 AF
Multi-task learning	iCaRL [11]-SumLog [8]	91.72	-4.28	90.32	-5.71
	iCaRL [8]-**SumLogit**	93.19	-1.02	86.77	-10.09
	LUCIR [6]-**SumLogit**	92.67	-2.34	91.88	-2.65

Table 3. Continual GANfake detection accuracies of the main evaluated methods on the GANfake dataset [8] after the training on the last GAN using different memory budgets (1024 and 512).

to train the evaluated methods. It is interesting to see that mixup brings a clear performance degradation in the case of LRCIL. This is because the mixup operations are performed in the space of raw images, which is not consistent with learning mechanism of LRCIL that performs rehearsal in the latent space of feature maps. Except for LRCIL and CCIL*, the other two methods iCaRL, LUCIR favor the mixup technique for further improvement in terms of AA on the suggested CDDB.

4. Evaluation on the GANfake Dataset

For the GANfake [8] dataset[4], Table 3 summarizes the evaluation results of the four main evaluated methods, i.e., the variant of iCaRL method [8], our suggested SumLogit variants of LRCIL [10], iCaRL [11] and LUCIR [6]. The results show that the suggested MT variants mostly work better than the original one (i.e., SumLog). From the overall results, we can find that the highest AA score[5] (i.e., 93.19) for the case of memory budget=1024 on GANfake is clearly higher than the one (i.e., 82.53) on our suggested CDDB-HARD1000 (memory budget=1000). Besides, the highest AA score (i.e., 91.88) for the case of memory size=512 on GANfake is much higher than the one (i.e., 80.77) on the suggested CDDB-HARD evaluation with memory budget = 500. Moreover, the highest AA score (i.e., 93.19) on GANfake (memory size=1024) is even higher than the one (i.e., 92.00) of our easiest evaluation, i.e., CDDB-EASY1500 (memory size=1500). This implies that our suggested CDDB is clearly more challenging than GANfake, and therefore we believe it will has a high potential to promote more solid research in the domain of continual deepfake detection.

5. Experiments on the Generalization Capability of Suggested Methods

The knowledge accumulation nature of the suggested CDDB allows for a better generalization to deepfakes from unseen domains. To verify this, we compare the adapted continual learning methods (e.g., LRCIL [10], iCaRL [11], LUCIR [6]) against CNNDet [14] (pre-trained on ProGAN) and CNNDet [14]-Finetune (starts from ProGAN to the

[4]Since the GANfake [8] dataset does not release the detailed train/val/test splits, we can merely evaluate the methods on our own splits following the description in the oiriginal paper. Accordingly, the reported results are not really comparable with those reported in [8].

[5]The reported highest AA is 97.22 in GANfake [8] when the memory budget is 1024. It is even higher than our reported one, while their train/val/test splits in [8] are different from ours.

	IMLE	CRN	Glow	Average Acc (↑)	mean Average Pre (↑)
CNNDet [66]	53.64	53.64	45.56	50.95	81.61
CNNDet [66]-Finetune	50.43	50.27	50.00	50.23	52.19
LRCIL[51]-Adapted	76.72	93.06	68.02	**79.28**	**87.78**
iCaRL[52]-Adapted	83.31	84.17	66.73	78.07	82.31
LUCIR[23]-Adapted	70.02	93.30	70.90	78.07	86.39

Table 4. Accuracies of generalization of the trained models on CDDB-Hard to 3 unseen domains (IMLE, CRN, Glow), which are of non-GANs and thus highly heterogeneous to the used CDDB-Hard data. **Bold**: best, Underline: second best. *Note that we have different test data on IMLE and CRN from CNNDet [66] that has no Glow data, the reported mean average precisions (mAPs) are somewhat different.*

	GauGAN	BigGAN	WildFake	WhichFace	SAN	Average Acc (↑)
CNNDet	53.80	58.00	61.50	48.96	63.33	57.12
CNNDet[66]-Finetune	50.95	50.25	52.73	48.25	71.11	54.66
LRCIL[51]-Adapted	61.20	61.75	56.86	59.75	48.89	57.69
iCaRL[52]-Adapted	67.35	64.38	66.36	69.75	56.67	**64.90**
LUCIR[23]-Adapted	62.45	61.00	56.42	56.75	57.78	58.88

Table 5. Accuracies of generalization to corrupted (Blur+JPEG) test images that are from the five tasks of CDDB-Hard. **Bold**: best, Underline: second best

CDDB-Hard data stream). Due to the CDD setup, we cannot compare CNNDet trained directly on the whole data stream. Table 4 reports the results, showing the clearly better generalization of our adapted ones. On the other hand, we also test the generalization ability of these methods to corrupted test images using the same strategy, i.e., Blur+JPEG (0.5), from [14]. Note that [14] suggests Blur+JPEG (0.5) for data augmentation during training, and testing on clean images. Despite no Blur+JPEG augmentation for training, our adapted ones (especially iCaRL) generally work better than the competitors (Table 5).

