A Continual Deepfake Detection Benchmark: Dataset, Methods, and Essentials

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

Figure 1: The suggested continual deepfake detection benchmark (CDDB) aims to promote the research of learning a unified model over a stream of likely heterogeneous deepfakes sequentially. The longest CDDB stream consists of the above 12 types of deepfake sources (Reals: green boundary, Fakes: red boundary).

Abstract

There have been emerging a number of benchmarks and techniques for the detection of deepfakes. However, very few works study the detection of incrementally appearing deepfakes in the real-world scenarios. To simulate the wild scenes, this paper suggests a continual deepfake detection benchmark (CDDB) over a new collection of deepfakes from both known and unknown generative models. The suggested CDDB designs multiple evaluations on the detection over easy, hard, and long sequence of deepfake tasks, with a set of appropriate measures. In addition, we exploit multiple approaches to adapt multiclass incremental learning methods, commonly used in the continual visual recognition, to the continual deepfake detection problem. We evaluate existing methods, including their adapted ones, on the proposed CDDB. Within the proposed benchmark, we explore some commonly known essentials of standard continual learning. Our study provides new insights on these essentials in the context of continual deepfake detection. The suggested CDDB is clearly more challenging than the existing benchmarks, which thus offers a suitable evaluation avenue to the future research. Both data and code are available at https://github.com/Coral79/CDDB.

*Corresponding Author

1. Introduction

Deepfakes (deep learning generated fake images/videos) have become ubiquitous with the advent of increasingly improved deep generative models, such as autoencoders [38], generative adversarial nets (GANs) [25] and generative normalizing flows (Glows) [18]. As a result, there is a growing threat of "weaponizating" deepfakes for malicious purposes, which is potentially detrimental to privacy, society security and democracy [12]. To address this issue, many deepfake detection datasets (e.g.,[40, 44, 46, 21, 64, 19, 31]) and techniques (e.g., [86, 4, 79, 57, 56, 5]) are proposed.

State-of-the-art deep neural networks have made tremendous progress for deepfake detection tasks in a stationary setup, where a large amount of relatively homogeneous deepfakes are provided all at once. In this paper, we study a natural extension from this stationary deepfake detection scenario to a dynamic (continual) setting (Fig.1): a

stream of likely heterogeneous deepfakes appear time by time rather than at once, and the early appeared deepfakes cannot be fully accessed due to the streaming nature of data, privacy concerns, or storage constraints. In such a scenario, at each learning session when trained on a new deepfake detection task, standard neural networks typically forget most of the knowledge related to previously learned deepfake detection tasks [53, 36]. This is essentially one of the most typical continual learning problems that are well-known to result in catastrophic forgetting [63, 60, 52, 9]. Nevertheless, the study on the specific continual deepfake detection (CDD) problem, benchmarks as well as its particular nature remains fairly limited.

In this paper, we establish a challenging continual deepfake detection benchmark (CDDB) by gathering publicly available deepfakes, from both known and unknown generative models. The CDDB gradually introduces deepfakes to simulate the real-world deepfakes' evolution, as shown in Fig.1. The key benchmarking task is to measure whether the detectors are able to incrementally learn deepfake detection tasks without catastrophic forgetting. Up to our knowledge, there exist only two similar benchmarks [53, 36]. Both of these benchmarks are limited as they merely perform CDD on only one deepfake type of known generative models (e.g., GANs or face-swap like deepfake models). As mentioned, the source of deepfakes may not only be unknown but also be of diverse types, in practice. Therefore, we proposed a new CDDB which better resembles the real-world scenarios. Furthermore, our CDDB also categorizes the evaluation protocol into different cases: from easy to hard and short to long CDD sequences. Such categorization allows us to better probe the CDD methods.

We evaluate a set of well-known and most promising existing continual learning methods on the established benchmark. In this process, we first evaluate the popular multi-class incremental learning methods in the CDD settings. Furthermore, we develop multiple approaches to adapt these continual learning methods to the binary CDD problem. Our evaluations also include several other variants which are evaluated for easy/hard and short/long sequences. These exhaustive evaluations offer two major benefits: (a) suitable baselines for the future CDD research; (b) new insights on the established essentials in the context of CDD. In the latter case, we explore particular essentials including (i) knowledge distillation; (ii) class imbalance issue; (iii) memory budget. Notably, our empirical evidence suggests that the existing consideration for class imbalance issues can be clearly hurtful to the CDD performance.

In summary, this paper makes three-fold contributions:

- We propose a realistic and challenging continual deepfake detection benchmark over a collection of public datasets to thoroughly probe CDD methods.
- We comprehensively evaluate existing and adapted methods on the proposed benchmark. These evaluations serve as lock, stock, and barrel of CDD baselines.
- Using the proposed dataset and conducted evaluations, we study several aspects of CDD problem. Our study offers new insights on the CDD essentials.

Dataset	Real Source	Deepfake Source	Continual
Deepfake-TIMIT [40]	VidTIMIT dataset [66]	Known Deepfake tech	✗
UADFW [79]	EBV dataset [44]	Known Deepfake tech	✗
FaceForensics++ [64]	YouTube	Known Deepfake tech	✗
Celab-DF v2 [47]	YouTube	Known Deepfake tech	✗
DFDC [19]	Actors	Known Deepfake tech	✗
WildDeepfake [88]	Internet	Unknown Deepfake tech	✗
WhichFaceReal [3]	Internet	Unknown Deepfake tech	✗
CNNfake [75]	Multi-Datasets	Known Deepfake tech	✗
GANfake [53]	Multi-Datasets	Known Deepfake tech	✓
CoReD [36]	Multi-Datasets	Known Deepfake tech	✓
CDDB (ours)	**Multi-Datasets&Internet**	**Known&unknown tech**	✓

Table 1: Comparison of prominent deepfake datasets. Only CoReD [36], GANfake [53] and our CDDB study continual fake detection benchmarks. However, both CoReD and GANfake merely detect pure GAN-generated images (or pure deepfake-generated videos), while ours studies a high mixture of deepfake sources, which are from either known generative models or unknown ones (i.e., directly from internet).

2. Related Work

Datasets and Benchmarks for Deepfake Detection. To evaluate deepfake detection methods, many datasets and benchmarks have been proposed. For instance, FaceForensic++ [64] contains face videos collected from YouTube and manipulated using deepfake [2], Face2Face [72], Faceswap [1] and Neural Texture [71]. WildDeepfake [88] aims at real-world deepfakes detection by collecting real and fake videos with unknown source model directly from the internet. CNNfake [75] proposes a diverse dataset obtained from various image synthesis methods, including GAN-based techniques (e.g., [33], [7]) and traditional deepfake methods. Table 1 summarizes the prominent benchmark datasets for deepfake detection. The majority of the proposed benchmarks do not include incremental detection in their experimental setups. While few works, namely GANfake [53] and CoReD [36], have addressed the CDD setting, they either only address known GAN-based deepfakes [53] or only treat known GAN fakes and known traditional deepfakes separately [36]. Furthermore, their studied task sequences are generally short (e.g. they consist of 4 or 5 tasks). However, in the real-world scenario, deepfakes might come from known or unknown source models. These models might be based on GANs or traditional methods, and finally, they form a long sequence of tasks evolving through time. To bridge the gap between the current benchmarks and the real-world scenario, our suggested dataset includes a collection of deepfakes from both known or unknown sources. In addition, the suggested benchmark provides three different experimental setups (Easy, Hard, and Long) for a thorough evaluation of CDD methods.

