ECAI 2024
U. Endriss et al. (Eds.)

doi:10.3233/FAIA240516

Exemplar-Free Incremental Deepfake Detection

Wuti Xiong[1], Guoying Zhao[1] and Xiaobai Li[2,1,*]

[1]Center for Machine Vision and Signal Analysis, University of Oulu, Oulu, Finland
[2]State Key Laboratory of Blockchain and Data Security, Zhejiang University, Hangzhou, China

Abstract. Incremental Deepfake Detection (IDD) aims to continuously update models with new domain data, adapting to evolving forgery techniques. Existing works require extra buffers to store old exemplars for maintaining previously learned knowledge. However, it is infeasible when previous data is unavailable due to storage and privacy issues. This paper focuses on a more challenging but practical exemplar-free IDD problem that requests zero old exemplars when updating the model. To address this problem, we design a domain-adaptive module that uses independent adapters to learn domain-specific knowledge for each domain, avoiding using old exemplars. Besides, we introduce an uncertainty optimization strategy to optimize the adapters more efficiently. With excellent scalability, our method can be easily deployed to various models. To simulate the practical scenarios, we designed two new protocols based on diverse deepfake datasets. Extensive experimental results demonstrate that our method outperforms the state-of-the-art methods by a large margin. The code is available at *https://github.com/woody-panda/EF-IDD*.

1 Introduction

Deepfake detection aims to recognize forged or manipulated faces in digital images or videos, which is critical to ensure the authenticity and reliability of the information presented in the real world. With the ever-evolving forgery techniques, realistic forged media are popping up in various real-world scenarios. In the presence of a large domain gap [18], a pre-trained deepfake detection model struggles to recognize forged images or videos from the new domain accurately. In this case, it is necessary to continuously update the model with new domain data to perform well in emerging new domains. However, the upgraded model usually forgets previously learned knowledge, leading to a drastic performance drop in the previous domains. This is one of the main incremental learning challenges, called the catastrophic forgetting problem [20]. Therefore, it is crucial to develop anti-forgetting algorithms for incremental deepfake detection (IDD) to counter the challenge.

Recently, a few works have been proposed for IDD [28, 23, 30]. To alleviate catastrophic forgetting, these works use extra replay buffers to store old exemplars for either rehearsal [37] or distillation [33] when tuning the whole network to new domains. However, the previous data is not always available due to data ownership constraints and privacy issues. Furthermore, the use of replay buffers causes extra storage burdens. Therefore, this paper focuses on a more challenging but practical Exemplar-Free IDD (EF-IDD) problem. As shown in Figure 1, EF-IDD requests zero old exemplars when updating the model, aiming for robust continuous learning with better data security, privacy protection, and cheaper memory consumption.

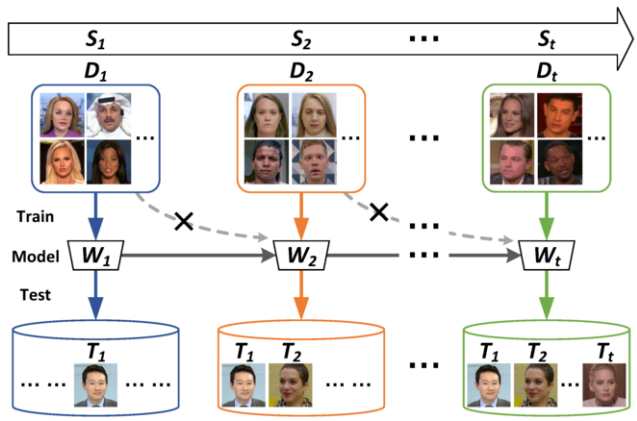

Figure 1: IDD consists of a sequence of sessions $S = \{S_1, S_2, \cdots, S_t\}$. The updated model W_t must continuously adapt to new domains in successive sessions without forgetting, $i.e.$, perform well on all domains $T = \{T_1, T_2, \cdots, T_t\}$. Current IDD methods require storing exemplars from previous data (dotted lines). We propose exemplar-free IDD, $i.e.$, at session S_t only the current domain D_t is available for training.