6. More Overviews of Five CDDB Evaluations

Fig.2 presents radar plots on the four evaluation metrics, i.e., AAs, AFs, mAPs and AA-Ms, of the evaluated methods for *EASY1500*, *LONG1500*, *HARD1500*, *HARD1000*, and *HARD500*, where 1500, 1000, 500 are three studied memory budgets, AA, AF, AA-M, mAP are Average Accuracy for deepfake detection, Average Forgetting degree, Average Accuracy for deepfake recognition, and mean Average Precision respectively. The mAP is calculated by the mean of areas under the precision-recall (PR) curves. For those multi-class and multi-task learning methods, we feed the PR calculation function with the normalized value of the maximum $\varphi(\cdot)$ activation output on fakes. Formally, the normalized value is $p_F = \frac{M_F}{M_F+M_R}$, where M_R and M_F are the maximum $\varphi(\cdot)$ activation values for reals and fakes respectively.

From the overall results in Fig.2, we can see that LUCIR-MT (multi-task learning) generally works the best than the other evaluated methods in terms of AA, AF, AA-M and mAP for the five evaluations. By comparing the five different evaluations, HARD500's AA, AF, AA-M, mAP scores are clearly worse than the other four evaluations. This implies that the detection over HARD500 is the most challenging. In particular, it suffers from much more serious catastrophic forgetting due to the smaller memory budget. Fig.3 and Fig.4 present the PR curves of the LUCIR method and the DyTox method for the five evaluations *EASY1500*, *LONG1500*, *HARD1500*, *HARD1000*, and *HARD500*. The PR curves demonstrate that the most difficult tasks are generally the ones on SAN [2], WildDeepfake [15], and WhichFaceReal [1] for the different evaluations. This is because SAN is a small data set, and WildDeepfake/WhichFaceReal consists of unknown deepfakes in the wild scenes.

Figure 2. Radar plots on AAs, AFs, mAPs and AA-Ms of the evaluated methods for *EASY1500*, *LONG1500*, *HARD1500*, *HARD1000*, and *HARD500*, where 1500, 1000, 500 are memory budgets, BC/MC/MT: binary-class/multi-class/multi-task learning methods that have the highest AAs/mAPs, AA: Average Accuracy (detection), AF: Average Forgetting degree, AA-M: Average Accuracy (recognition), mAP: mean Average Precision, i.e., the mean of areas under the PR curve.

References

[1] Which face is real? *http://www.whichfaceisreal.com.*

[2] Tao Dai, Jianrui Cai, Yongbing Zhang, Shu-Tao Xia, and Lei Zhang. Second-order attention network for single image super-resolution. In *CVPR*, 2019.

[3] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

[4] Arthur Douillard, Alexandre Ramé, Guillaume Couairon, and Matthieu Cord. Dytox: Transformers for continual learning with dynamic token expansion. In *CVPR*, 2022.

[5] Stéphane d'Ascoli, Hugo Touvron, Matthew L Leavitt, Ari S Morcos, Giulio Biroli, and Levent Sagun. Convit: Improving