Deepfake Detection Methods. Along with the discussed benchmarks, many approaches have been proposed for deepfake detection (e.g., [64, 45, 57, 4, 54, 56, 53, 75, 24,

76]). These approaches mainly aim at finding generalizable features from a set of available samples that can be used to detect deepfakes at test time. For instance, [64] employs XceptionNet [14], a CNN with separable convolutions and residual connections, pre-trained on ImageNet [16] and fine-tuned for deepfake detection. Similarly, [75] uses ResNet-50 [26] pretrained with ImageNet, and further train it in a binary classification setting for deepfake detection. Different to the aforementioned methods, [53] and [36] address the CDD problem. [53] adapts one of the traditional CIL methods, i.e., incremental classifier and representation learning (iCaRL) [60], through a multi-task learning scheme over both deepfake recognition and detection tasks. To mitigate the catastrophic forgetting, [53] keeps using the original iCaRL[60]'s knowledge distillation loss, which enforces the newly updated network to output close prediction results to those of the network trained on the previous task given the same samples from an exemplar set of old samples. Similarly, CoReD [36] tackles forward learning and backward forgetting with a student-teacher learning paradigm, where the teacher is the model trained for the previous tasks, and the student is the new model being adapted to also include the current task. [36] only uses samples from the current task for the teacher-to-student knowlege distillation. To further alleviate the forgetting, [36] adds a feature-level knowledge distillation loss (i.e., representation loss).

Class-incremental Learning (CIL). In this paper, we focus on studying three prominent categories of CIL methods: *gradient-based*, *memory-based* and *distillation-based*.[1]

Gradient-based methods (e.g., [62, 84, 23, 65, 69, 74]) overcome catastrophic forgetting by minimizing the interference among task-wise gradients when updating network weights. For instance, [74] proposes a null space CIL (NSCIL) method to train the network on the new task by projecting its gradient updates to the null space of the approximated covariance matrix on all the past task data.

Memory-based methods (e.g., [63, 60, 52, 9, 29, 35, 10, 68, 59, 73]) generally mitigate forgetting by replaying a small set of examples from past tasks stored in a memory. For example, on the selected exemplars from previous tasks, latent replay CIL (LRCIL) [59] suggests to replay their latent feature maps from intermediate layers of the model in order to reduce the required memory and computation.

Distillation-based methods (e.g., [48, 60, 28, 8, 78, 82, 70, 50, 55]) apply knowledge distillation [27] between a network trained on previous tasks and a network being trained on current task to alleviate the performance degradation on previous tasks. iCaRL [60] applies the distillation on a exemplar set that is selected by using a herding method, which chooses samples closest to the sample means. [60] also identifies *class imbalance* as a crucial challenge for continual multi-class classification (CMC). To address this, [60] suggests a classification strategy named nearest-mean-of-exemplars. Following the same motivation, LUCIR [28] applies cosine normalization to the magnitude of the parameters in fully-connected (FC) layers. Based on the distillation idea, DyTox [20][2] applies the transformer ConViT [22] to achieve the state-of-the-art CIL.

Discussion. The discussed CIL methods are mostly designed for CMC, which aims to learn a unified classifier for a set of sequentially-encountered classes. As one of the continual binary classification (CBC) problems, CDD can be regarded as a binary task or a set of binary tasks [53], and therefore we further investigate three general approaches of adapting these CIL methods to the CDD problem.

Family	Deepfake Source	Real Source	# Images
GAN Model	ProGAN [33]	LSUN	736.0k
	StyleGAN [34]	LSUN	12.0k
	BigGAN [7]	ImageNet	4.0k
	CycleGAN [87]	Style/object transfer	2.6k
	GauGAN [58]	COCO	10.0k
	StarGAN [13]	CelebA	16.8k
Non-GAN Model	Glow [37]	CelebA	16.8k
	CRN [11]	GTA	12.8k
	IMLE [43]	GTA	12.8k
	SAN [15]	Standard SR benchmark	440
	FaceForensics++ [64]	YouTube	5.4k
Unknown Model	WhichFaceReal [3]	Internet	2.0k
	WildDeepfake [88]	Internet	10.5k

Table 2: A new collection of mixed deepfake sources for the suggested CDDB.

3. Suggested CDD Benchmark

For a more real-world CDD benchmark, we suggest enforcing high heterogeneity over the deepfake data stream. In particular, we construct a new collection of deepfakes by gathering highly heterogeneous deepfakes collected by [75, 53, 3, 88], which are from remarkably diverse resources. Moreover, the deepfakes from [3, 88] has no information about their source generative models, and thus the new data collection reaches a more real-world scenario, which is always full of arbitrary deepfakes from either known or unknown sources.

3.1. Data Collection

The new data collection comprises 3 groups of deepfake sources: 1) GAN models, 2) non-GAN models, and 3) unknown models. Below details the deepfake sources and their associated real sources, which are listed in Table 2.

GAN Models. This group consists of fake images synthesized by *6 GAN models*. ProGAN [33] and StyleGAN [34] are two of the most popular unconditional GANs. They were trained on each category of the dataset LSUN [81] and thus they can produce realistic looking LSUN images. BigGAN [7] is one of the state-of-the-art class-conditional GAN models trained on ImageNet [16]. Moreover, we include three image-conditional GAN models for image-to-

[1] Other CILs are based on regularization or expansion [39, 85, 6, 80, 41, 30, 67].

[2] Other transformer-based CIL methods (e.g. [77, 17, 83, 32, 42]) are emerging.

image translation, namely CycleGAN [87], GauGAN [58] and StarGAN [13]. These models were trained on one style/object transfer task selected from the dataset collected by [87], COCO [49], and CelebA [51], respectively.

Non-GAN Models. This set contains deepfakes generated by *8 non-GAN models*, including Generative Flow (Glow) [37], Cascaded Refinement Networks (CRN) [11], Implicit Maximum Likelihood Estimation (IMLE) [43], second-order attention network (SAN) [15] and 4 other deepfake models (Deepfake [2], Face2Face [72], Faceswap [1] and Neural Texture [71]) from [64]. These models were trained on CelebA [51], GTA [61], a super-resolution dataset and YouTube videos, respectively, for image synthesis.

Unknown Models. This group includes deepfake images from 2 *unknown generative models*, one collected by WildDeepfake [88] and one by WhichFaceReal [3]. They both collect deepfakes and real images/videos directly from the internet. WildDeepfake [88] originally contains deepfake/real videos. As our focus is on the detection of deepfake images, we randomly select a number of frames from each video. This group of models are included to further simulate the real-world, where the source model of the encountered deepfakes might not be known.

3.2. Evaluation Scenarios

From the new collection, a large number of differently-ordered sequences of tasks can be produced to study CDD. In our benchmark, we suggest three different evaluation scenarios: an easy task sequence (*EASY*), a hard task sequence (*HARD*), and a long task sequence (*LONG*). The *EASY* setup is used to study the basic behavior of evaluated methods when they address an easy CDD problem. The *HARD* setup aims to evaluate the performance of competing methods when facing a more challenging CDD problem. The *LONG* setup is designed to encourage methods to better handle long sequences of deepfake detection tasks, where the catastrophic forgetting might become more serious. The three evaluation sequences are detailed as follows:

1. *EASY*: {GauGAN, BigGAN, CycleGAN, IMLE, FaceForensic++, CRN, WildDeepfake}
2. *HARD*: {GauGAN, BigGAN, WildDeepfake, WhichFaceReal, SAN}
3. *LONG*: {GauGAN, BigGAN, CycleGAN, IMLE, FaceForensic++, CRN, WildDeepfake, Glow, StarGAN, StyleGAN, WhichFaceReal, SAN}

More concretely, the procedure on each sequence is to train a given model through the stream of involved training data sequentially, followed by an evaluation using the set of associated test data. Following the common practice in CIL [60, 28, 70, 55] and the suggestion in [75], we allow pre-training evaluated methods over the fake images of ProGAN and the corresponding real samples as a warm-up step.