EF-IDD belongs to the exemplar-free domain-incremental learning (EF-DIL) task with a special focus on discriminating forged inputs from authentic ones. In general EF-DIL studies, one prevailing solution [45, 44, 41] is to learn a set of prompts over transformers [7]. A prompt pool is used to store the domain-specific knowledge for each domain; hence, a rehearsal buffer is no longer necessary. Also, a similarity-based selection strategy ($e.g.$, KNN [41] or cosine distance [44, 45]) is used to optimize the network and match the input with relevant prompts during the inference phase. Despite demonstrated promising results for general EF-DIL tasks, it cannot be directly used for EF-IDD for the two problems below. *1) Adaptability problem.* These EF-DIL approaches are tailored for transformer-based models only and cannot be deployed on other models, while the EF-IDD solution is supposed to be model-agnostic. A more generic EF-IDD solution that can work with various architectures for real-world deployment is needed. *2) Observation noise problem.* Deepfake detection tasks concern highly homogeneous data compared to other visual tasks, $i.e.$, high similarities between authentic and forged faces, from old and new domains. Thus, the naive similarity-based selection strategies might not be reliable and impact the learnable prompts negatively.

* Corresponding Author. Email: xiaobai.li@zju.edu.cn

In this paper, we focus on the EF-IDD task addressing the two problems. For the *adaptability problem*, we propose a highly adaptable Domain-adaptive Module (DaM) that uses independent adapters [32] to encode domain-specific knowledge. Compared with the prompt pool, DaM has better scalability so that it can be integrated into various visual models and better adapted to real-world scenarios. For the *observation noises problem*, we introduce an Uncertainty Optimization Strategy (UOS) that incorporates the uncertainty into the final loss for more effective optimization. Specifically, we convert the observed similarities between domain centers into a probabilistic distribution, with the variance characterizing the uncertainty of the observed similarities. An auxiliary loss is calculated based on Monte Carlo sampling for uncertainty optimization. The UOS can facilitate model learning by alleviating the influence of uncertainty of noisy inputs [17].

To evaluate the proposed method, we introduce two protocols to simulate the practical scenarios. 1) *Dataset-Incremental Deepfake Detection (D-IDD)*, $i.e.$, a new dataset with unknown forgery types and sources (achieved from a new domain) is available in each session. 2) *Type-Incremental Deepfake Detection (T-IDD)*, $i.e.$, a new forgery type (achieved with a new forgery technique) is available in each session. There could be either of the two scenarios in a practical situation. Our method is evaluated on both protocols to demonstrate its adaptability for the IDD task. The main contributions are as follows:

- We address the importance of EF-IDD and propose a highly adaptable EF-IDD framework, which can continuously adapt to both new data domains and new forged types while keeping all learned knowledge for robust deepfake detection.
- In the proposed framework, DaM is proposed using independent adapters to encode domain-specific knowledge to mitigate forgetting, which can work for various vision models. UOS is designed to alleviate the influence of observation noises by performing uncertainty optimization.
- Extensive experiments demonstrate that the proposed method significantly outperforms the SOTA methods on both D-IDD and T-IDD protocols. Furthermore, the model not only generalizes well to unseen domains detecting forgery faces, but also works well on a general scope of detecting forgery images (faces and others).

2 Related Work

2.1 Deepfake Detection

Owing to the success of Generative Adversarial Nets (GANs) [9], deepfake generation methods [2, 16] have achieved tremendous progress. To mitigate potential social risks, deep learning-based deepfake detection has received considerable research attention within the computer vision community. Early works [35, 24] adopt off-the-shelf image classification backbones ($e.g.$, Xception [3] and ResNet [12]) to perform binary classification on cropped facial images. However, these vanilla backbones can only capture limited spatial information on facial regions, which is not sufficient for a full understanding of forgery. To detect the subtle clues in forged faces, some recent works further mine specific forgery patterns, such as noise statistics [10], local textures [48], frequency information [31, 43], reconstruction difference [1], forgery inconsistency [49] and implicit identity information [15]. Despite significant progress, these methods do not consider updating models to deal with evolving forgery techniques.

2.2 Incremental Learning

Incremental learning [40] aims to continuously update models as new data emerges without forgetting previously learned knowledge. There are three common continual learning scenarios. Class-incremental learning focuses on learning new classes in a fixed-label space. Task-incremental learning continuously learns a sequence of disjoint tasks with dynamic and unrelated label spaces. Domain-incremental learning (DIL) aims to adapt a pre-trained model to a new domain with the same label space and new input distribution. Our work is related to DIL, specifically EF-DIL.