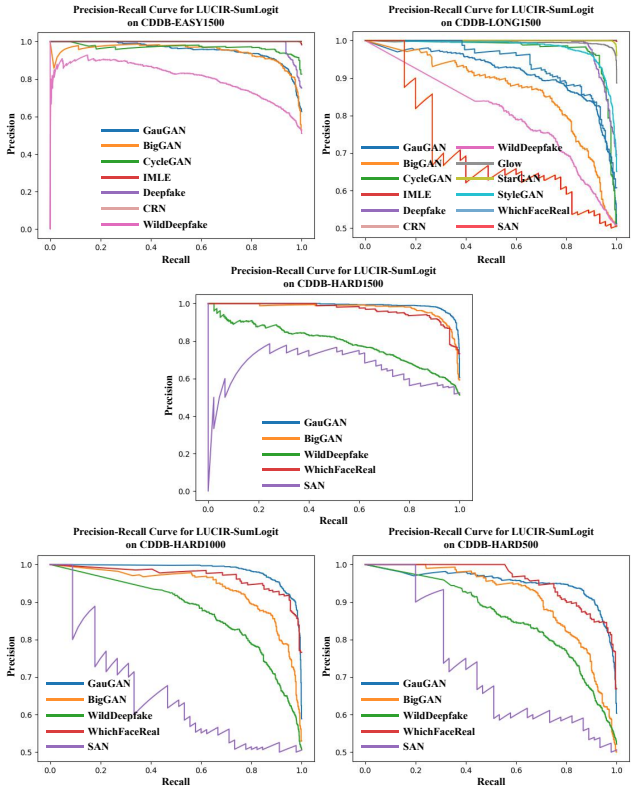

Figure 3. Precision-Recall (PR) curves of LUCIR-SumLogit with the highest mAP for the five CDDB evaluations *EASY1500*, *LONG1500*, *HARD1500*, *HARD1000*, and *HARD500*, where 1500, 1000, 500 are memory budgets, and mAP is mean Average Precision, i.e., the mean of areas under all the PR curves.

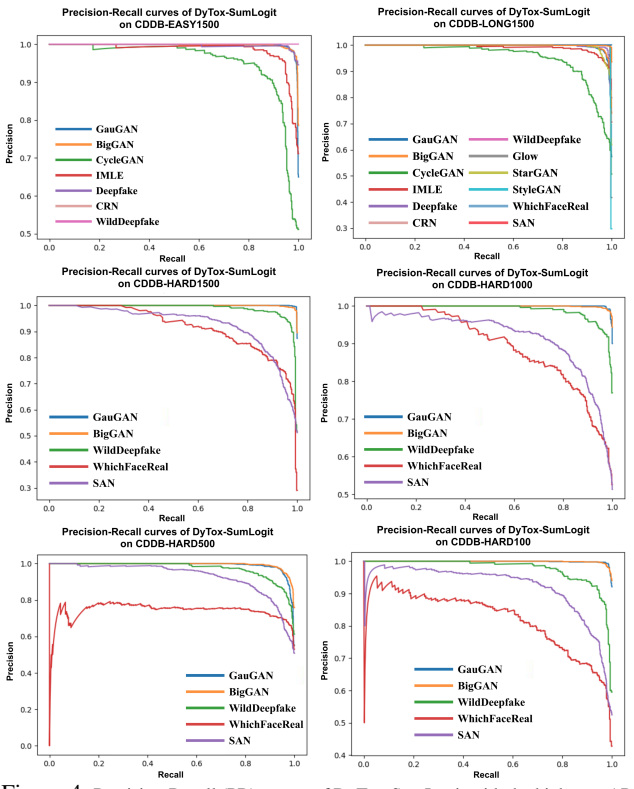

Figure 4. Precision-Recall (PR) curves of DyTox-SumLogit with the highest mAP for the five CDDB evaluations *EASY1500*, *LONG1500*, *HARD1500*, *HARD1000*, and *HARD500*, where 1500, 1000, 500 are memory budgets, and mAP is mean Average Precision, i.e., the mean of areas under all the PR curves.

vision transformers with soft convolutional inductive biases. In *ICLR*, 2021.

[6] Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. Learning a unified classifier incrementally via rebalancing. In *CVPR*, 2019.

[7] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In *ICLR*, 2018.

[8] Francesco Marra, Cristiano Saltori, Giulia Boato, and Luisa Verdoliva. Incremental learning for the detection and classification of gan-generated images. In *WIFS*, 2019.

[9] Sudhanshu Mittal, Silvio Galesso, and Thomas Brox. Essentials for class incremental learning. In *CVPR*, 2021.

[10] Lorenzo Pellegrini, Gabriele Graffieti, Vincenzo Lomonaco, and Davide Maltoni. Latent replay for real-time continual learning. In *IROS*, 2020.

[11] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *CVPR*, 2017.

[12] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, 2016.

[13] Shipeng Wang, Xiaorong Li, Jian Sun, and Zongben Xu. Training networks in null space of feature covariance for continual learning. In *CVPR*, 2021.

[14] Sheng-Yu Wang, Oliver Wang, Richard Zhang, Andrew Owens, and Alexei A Efros. CNN-generated images are surprisingly easy to spot... for now. In *CVPR*, 2020.

[15] Bojia Zi, Minghao Chang, Jingjing Chen, Xingjun Ma, and Yu-Gang Jiang. Wilddeepfake: A challenging real-world dataset for deepfake detection. In *ACMMM*, 2020.