3.3. Evaluation Metrics

CDD is a continual learning problem, and thus it should study the performance of the evaluated methods in terms of forward learning of each new task, as well as backward forgetting of previous ones. Accordingly, we suggest using the average detection accuracy (*AA*) and the average forgetting degree (*AF*), i.e., mean of backward transfer degradation (BWT) [52], as the evaluation metrics. Formally, we can obtain a test accuracy matrix $B \in \mathbb{R}^{n\times n}$ (i.e., upper triangular matrix), where each entry $B_{i,j}$ indicates the test accuracy of the i-th task after training the j-th task and n is the total number of involved tasks. *AA* and *AF* can be calculated as $AA = \frac{1}{n}\sum_{i=1}^{n} B_{i,n}$, $AF = \frac{1}{n-1}\sum_{i=1}^{n-1} BWT_i$, where $BWT_i = \frac{1}{n-i-1}\sum_{j=i+1}^{n}(B_{i,j} - B_{i,i})$.

As CDD is also a detection problem, we suggest using mean average precision (*mAP*) to measure the performance of evaluated methods in terms of the trade-off between precision and recall, where we regard real samples as negatives and fake samples as positive. Each AP is defined as the area under the precision-recall (PR) curve for deepfake detection over one single deepfake task in the sequence [75]. *mAP* is the average of all the APs calculated over all detection tasks.

As different tasks may contain the same or similar real samples, and some fake samples could be from unknown generative models, we additionally employ the average deepfake recognition accuracy (*AA-M*) to study the challenge for identifying the specific deepfake and real resources. This metric is mainly used for understanding the gap between CMC and CBC.

4. Proposed Benchmarking Methods

The studied CDD problem requires to distinguish real and fake samples from the sequentially occurring sources of real/fake pairs. This section focuses on studying three main adaptions of CIL to better address the CDD problem.

4.1. Problem Definition and Overview

In the CDD problem, deepfakes and their corresponding real images appear sequentially in time, forming the sequence $\mathcal{X} = \{\mathcal{X}_1, \mathcal{X}_2 \ldots, \mathcal{X}_t\}$, where $\mathcal{X}_i = (X_i^R, X_i^F)$ represents the paired set of real and fake images corresponding to source i. At each incremental step t, complete data is only available for the new pair of real and fake image sets $\mathcal{X}_t = (X_t^R, X_t^F)$. For memory-based and distillation-based methods, we additionally use a small amount of exemplar data $\mathcal{P} = \{\mathcal{P}_1, \mathcal{P}_2 \ldots, \mathcal{P}_{t-1}\}$, which is selected from previous data $\mathcal{X} = \{\mathcal{X}_1, \mathcal{X}_2 \ldots, \mathcal{X}_{t-1}\}$ for rehearsal or distillation. The trained models at step t are finally expected to distinguish all the fake and real images corresponding to the sources observed up to and including step t.

The CDD problem can be relaxed to a CMC problem, once each real/fake source is regarded as an independent

class. In this case, traditional CIL methods can be applied to CDD. A *basic CIL system* is to train a network model Θ that consists of a deep feature extractor $\phi(\cdot)$ and a fully-connected (FC) layer $f_c(x)$. Like a standard multi-class classifier, the output logits are processed through an activation function $\varphi(\cdot)$ before classification loss ℓ_{class} is evaluated corresponding to the correct class. To address catastrophic forgetting, many of the state-of-the-art CIL methods apply distillation loss ℓ_{distill} over the exemplar set $\mathcal{P}$ of old samples from previous tasks. Besides, a supplementary loss ℓ_{supp} is often applied to strengthen the classification over multiple classes at $\mathcal{P}$. In general, the loss function for CIL systems can be formulated as:

$$\ell_{\text{CIL}}(\Theta) = \ell_{\text{class}}(\mathcal{X}_t, \Theta) + \gamma_d \ell_{\text{distill}}(\mathcal{P}, \Theta) + \gamma_m \ell_{\text{supp}}(\mathcal{P}, \Theta) \tag{1}$$

where γ_d, γ_m are hyperparameters for the trade-off among the three losses. The classification loss is the usual cross-entropy loss calculated on the new data

$$\ell_{\text{class}}(\mathcal{X}_t, \Theta) = -\sum_{x_i \in \mathcal{X}_t} \sum_{y_j=1}^{k} \delta_{y_j=y_i} \log g_{y_j}(x_i) \tag{2}$$

where $\delta_{y_j=y_i}$ is an indicator to check whether the prediction y_j is in line with the ground truth y_i, k is the total class number, $g_{y_j}(\cdot) = \varphi(\cdot) \circ f_c() \circ \phi(\cdot)$ computes the probability of the class y_j, and $\circ$ is function composition. The distillation term is a *KL-divergence* loss [27] with temperature T:

$$\ell_{\text{distill}}(\mathcal{P}, \Theta) = -\sum_{x_i \in \mathcal{P}} T^2 D_{KL}(g^T(x_i) || \tilde{g}^T(x_i)) \tag{3}$$

where $\tilde{g}(\cdot)$ is the old classifier, which is the one before the current updating phase. As done by [28], the distillation can be also performed over the feature level, i.e. $\ell_{\text{distill}}(\mathcal{P}, \Theta) = -\sum_{x_i \in \mathcal{P}} 1 - \langle \tilde{\phi}(x_i), \phi(x_i) \rangle$, where $\tilde{\phi}(\cdot), \phi(\cdot)$ are the feature extractors of the old and new networks respectively, and $\langle \cdot \rangle$ denotes cosine similarity. The additional term could be a margin ranking loss from [28]:

$$\ell_{\text{supp}}(\mathcal{P}, \Theta) = \sum_{x_i \in \mathcal{P}} \sum_{y_j=1} \max(\tau - \langle \theta^{y_i}, \phi(x_i) \rangle + \langle \theta^{y_j}, \phi(x_i) \rangle, 0) \tag{4}$$

where $\langle \cdot \rangle$ indicates cosine similarity, τ is the margin threshold, θ is class embeddings (i.e., weight parameterization of the FC layer), θ^{y_i} is the ground truth class embedding of x_i, θ^{y_j} is one of the nearest-J class embeddings. The loss pushes nearest classes away from the given class. ℓ_{supp} could be also a divergence loss [20], which acts as an auxiliary classifier encouraging a better diversity among new classes and old ones. More details are presented in [20].

The three losses can be also applied to different data. For example, ℓ_{class} and ℓ_{distill} can be also used to simultaneously learn over both the new and old data, i.e., $\mathcal{P} \cup \mathcal{X}_t$. In this paper, we mainly study four representative methods LRCIL [59], iCaRL [60], LUCIR [59] and DyTox [20], all of which can be formulated by Eqn.1. For LRCIL, γ_d and γ_m are zeros. For iCaRL, γ_m is zero. By comparison, LUCIR and DyTox applies non-zero γ_d and γ_m.

4.2. Adapting CIL to CDD

We study three main adaptations of CIL methods for CDD. The key idea for the adaption is to enforce the classification loss ℓ_{class} and the distillation loss ℓ_{distill} to better fit the binary classification (i.e., detection) task, which is formulated as

$$\ell_{\text{BIL}}(\Theta) = \hat{\ell}_{\text{class}}(\mathcal{X}_t, \Theta) + \gamma_d \hat{\ell}_{\text{distill}}(\mathcal{P}, \Theta) + \gamma_m \ell_{\text{supp}}(\mathcal{P}, \Theta) \tag{5}$$

where $\hat{\ell}_{\text{class}}$ and $\hat{\ell}_{\text{distill}}$ are the main adapted components. Below details three main ways for the adaption on them.