The main challenge of EF-DIL is to mitigate catastrophic forgetting in the absence of old exemplars. Several solutions have been proposed to address the problem. Regularization-based methods [25, 20, 47] assign importance or penalty to certain weights based on the sample's contribution to previous tasks, which limits the model's ability to represent complex relationships and causes the underfitting problem. Prompt-based methods [45, 44, 41, 38] learn a small set of embeddings (prompts) to store domain-specific knowledge. However, these methods specialize in the vision transformer architecture [7] and cannot be deployed to other models in real-world scenarios.

Recently, a few works [28, 23, 30] explored deepfake detection under incremental learning settings. However, they require storing old exemplars to mitigate forgetting, which might be infeasible in real-world scenarios due to privacy and storage issues.

2.3 Adaptive Tuning

Adaptive-tuning [32, 34] aims to insert trainable lightweight components into the pre-trained models to adapt to downstream tasks. Compared with expensive full-model fine-tuning, adapter-tuning only requires smaller training and storage costs to obtain similar results to the full-model fine-tuning method, which has been widely used in natural language processing, such as natural language understanding [36] and neural machine translation [14]. Inspired by these works, we develop a highly adaptable EF-IDD framework based on adaptive tuning.

2.4 Uncertainty in Deep Learning

In deep learning, uncertainty is the confidence degree of the model in its predictions [17]. A model usually has high uncertainty on noisy input or rarely seen input. Combining the uncertainty for network optimization can effectively improve the robustness of deep learning models. Therefore, uncertainty optimization is widely used in various vision tasks, such as object detection [11] and semantic segmentation [46]. In this paper, we model the data-dependent uncertainty of the calculated similarities between a query sample and the prototypes, and leverage it to better optimize the network.

3 Methodology

We propose a highly adaptable framework for exemplar-free incremental deepfake detection (EF-IDD). The problem definition is shown in Figure 1. We leverage a Domain-adaptive Module (DaM) that uses independent adapters to encode domain-specific knowledge, avoiding using old exemplars. To obtain reliable adapters, we design an Uncertainty Optimization Strategy (UOS) to incorporate the uncertainty into the final loss for effective optimization. Our method not only significantly reduces forgetting, but also achieves excellent scalability for deployment to diverse vision models.