Binary-class (BC) Learning. One of the most straightforward solutions is to change the categorical cross-entropy loss with the binary one:

$$\hat{\ell}_{\text{class}}(\mathcal{X}_t, \Theta) = -\sum_{x_i \in \mathcal{X}_t} \delta_{y=y_i} \log g_y(x_i) + (1 - \delta_{y=y_i}) \log(1 - g_y(x_i)) \tag{6}$$

where $\delta_{y=y_i}$ is an indicator for the ground-truth label y_i, $g_y(x_i)$ applies Sigmoid function $\varphi(x_i)$ instead, calculating the probability of the given sample x_i being a fake sample. In addition, the distillation loss $\hat{\ell}_{\text{distill}}$ is based on the binary prediction. For the final classification, we apply the Sigmoid function based results. As ℓ_{supp} is originally designed for better multi-class classification, it can be ignored in the binary adaptation.

Multi-class (MC) Learning. For this approach, we use the original classification, distillation, and supplementary losses, i.e. $\hat{\ell}_{\text{class}} = \ell_{\text{class}}$, $\hat{\ell}_{\text{distill}} = \ell_{\text{distill}}$, $\hat{\ell}_{\text{supp}} = \ell_{\text{supp}}$. We regard each real/fake class from different tasks as an independent class. For classification, we apply $\hat{y}_i = \arg\max_{y_j} g_{y_j}(x_i)$, given a sample x_i. If $\hat{y}_i$ is one of the fake/real classes, we will predict x_i to fake/real.

Multi-task (MT) Learning. Another adaption is to apply a multi-task learning formulation. In particular, both the multi-class classification and the binary-class classification (i.e., detection) tasks are managed by the same classifier $g(\cdot)$. To this end, we adapt the classification loss by adding a binary cross-entropy term to the categorical cross-entropy

$$\hat{\ell}_{\text{class}}(\mathcal{X}_t, \Theta) = (1 - \lambda) \ell_{\text{class}}(\mathcal{X}_t, \Theta) + \lambda \ell'_{\text{class}}(\mathcal{X}_t, \Theta) \tag{7}$$

where ℓ_{class} is the regular multi-class classification task loss Eqn.2, the binary-class classification task loss ℓ'_{class} is computed by taking into account the activations, $g(\cdot)$, of all the classes, separately for the fake and real classes, and the hyperparameter λ is for the balance between these two task-based losses. Formally, ℓ'_{class} is computed by

$$\ell'_{\text{class}}(\mathcal{X}_t, \Theta) = \sum_{x_i \in (\mathcal{X}_t)} \delta_{\text{Y=F}} d_F(x_i) + \delta_{Y=R} d_R(x_i) \tag{8}$$

where $d_F(x_i)$ and $d_R(x_i)$ are designed for the aggregation over all fake classes and all real classes respectively. In this paper, we study the following four aggregation approaches: (1) *SumLog* (Eqn.9a) that is proposed in [53], (2) *SumLogit* (Eqn.9b), (3) *SumFeat* (Eqn.9c), and (4) *Max* (Eqn.9d).

$$d_F(x_i) = \sum_{y\in\{fake\}} \log g_y(x_i), \qquad d_R(x_i) = \sum_{y\in\{real\}} \log g_y(x_i) \quad (9a)$$

$$d_F(x_i) = \log \sum_{y\in\{fake\}} g_y(x_i), \qquad d_R(x_i) = \log \sum_{y\in\{real\}} g_y(x_i) \quad (9b)$$

$$d_F(x_i) = \log g_y(\sum_{y\in\{fake\}} x_i), \qquad d_R(x_i) = \log g_y(\sum_{y\in\{real\}} x_i) \quad (9c)$$

$$d_F(x_i) = \max_{y\in\{fake\}} (\log g_y(x_i)), \quad d_R(x_i) = \max_{y\in\{real\}} (\log g_y(x_i)) \quad (9d)$$

where $y \in \{fake\}$ indicates all the associated fake classes, and $y \in \{real\}$ corresponds to all the real classes.

For the MT case, we use the original distillation and supplementary losses, i.e. $\hat{\ell}_{\text{distill}} = \ell_{\text{distill}}$, $\hat{\ell}_{\text{supp}} = \ell_{\text{supp}}$, where ℓ_{distill} and ℓ_{supp} are computed using Eqn.3 and Eqn.4, respectively. As done in the MC case, we use $\hat{y}_i = \arg\max_{y_j} g_{y_j}(x_i)$ for the final classification.

5. Benchmarking Results

We evaluate three families of CIL methods with our exploited variants (using BC, MC, MT) on the suggested CDD benchmark for the three scenes (Sec.3.2): 1) *EASY*, 2) *HARD*, and 3) *LONG*, with the introduced measures (Sec.3.3). The three sets of state-of-the-art CIL methods are: 1) Gradient-based: NSCIL [74], 2) Memory-based: LRCIL [59], and 3) Distillation-based: iCaRL [60], LUCIR [28] and DyTox [20]. Besides, we evaluated the multi-task variant of iCaRL (iCaRL-SumLog)[3] [53]. We adapted the top-4 best CIL methods (i.e., LRCIL, iCaRL, LUCIR, DyTox) using BC, MC, MT. Note that it is non-trivial to adapt DyTox with BC, as the loss is tightly constrained for multiple-class classification. Thus, we do not evaluate its BC variant. For a fair comparison, except for DyTox that is based on an ImageNet pre-trained transformer ConViT [22], we employ the state-of-the-art deepfake CNN detector (CNNDet) [75] that applies a ResNet-50 pretrained on ImageNet [16] and ProGAN [33] as the backbone for all the rest methods over the proposed CDDB. For most methods, we used their official codes, and tuned their hyperparameters for better performances. For a consistency, we empirically set the MT learning hyperparameter as $\lambda = 0.3$ for all the MT methods. For *EASY, HARD* and *LONG*, we assign the same memory budget (i.e., 1500) for all those methods that need a memory to save exemplars. Additionally, we study three reduced memory budgets (i.e., 1000, 500, 100) for *HARD*. For all the evaluations, we evaluate the joint training methods using CNNDet/ConViT with either binary classification loss (CNNDet-Binary) or multi-class classification loss (CNNDet/ConViT-Multi) to study the approximated upper bound for the incremental learning methods. The detailed settings, the parameter setups, and more empirical studies are presented in Suppl. Material[4].

[3] We overlooked the method [36], as its code is not publicly available.

5.1. EASY Evaluation

Table 3 reports the benchmarking results of the evaluated methods for the CDDB *EASY*. Table 4 studies other essential components for the CDD problem.

Vanilla Deepfake Detection vs. Continual one (CDD). As discussed in Sec.1 and Sec.2, most of deepfake detection techniques are designed to single/stationary deepfake detection tasks. We follow CNNDet[75]'s suggestion to train it on the ProGAN dataset, and applied the pre-trained model to the seven tasks directly in the EASY evaluation. Thus, we further name this method as CNNDet[75]-Zeroshot. Besides, we also compare the one that finetunes the pre-trained CNNDet on each new task, due to the access limit to the old task data. The same training strategies are also applied to one of the state-of-the-art transformers, i.e., ConViT [22]. By comparing the continual learning setups (BC, MC, MT), we can find that all the zeroshot and finetune setups perform clearly worse, showing that the necessity of applying the continual learning approaches to the CDD problem.

Basic Findings on Essentials for CDD. 1) Due to the serious *forgetting* problem, *finetuning* the CNNDet/ConViT methods works worse than the pretrained CNNDet/ConViT models over ProGAN. 2) The clearly inferior performances of the *state-of-the-art gradient-based method NSCIL* might be because its null space cannot be approximated well on the highly heterogeneous fakes and reals. 3) LRCIL merely performs a rehearsal on the data to address forgetting, and thus we added a *knowledge distillation* (KD) term to LRCIL (i.e., LRCIL-KD), which further improves AA with a marginally worse AA-M. (see Table 4). 4) *class imbalance issue* is not so important to CDD. From Table 4, we discover that its used cosine normalization based fully connected layer (CosFC) [28] clearly hurts the performance (see the comparison LUCIR-CosFC vs. LUCIR-LinFC in Table 4). CosFC is proposed to address the *class imbalance issue*, with which the test samples are often predicted into new classes in the CMC context. However, predicting the test fakes into the new fake class is acceptable for our studied CDD problem. Therefore, the normalization techniques is very likely to hurt the CDD performance. Based on the observation, we replace CosFC with regular FC (LinFC) in all LUCIR based variants for better performances.

CNN vs. ViT. The CNN-based methods are mostly outperformed by the ViT-based methods (ConViT and DyTox), as we can see in Table 3. Nevertheless, we should notice that the parameter size (≈86M) of the used ConViT is about 3 times larger than that (≈25M) of CNNDet (ResNet50). The study provides two choices when we address the CDD prob-

[4] The supp. metrial also studies GANfake [53]. As it does not release the detailed train/val/test splits, we can only empirically study our own splits following the description in the oiriginal paper.

Learning Sys.	Evaluated Method	CDDB-EASY1500 Task1	Task2	Task3	Task4	Task5	Task6	Task7	AA	AF	AA-M	mAP
Baseline	CNNDet[75]-Zeroshot	63.85	68.75	71.95	60.93	91.31	60.93	49.01	66.68	NA	NA	79.02
	CNNDet[75]-Finetune	57.25	51.00	57.63	47.81	80.41	48.00	78.77	60.28	-14.29	NA	61.67
	ConViT[22]-Zeroshot	89.55	75.38	94.66	98.51	80.22	96.67	49.25	83.46	NA	NA	61.04
	ConViT[22]-Finetune†	51.45	49.25	52.86	51.49	77.63	55.49	86.28	60.64	-42.75	NA	56.92
Binary-class (BC) learning	NSCIL[74]-**Sigmoid***	48.35	50.25	49.43	56.58	56.56	70.73	57.63	55.65	-42.04	NA	63.50
	LRCIL[59]-**Sigmoid***	83.00	88.00	82.82	96.20	79.02	97.14	62.82	84.14	-9.15	NA	91.37
	iCaRL[60]-**Sigmoid***	76.90	80.00	88.93	99.41	85.03	99.45	76.64	87.05	-10.64	NA	92.25
	LUCIR[28]-**Sigmoid***	90.60	91.05	90.46	99.80	91.04	99.80	75.38	91.16	-4.76	NA	**95.94**
Multi-class (MC) learning	NSCIL[74]	46.80	52.50	47.90	45.26	35.21	51.33	58.85	48.26	-50.88	8.41	44.26
	LRCIL[59]	83.50	77.88	90.84	98.90	84.75	98.86	65.92	85.81	-5.88	67.11	92.63
	iCaRL[60]	77.50	71.38	91.22	99.57	95.66	99.92	78.28	87.65	-9.41	65.39	94.12
	LUCIR[28]-**LinFC**	91.60	89.12	92.56	99.76	94.45	99.80	71.21	91.21	-2.88	74.62	94.75
	DyTox[20]	98.30	94.25	98.85	100.00	95.66	100.00	85.94	96.14	-1.24	83.66	93.90
Multi-task (MT) learning	LRCIL[59]-SumLog[53]	86.65	85.63	91.79	99.22	88.72	99.57	68.27	88.76	-3.99	66.03	93.95
	iCaRL[60]-SumLog[53]	74.40	78.38	88.36	99.65	92.24	99.69	79.84	87.05	-10.72	71.41	93.02
	LUCIR[28]-SumLog[53]	88.70	88.62	93.89	99.37	94.82	99.80	75.71	91.56	-3.65	74.47	95.47
	DyTox[20]-SumLog[53]‡	98.30	95.00	99.43	100.00	96.30	100.00	85.89	96.42	-0.72	83.94	94.01
	LRCIL[59]-**SumLogit**	86.95	88.12	92.75	99.45	88.35	99.45	70.24	**89.33**	-4.27	68.00	**94.84**
	iCaRL[60]-**SumLogit**	85.25	86.12	88.93	99.65	92.98	99.80	80.27	**90.43**	-6.12	74.18	**95.27**
	LUCIR[28]-**SumLogit**	89.95	89.62	94.47	99.65	95.75	99.80	74.79	**92.00**	-3.13	73.55	95.26
	DyTox[20]-**SumLogit**	98.30	94.75	99.05	100.00	96.86	100.00	86.82	**96.54**	-0.82	84.42	**94.23**
	LRCIL[59]-**SumFeat**	84.85	85.13	92.56	99.26	87.15	99.45	69.90	87.81	-5.30	66.40	93.73
	iCaRL[60]-**SumFeat**	77.90	84.25	90.84	99.65	82.72	99.69	79.11	87.74	-9.52	68.15	93.88
	LUCIR[28]-**SumFeat**	90.05	89.38	94.85	99.92	95.01	99.92	73.53	91.81	-3.12	74.23	95.66
	DyTox[20]-**SumFeat**	98.05	93.63	99.24	100.00	96.12	100.00	86.67	96.24	-1.13	83.98	94.20
	LRCIL[59]-**Max**	87.80	89.00	92.18	67.96	87.15	99.22	69.85	89.16	-4.43	69.38	94.60
	iCaRL[60]-**Max**	82.35	87.00	92.94	99.76	91.77	99.73	79.74	89.92	-6.79	73.47	94.87
	LUCIR[28]-**Max**	89.85	91.25	94.08	99.53	92.51	99.65	72.61	91.21	-4.00	74.06	95.41
	DyTox[20]-**Max**	98.80	92.63	99.05	100.00	96.12	100.00	86.23	96.12	-1.19	84.14	94.08
Joint Training	CNNDet [75]-Binary	98.65	98.38	96.37	100.00	95.19	100.00	79.59	95.20	NA	NA	98.36
	CNNDet [75]-Multi	95.70	97.00	95.80	99.96	96.30	99.96	75.28	94.29	NA	79.28	97.25
	DyTox[20]-Multi	99.40	96.37	98.28	100.00	94.64	100.00	80.22	95.53	NA	81.87	95.34

Table 3: Benchmarking results on the suggested CDDB's *EASY* evaluation. CNNDet/ConViT-Zeroshot is merely trained on ProGAN, and CNNDet/ConViT-Finetune is tuned over the 7 tasks. ConViT[22]-Finetune†: low AA/mAP seems attributed to huge forgetting. Sigmoid*: applying Sigmoid function based classification loss. SumLog[53]‡: most cases fail, only $\lambda = 0.0001$ works. AA: Average Accuracy for deepfake detection, AF: Average Forgetting degree, AA-M: Average Accuracy for deepfake recognition, mAP: mean Average Precision. green: LRCIL, blue: iCaRL, red: LUCIR, cyan: DyTox. **Bold**: best **green/blue/red/cyan** results, Underline: second best green/blue/red/cyan results.

Learning System	Evaluated Method	CDDB-EASY1500 AA	AF	AA-M
Multi-class	LRCIL[59]	85.81	-5.88	67.11
	LUCIR(CosFC)[28]	87.24	-10.32	71.42
	LRCIL[59]-**KD**	86.85	-5.50	66.57
	LUCIR[28]-**LinFC**	91.21	-2.88	95.41

Table 4: Evaluation results on essentials of CILs on the *EASY* evaluation. AA: Average Accuracy for detection, AF: Average Forgetting degree, AA-M: Average recognition Accuracy. Results of LRCIL and LUCIR are in green, red respectively.

lem. One is the family of light CNNDet models, and the other one is the group of heavy ConViT mdoels.