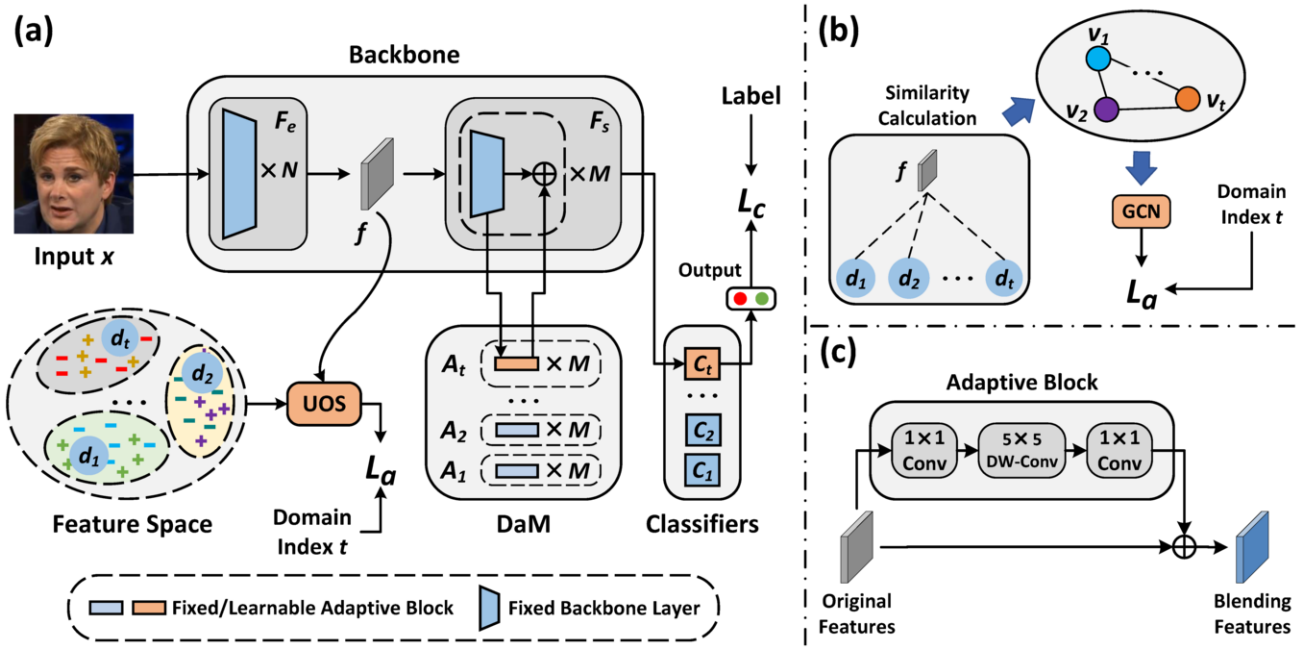

Figure 2: (a) Overview of the proposed framework. We divide the backbone into the first N layers as F_e and the last M layers as F_s. The training image x is processed by the F_e to generate f. Then, we perform the Uncertainty Optimization Strategy (UOS) on the feature f, domain centers $\{d_i\}_{i=1}^{t}$ and domain index t to obtain the auxiliary loss L_a. The t-th adapter A_t from the Domain-adaptive Module (DaM) is used to generate domain-specific prompts for blending intermediate features from F_s. The final loss is the sum of the classification loss L_a and L_c. (b) Illustration of UOS. We calculate the similarity between the feature f and t domain centers and convert it into a matrix $V = (v_1, v_2, \cdots, v_t) \in \mathbb{R}^{t\times L}$. Then, we employ a graph-based model to jointly infer the uncertainty to generate the auxiliary loss L_a. (c) Illustration of the adaptive block. The adaptive block consists of three sequential lightweight convolutional layers: 1×1 convolution, 5×5 depthwise convolution and 1×1 convolution.

3.1 Overview

The overview of the proposed method is illustrated in Figure 2 (a). Suppose the pre-trained backbone consists of total $M + N$ layers, we divide it into the first N layers as F_e and the last M layers as F_s, which can be varied to achieve the best performance and computational efficiency [44]. Note that vision backbones, $e.g.$, transformer-based and CNN-based, usually contain multiple blocks in one layer. To ensure the generality of our method, we follow their original concepts of layer partitioning.

Given a training image x, we first feed it to the F_e to generate feature $f \in \mathbb{R}^{H\times W\times C}$ as

$$f = F_e(x). \tag{1}$$

Then, we generate an auxiliary loss for better optimization [45]. At the t-th session, we assign x a domain index t and generate a domain center $d_t \in \mathbb{R}^C$ from D_t. Specifically, we use F_e to obtain feature embeddings for each sample. Then, we average the feature embeddings of all training examples and perform a global average pooling operation on that average feature embedding. The process can be formulated as

$$d_t = AvgPool(\frac{1}{n}\sum_{i=1}^{n} F_e(x_i)), \quad D_t = \{x_i\}_{i=1}^{n}, \tag{2}$$

where $AvgPool$ denotes the global averaging pooling operation. The auxiliary loss L_a is calculated by performing UOS (will be elaborated in Section 3.2) on the feature f, domain index t and domain centers $\{d_i\}_{i=1}^{t}$. Next, the feature f is fed into F_s for multi-stage blending with prompts from the t-th adapter A_t of DaM (will be elaborated in Section 3.3), which can minimize the knowledge gap between the pre-trained backbone and the current domain and avoid catastrophic forgetting. We denote the final blending output as $\tilde{f}_s$. The final result p is generated by the t-th classifier C_t which is a normal fully connected (FC) layer made by $[W_t, b_t]$ calculated as

$$p = softmax\left(W_t\tilde{f}_s + b_t\right). \tag{3}$$

Given the binary label y (0 or 1) indicating the real or fake of the input face image x, we employ the cross-entropy loss as the classification function:

$$L_c = -[y \cdot log(p) + (1-y) \cdot log(1-p)]. \tag{4}$$

The total loss is the sum of the classification loss L_c and the auxiliary loss L_a as:

$$L_{total} = L_c + L_a. \tag{5}$$

Note that we only tune the current domain-related adapter A_t, UOS and the classifier C_t, as the orange parts in Figure 2, while the rest components, $i.e.$, the backbone, the unrelated adapters and classifiers, are frozen, as the blue parts in Figure 2. In this case, each domain-specific component is learned independently from other domains. In the following sections, we describe the two key components UOS, DaM and inference phase in detail.