BC vs. MC vs. MT Learning Systems. Table 3 reflects that DyTox gets almost saturated performances on the EASY eveluation (even higher than its upperbound DyTox-Multi). Also we discover that its variants performs almost the same in terms of AA, while DyTox-MC works generally worse than DyTox-MT in the reduced memory case (see Tabel 7). Hence, we turn to concentrate on the comparison over the lighter CNN-based methods. Except for the case on LRCIL, Table 3 shows that the BC variants of the rest models (like iCaRL and LUCIR) perform very comparably with their corresponding MC models in terms of AA and AF, showing that the fine-grained classification benefits the detection. It is also good for the MC method to eventually label an image as a fake if the classifier decides for any of the fake classes. By comparison, most of the MT variants of LRCIL, iCaRL and LUCIR work better than (or at least comparable with) their corresponding BC and MC models in terms of AA, and some of MTs perform clearly better than their corresponding MCs in terms of AA-M. This mainly stems from the natural complementary properties between the fine-grained multi-class separation and the coarse-grained binary-class cohesion, and most of the suggested MT methods balance them well.

SumLog vs. SumLogit vs. SumFeat vs. Max. From Table 3, we can see that the *SumLogit* and *Max* variants mostly work better than the original *SumLog* [53] for the two main measures, i.e., AA and mAP. This is mainly because of consistency with final classifier's operation, which applies $argmax$ to the resulting logits. By comparison, the *SumFeat* variants merely perform better than the original *SumLog* for LUCIR in terms of AA and mAP. This might be because, different from LRCIL and iCaRL, it additionally applies the metric learning like loss (i.e., margin ranking loss) and thus the resulting features might be more discriminative for the aggregation to address the binary classification.

5.2. HARD and LONG Evaluations

We select the top-2 BC, MC and MT models in terms of AA for LRCIL, iCaRL and LUCIR from the *EASY* evaluation for bechmarking *HARD* and *LONG*.

EASY vs. HARD vs. LONG. As the LONG evaluation includes both easy and hard tasks, the evaluated methods generally get higher scores than those in HARD. Besides, LONG's overall AAs and mAPs are visibly worse than those on EASY, meaning that incremental learning over a longer sequence is more challenging. Interestingly, the AFs are not clearly worse, showing the forgetting issue is not serious in this context. By comparison, HARD's overall AAs, mAPs and AA-Ms are clearly lower than those of EASY and LONG, because it contains more challenging tasks such as WildDeepfake, WhichFaceReal that are from the wild scenes and SAN that merely has a small data for training.

Memory Budget. As all the evaluated methods require a memory to utilize an exemplar set of old samples to address the forgetting problem, we evaluate the performance as a function of the memory budget. Table 6 summarizes the

		CDDB-HARD1500				CDDB-LONG1500			
Learning System	Evaluated Method	AA	AF	AA-M	mAP	AA	AF	AA-M	mAP
Binary-class	LRCIL [59]-**Sigmoid***	74.07	-5.43	NA	80.40	87.06	-4.79	NA	91.81
	iCaRL [60]-**Sigmoid***	80.52	-7.89	NA	87.76	87.12	-6.92	NA	91.68
	LUCIR [28]-**Sigmoid***	83.17	-4.45	NA	**89.72**	87.19	-6.91	NA	92.11
Multi-class	LRCIL [59]	74.22	-5.41	57.74	80.21	86.40	-4.65	62.32	91.28
	iCaRL [60]	76.72	-7.79	67.46	85.05	82.50	-10.76	53.49	89.63
	LUCIR [28]-**LinFC**	82.33	-3.12	66.27	86.73	86.95	-6.77	69.74	91.43
	DyTox [20]	88.46	-0.72	79.68	88.06	93.34	-1.67	78.62	**94.08**
Multi-task	LRCIL [59]-**SumLogit**	**77.28**	-2.70	60.22	**81.38**	87.93	-2.99	65.60	92.45
	iCaRL [60]-**SumLogit**	81.16	-7.84	69.00	**90.22**	88.68	-5.12	70.51	93.10
	LUCIR [28]-**SumLogit**	**83.42**	-3.28	65.31	87.89	**88.57**	-5.78	71.55	**92.83**
	DyTox [20]-**SumLogit**	88.60	-0.62	80.59	89.15	93.80	-1.41	78.86	92.47
	LRCIL [59]-**Max**	76.93	-2.55	59.20	81.20	**88.49**	-2.64	65.67	**92.49**
	iCaRL [60]-**Max**	**81.28**	-8.95	60.64	88.61	**89.05**	-5.24	71.35	**94.03**
	LUCIR [28]-**SumFeat**	82.14	-2.90	65.59	87.39	88.40	-5.30	70.99	92.70
	DyTox [20]-**SumFeat**	**88.84**	-0.76	80.57	**89.45**	**94.04**	-1.20	79.62	**94.08**
Joint Training	CNNDet [75]-Binary	85.29	NA	NA	91.34	93.17	NA	NA	94.69
	CNNDet [75]-Multi	84.63	NA	70.59	90.10	92.30	NA	78.71	94.82
	DyTox [20]-Multi	87.93	NA	75.46	87.53	95.31	NA	79.98	93.81

Table 5: Benchmarking results on CDDB's *HARD* and *LONG*. AA: Average detection Accuracy , AF: Average Forgetting, AA-M: Average Accuracy for recognition. mAP: mean Average Precision. green: LRCIL, blue: iCaRL, red: LUCIR, cyan: DyTox. **Bold**: best results, Underline: second best results.

		Reduced Memory Budgets for **CDDB-HARD**							
		1000				500			
Learning System	Evaluated Method	AA	AF	AA-M	mAP	AA	AF	AA-M	mAP
Binary-class	LRCIL [59]-**Sigmoid***	75.61	-6.35	NA	80.30	73.51	-8.96	NA	80.36
	iCaRL [60]-**Sigmoid***	76.99	-10.28	NA	85.46	72.91	-16.01	NA	81.34
	LUCIR [28]-**Sigmoid***	81.75	-6.12	NA	88.46	78.99	-9.43	NA	85.81
Multi-class	LRCIL [59]	**76.39**	-4.39	55.10	80.45	73.18	-9.01	46.69	79.90
	iCaRL [60]	72.37	-13.04	41.43	85.25	71.75	-13.52	42.07	**83.61**
	LUCIR [28]-**LinFC**	81.57	-3.09	65.96	85.97	78.57	-6.97	59.87	84.87
	DyTox [20]	**88.64**	-0.86	80.02	89.41	87.27	-2.02	76.54	87.77
Multi-task	LRCIL [59]-**SumLogit**	75.14	-3.53	56.31	81.05	73.05	-9.08	49.99	**80.71**
	iCaRL [53]-**SumLogit**	**79.76**	-8.73	66.66	**87.75**	**73.98**	-14.50	58.44	81.00
	LUCIR [28]-**SumLogit**	**82.53**	-5.34	72.00	88.14	79.70	-8.18	65.86	**88.08**
	DyTox [20]-**SumLogit**	87.39	-1.375	78.60	88.77	87.64	-1.92	77.65	88.14
	LRCIL [59]-**Max**	74.52	-5.18	43.65	**81.39**	**74.01**	-8.62	49.59	78.72
	iCaRL [53]-**Max**	78.55	-8.69	65.34	86.78	73.70	-14.65	58.85	**81.06**
	LUCIR [28]-**SumFeat**	82.35	-4.76	71.22	**88.93**	**80.77**	-7.85	70.03	87.84
	DyToX [20]-**SumFeat**	88.59	-1.27	80.57	**89.70**	**87.73**	-1.80	77.88	87.89

Table 6: Benchmarking results on reduced memories for the suggested CDDB's *HARD* evaluation. green: LRCIL, blue: iCaRL, red: LUCIR, cyan: DyTox. **Bold**: best results, Underline: second best results.

		100 Memory Budgets for **DyTox**							
		EASY				HARD			
Learning System	Evaluated Method	AA	AF	AA-M	mAP	AA	AF	AA-M	mAP
Multi-class	DyTox [20]	91.99	-6.33	74.22	89.41	84.53	-4.26	70.08	85.45
Multi-task	DyTox [20]-**SumLogit**	**93.58**	-4.33	76.45	90.30	**86.38**	-3.38	68.94	86.23
	DyTox [20]-**SumFeat**	93.32	-4.61	74.67	**91.37**	85.23	-4.74	70.31	84.40
	DyTox [20]-**Max**	93.47	-4.71	71.24	90.71	86.35	-2.68	70.85	**86.47**

Table 7: Benchmarking results on the reduced memory (memory=100) for DyTox's *HARD* and *EASY* evaluation. **bold**: best results, underline: second best results

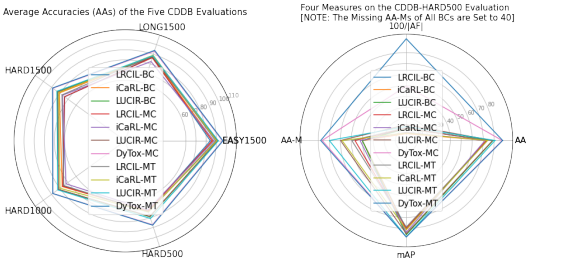

Figure 2: **Left**: Radar plot (the bigger covering area the better) on AAs of the evaluated methods for CDDB's *EASY1500, LONG1500, HARD1500, HARD1000, HARD500*. **Right**: Radar plot on AAs, AFs, mAPs and AA-Ms of the evaluated methods for *HARD500*. HARDm: HARD with memory budget= m, BC/MC/MT: binary-class/multi-class/multi-task 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.

results for the HARD evaluation with memory budget being 1000 and 500. The results again demonstrates the suggested MTs mostly outperform their competitors in terms of AA, mAP, AA-M. The LUCIR-SumLogit and LUCIR-SumFeat generally performs the best. The memory reduction from 1500 to 500 result in clear drops in the terms of AAs, AA-Ms, AFs and mAPs, showing the memory budget is one of the most essential factors for the CDD problem. Table 7 studies the performances of DyTox and its variants for the further reduced memory (memory=100). The results demonstrate that most MT variants of DyTox clearly outperform its MC variant when the memory is very limited.

Overall Spotlight. Fig.2 shows a radar plot of the five CDDB evaluations in terms of AAs, and a radar plot of the HARD500 evaluations in terms of the suggested four measures (i.e., AAs, AFs, mAP, AA-Ms). Our CDDB-HARD uses more advanced CIL method (DyTox) to reach the highest AA of around 86% when memory=100 (/500), while the existing CDD benchmarks (GANfake [53] and CoReD [36]) utilizes earlier CIL method (like iCaRL) to get highest AAs of above 96% (memory=512) and 86% (memory=0) respectively. The higher challenge is mainly attributed to the suggested CDD benchmark on the collection of known and unknown deepfakes, whereas they proposed performing CDD on either pure GAN-generated fakes or pure traditional deepfake-produced images. Besides, the rest three measures (i.e., AFs, mAP, AA-Ms) are overlooked by the two benchmarks, which however are valuable to study CDD. The low scores on them further imply that the suggested CDD benchmark is challenging and thus will open up promising directions for more solid research on the CDD problem.

6. Conclusion and Outlook

The continual deepfake detection benchmark proposed in this paper attempts to bring attention to the real world problem of detecting evolving deepfakes. In this front, difficult cases of deepfakes, long-term aspects of continual learning, and the varieties of deepfake sources are considered. To invite novel solutions, their evaluation protocols and several baseline methods are established by borrowing the most promising ones from the literature. The proposed dataset, evaluation protocol, and established baselines will allow researchers to quickly test their creative ideas and probe them against the existing ones. Additionally, through our evaluations, we are able to provide empirical study to analyze and discuss the common practices of continual learning in the context of the CDD problem.

Our experiments show that the proposed CDDB is clearly more challenging than the existing benchmarks. Hence we believe it will open up promising new directions of solid research on continual deepfake detection. As future works, other essentials remain to be studied, including 1) exemplar selection, 2) knowledge distillation, and 3) data augmentation. Due to the nature of CDD, we recommend using the new collection of deepfakes as an open set. Accordingly, we welcome external contributions to include any newly appeared deefake resources with the benchmark to simulate the real-world scenarios.

Acknowledgments. This work was supported by the Singapore Ministry of Education (MOE) Academic Research Fund (AcRF) Tier 1 grant (MSS21C002). This work was also supported in part by the ETH Zürich Fund (OK), an Amazon AWS grant, and an Nvidia GPU grant. The authors thank Ankush Panwar for processing the WildDeepFake dataset.

References

[1] Deepfakes faceswap github repository. *https://github.com/MarekKowalski/FaceSwap*.

[2] Deepfakes github. *https://github.com/deepfakes/faceswap*.

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

[4] Darius Afchar, Vincent Nozick, Junichi Yamagishi, and Isao Echizen. Mesonet: a compact facial video forgery detection network. In *WIFS*. IEEE, 2018.

[5] Shruti Agarwal, Hany Farid, Yuming Gu, Mingming He, Koki Nagano, and Hao Li. Protecting world leaders against deep fakes. In *CVPR workshops*, 2019.

[6] Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. Memory aware synapses: Learning what (not) to forget. In *ECCV*, 2018.

[7] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. 2019.

[8] Francisco M Castro, Manuel J Marín-Jiménez, Nicolás Guil, Cordelia Schmid, and Karteek Alahari. End-to-end incremental learning. In *ECCV*, 2018.

[9] Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with a-gem. *arXiv preprint arXiv:1812.00420*, 2018.

[10] Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet Kumar Dokania, Philip HS Torr, and Marc'Aurelio Ranzato. Continual learning with tiny episodic memories. *arXiv preprint arXiv:1902.10486*, 2019.

[11] Qifeng Chen and Vladlen Koltun. Photographic image synthesis with cascaded refinement networks. In *ICCV*, 2017.

[12] Bobby Chesney and Danielle Citron. Deep fakes: A looming challenge for privacy, democracy, and national security. *Calif. L. Rev.*, 107:1753, 2019.

[13] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *CVPR*, 2018.

[14] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *CVPR*, 2017.

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

[16] 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.

[17] Jieren Deng, Jianhua Hu, Haojian Zhang, and Yunkuan Wang. Incremental prototype prompt-tuning with pre-trained representation for class incremental learning. *arXiv preprint arXiv:2204.03410*, 2022.

[18] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real NVP. In *ICLR*, 2017.

[19] Brian Dolhansky, Joanna Bitton, Ben Pflaum, Jikuo Lu, Russ Howes, Menglin Wang, and Cristian Canton Ferrer. The deepfake detection challenge (dfdc) dataset. *arXiv preprint arXiv:2006.07397*, 2020.

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

[21] Nick Dufour and Andrew Gully. Contributing data to deepfake detection research. *Google AI Blog*, 2019.

[22] Stéphane d'Ascoli, Hugo Touvron, Matthew L Leavitt, Ari S Morcos, Giulio Biroli, and Levent Sagun. Convit: Improving vision transformers with soft convolutional inductive biases. In *ICLR*, 2021.

[23] Mehrdad Farajtabar, Navid Azizan, Alex Mott, and Ang Li. Orthogonal gradient descent for continual learning. In *AISTATS*, 2020.

[24] Candice R Gerstner and Hany Farid. Detecting real-time deep-fake videos using active illumination. In *CVPR*, 2022.

[25] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. In *NeurIPS*, 2014.