3.2 Uncertainty Optimization Strategy (UOS)

During the training phase, generating a domain classification loss based on the assigned domain index can achieve better optimiza-

tion [45, 44, 41]. However, naive similarity-based strategies suffer from the observation noise problem, where highly homogeneous data could be assigned to an inappropriate adapter, leading to negative optimization. To address this problem, we use uncertainty estimation to jointly optimize the network to reduce the influence of observation noise.

Similarity Calculation. As shown in Figure 2 (b), we first calculate the group-wise similarities between the feature f and domain centers $\{d_i\}_{i=1}^{t}$. For the feature f, we perform global averaging pooling to get a C-dimensional feature vector $f \in \mathbb{R}^C$. For the i-th ($i = 1, 2, \cdots, t$) domain center d_i, we both split it and f into L groups along their channel dimensions to have $\left\{d_i^l\right\}_{l=1}^{L}$ and $\left\{f^l\right\}_{l=1}^{L}$, where $L < C$. The similarity of the l-th group is calculated as

$$r_i^l = \frac{f^l {d_i^l}^T}{||f^l|| \cdot ||d_i^l||}. \tag{6}$$

Next, the group similarities are stacked as a relation feature vector $v_i = [r_i^1, r_i^2, \cdots, r_i^L] \in \mathbb{R}^L$. Note that the observed similarities across multiple groups provide valuable hints of the uncertainty since they reflect similarities from different perspectives. Formally, the observed similarities between f and $\{d_i\}_{i=1}^{t}$ can be are represented by a matrix $V = (v_1, v_2, \cdots, v_t) \in \mathbb{R}^{t \times L}$.

Uncertainty Estimation. Since the domain classification probability of a test sample is determined based on the similarity of this sample with t domain centers, this is a joint determination process. Therefore, we adopt a Graph Convolutional Network [19] (GCN) to jointly estimate the uncertainties for the t similarity pairs, which facilitates the information passing among them for the joint optimization.

Specifically, $V = (v_1, v_2 \cdots, v_t)$ is viewed as a graph containing t nodes. We first generate an adjacency matrix $E \in \mathbb{R}^{t \times t}$ denoted by modeling node affinity in the embedded space:

$$E_{ij} = \varphi_1(v_i)\varphi_2(v_j)^T, \tag{7}$$

where E_{ij} denotes the edge from the i-th node to the j-th node. φ_1 and φ_2 denote two linear projections implemented by the FC layer. To generate numerically stable messages through the the modeled graph, we use the $softmax$ function to normalize each row of E so that all edges connected to the target node have a value of 1. Next, we update the nodes through GCN as:

$$V = V + YW_v, \quad Y = EYW_y, \tag{8}$$

$W_v \in \mathbb{R}^{L \times L}$ and $W_y \in \mathbb{R}^{L \times L}$ are two learnable transformation matrices. W_y is implemented by a 1×1 convolutional layer. W_v is implemented by two stacked blocks. Each block consists of a 1×1 convolution layer, followed by a Batch Normalization layer and an LeakyReLU activation layer. We infer the similarity uncertainty vector $u = [\sigma_1, \sigma_2, \cdots, \sigma_t] \in \mathbb{R}^t$ by

$$u = \alpha\left(BN\left(VW_1\right)\right)W_2, \tag{9}$$

where $W_1 \in \mathbb{R}^{L \times L}$ and $W_2 \in \mathbb{R}^{L \times 1}$ are transformation matrices both implemented by a 1×1 convolutional layer. "BN" refers to the Batch Normalization layer and $\alpha(\cdot)$ refers to the LeakyReLU activation function. The i-th dimension of u, σ_i, is the similarity uncertainty for the feature f and the i-th domain center.

Auxiliary Loss Generation. The analytical solution to integrating the distributions to optimize the losses is difficult. Inspired by previous work [17], we use Monte Carlo integration to approximate the optimization objective. Specifically, Monte Carlo sampling is performed on the t similarity distributions. For the feature f and the i-th ($i = 1, 2, \cdots, t$) domain center d_i, we repeat K random sampling over the similarity distributions σ_i to obtain statistical results. At the k-th sample, the differentiable representation $s_{i,k}$ can be obtained by

$$s_{i,k} = \mu_i + \sigma_i \epsilon_k, \quad \epsilon_k \in \mathcal{N}(0, 1), \tag{10}$$

where $\mathcal{N}$ denotes Gaussian distribution. μ_i denotes the mean of the similarity that can be obtained by the inner product operation, $i.e.$, $\mu_j = \langle f, d_j \rangle$. For the given sample x with domain index t, we obtain its corresponding domain classification loss as:

$$L_a = -log(\frac{1}{K}\sum_{k=1}^{K}(e^{s_{t,k}} / \sum_{i=1}^{t} e^{s_{i,k}})). \tag{11}$$

Then we use L_a as an auxiliary loss for better optimization.