[26] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[27] Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

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

[29] Wenpeng Hu, Zhou Lin, Bing Liu, Chongyang Tao, Zhengwei Tao, Jinwen Ma, Dongyan Zhao, and Rui Yan. Overcoming catastrophic forgetting for continual learning via model adaptation. In *ICLR*, 2018.

[30] Ching-Yi Hung, Cheng-Hao Tu, Cheng-En Wu, Chien-Hung Chen, Yi-Ming Chan, and Chu-Song Chen. Compacting, picking and growing for unforgetting continual learning. *NeurIPS*, 2019.

[31] Liming Jiang, Ren Li, Wayne Wu, Chen Qian, and Chen Change Loy. Deeperforensics-1.0: A large-scale dataset for real-world face forgery detection. In *CVPR*, 2020.

[32] Patrick Kahardipraja, Brielen Madureira, and David Schlangen. Towards incremental transformers: An empirical analysis of transformer models for incremental nlu. *arXiv preprint arXiv:2109.07364*, 2021.

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

[34] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, 2019.

[35] Ronald Kemker and Christopher Kanan. Fearnet: Brain-inspired model for incremental learning. In *ICLR*, 2018.

[36] Minha Kim, Shahroz Tariq, and Simon S Woo. Cored: Generalizing fake media detection with continual representation using distillation. In *ACMMM*, 2021.

[37] Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. *NeurIPS*, 2018.

[38] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *ICLR*, 2014.

[39] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

[40] Pavel Korshunov and Sébastien Marcel. Deepfakes: a new threat to face recognition? assessment and detection. *arXiv preprint arXiv:1812.08685*, 2018.

[41] Sang-Woo Lee, Jin-Hwa Kim, Jaehyun Jun, Jung-Woo Ha, and Byoung-Tak Zhang. Overcoming catastrophic forgetting by incremental moment matching. *NeurIPS*, 30, 2017.

[42] Duo Li, Guimei Cao, Yunlu Xu, Zhanzhan Cheng, and Yi Niu. Technical report for iccv 2021 challenge sslad-track3b: Transformers are better continual learners. *arXiv preprint arXiv:2201.04924*, 2022.

[43] Ke Li, Tianhao Zhang, and Jitendra Malik. Diverse image synthesis from semantic layouts via conditional imle. In *ICCV*, 2019.

[44] Yuezun Li, Ming-Ching Chang, and Siwei Lyu. In ictu oculi: Exposing ai created fake videos by detecting eye blinking. In *WIFS*, 2018.

[45] Yuezun Li and Siwei Lyu. Exposing deepfake videos by detecting face warping artifacts. *arXiv preprint arXiv:1811.00656*, 2018.

[46] Yuezun Li, Xin Yang, Pu Sun, Honggang Qi, and Siwei Lyu. Celeb-df: A new dataset for deepfake forensics. *onikle.com*, 2019.

[47] Yuezun Li, Xin Yang, Pu Sun, Honggang Qi, and Siwei Lyu. Celeb-df: A large-scale challenging dataset for deepfake forensics. In *CVPR*, 2020.

[48] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017.

[49] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014.

[50] Yaoyao Liu, Yuting Su, An-An Liu, Bernt Schiele, and Qianru Sun. Mnemonics training: Multi-class incremental learning without forgetting. In *CVPR*, 2020.

[51] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *ICCV*, 2015.

[52] David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. *NIPS*, 2017.

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

[54] Falko Matern, Christian Riess, and Marc Stamminger. Exploiting visual artifacts to expose deepfakes and face manipulations. In *WACV Workshops*, 2019.

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

[56] Huy H Nguyen, Fuming Fang, Junichi Yamagishi, and Isao Echizen. Multi-task learning for detecting and segmenting manipulated facial images and videos. *arXiv preprint arXiv:1906.06876*, 2019.

[57] Huy H Nguyen, Junichi Yamagishi, and Isao Echizen. Use of a capsule network to detect fake images and videos. *arXiv preprint arXiv:1910.12467*, 2019.

[58] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *CVPR*, 2019.

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

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

[61] Stephan R Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for data: Ground truth from computer games. In *ECCV*, 2016.

[62] 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.

[63] Anthony Robins. Catastrophic forgetting, rehearsal and pseudorehearsal. *Connection Science*, 7(2):123–146, 1995.

[64] Andreas Rossler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Nießner. Faceforensics++: Learning to detect manipulated facial images. In *ICCV*, 2019.

[65] Gobinda Saha, Isha Garg, and Kaushik Roy. Gradient projection memory for continual learning. In *ICLR*, 2020.

[66] C. Sanderson and B.C. Lovell. Multi-region probabilistic histograms for robust and scalable identity inference. *Lecture Notes in Computer Science*, 2009.

[67] Mohamad Shahbazi, Zhiwu Huang, Danda Pani Paudel, Ajad Chhatkuli, and Luc Van Gool. Efficient conditional gan transfer with knowledge propagation across classes. In *CVPR*, 2021.

[68] Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. *arXiv preprint arXiv:1705.08690*, 2017.

[69] Shixiang Tang, Dapeng Chen, Jinguo Zhu, Shijie Yu, and Wanli Ouyang. Layerwise optimization by gradient decomposition for continual learning. In *CVPR*, 2021.

[70] Xiaoyu Tao, Xinyuan Chang, Xiaopeng Hong, Xing Wei, and Yihong Gong. Topology-preserving class-incremental learning. In *ECCV*, 2020.

[71] Justus Thies, Michael Zollhöfer, and Matthias Nießner. Deferred neural rendering: Image synthesis using neural textures. *ACM Transactions on Graphics (TOG)*, 38(4):1–12, 2019.

[72] Justus Thies, Michael Zollhofer, Marc Stamminger, Christian Theobalt, and Matthias Nießner. Face2face: Real-time face capture and reenactment of rgb videos. In *CVPR*, 2016.

[73] Gido M van de Ven, Zhe Li, and Andreas S Tolias. Class-incremental learning with generative classifiers. In *CVPR*, 2021.

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

[75] 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.

[76] Xueyu Wang, Jiajun Huang, Siqi Ma, Surya Nepal, and Chang Xu. Deepfake disrupter: The detector of deepfake is my friend. In *CVPR*, 2022.

[77] 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 *CVPR*, 2022.

[78] Yue Wu, Yinpeng Chen, Lijuan Wang, Yuancheng Ye, Zicheng Liu, Yandong Guo, and Yun Fu. Large scale incremental learning. In *CVPR*, 2019.

[79] Xin Yang, Yuezun Li, and Siwei Lyu. Exposing deep fakes using inconsistent head poses. In *ICASSP*, 2019.

[80] Jaehong Yoon, Eunho Yang, Jeongtae Lee, and Sung Ju Hwang. Lifelong learning with dynamically expandable networks. *arXiv preprint arXiv:1708.01547*, 2017.

[81] Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015.

[82] Lu Yu, Bartlomiej Twardowski, Xialei Liu, Luis Herranz, Kai Wang, Yongmei Cheng, Shangling Jui, and Joost van de Weijer. Semantic drift compensation for class-incremental learning. In *CVPR*, 2020.

[83] Pei Yu, Yinpeng Chen, Ying Jin, and Zicheng Liu. Improving vision transformers for incremental learning. *arXiv preprint arXiv:2112.06103*, 2021.

[84] Guanxiong Zeng, Yang Chen, Bo Cui, and Shan Yu. Continual learning of context-dependent processing in neural networks. *Nature Machine Intelligence*, 1(8):364–372, 2019.

[85] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *ICML*, 2017.

[86] Peng Zhou, Xintong Han, Vlad I Morariu, and Larry S Davis. Two-stream neural networks for tampered face detection. In *CVPR Workshops*, 2017.

[87] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, 2017.

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