3.3 Domain-adaptive Module (DaM)

To enable the pre-trained backbone to acquire knowledge from new domains while preserving previous knowledge efficiently, we introduce Domain-adaptive Modules (DaM) that use highly adaptable independent adapters to learn domain-specific prompts for each domain. In this case, the proposed framework can avoid using old exemplars to maintain previously learned knowledge while being deployable to diverse vision models.

As shown in Figure 2 (a), domain-specific prompts are generated by an independent adapter that contains M adaptive blocks to enrich the feature space from diverse levels. Specifically, each adaptive block corresponds to a sub-layer in F_s. At the t-th session, suppose that the output of the i-th layer of F_s is f_s^i, the blending representation is:

$$\tilde{f}_s^i = f_s^i + \beta A_t^i(f_s^i), \tag{12}$$

where β denotes a learnable parameter to balance the two terms. A_t^i denotes the i-th adaptive block in the t-th adapter. As shown in Figure 2 (c), we implement the adaptive block with three sequential lightweight convolutional layers: 1×1 convolution, 5×5 depthwise convolution [3] and 1×1 convolution. This design allows the pre-trained backbone to maintain a compact model size and significantly reduce computational cost under the EF-IDD setting since the deep convolutions can perform the entire calculation in a structured, sparse manner.

3.4 Inference Phase

For a test instance, we first obtain its feature from the pre-trained F_e. We then compute the similarities between the feature and all domain centers. The domain index is obtained by performing $softmax$ on the similarities. Based on the domain index, we perform the corresponding adapter and classifier on the features to generate the final prediction.

4 Experiments

4.1 Experimental Setup

Datasets and protocols. The proposed EF-IDD framework is evaluated under both D-IDD and T-IDD protocols. For the D-IDD protocol, eight large benchmark databases are used as listed in Table 1. For the T-IDD protocol, we use eight subsets from ForgeryNet [13]. Each

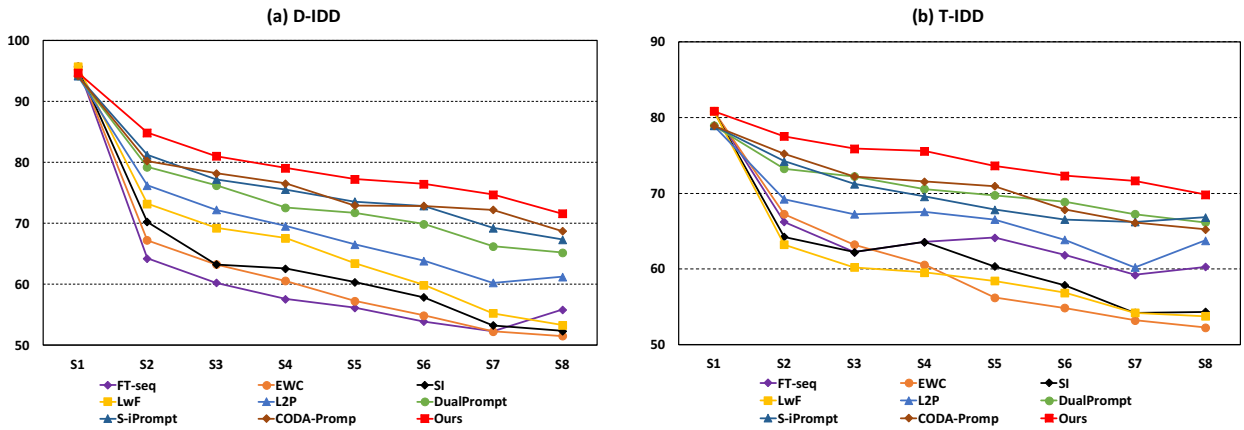

Figure 3: Performance comparison for each session on D-IDD and T-IDD protocols. We report the Average Accuracy (%).

Table 1: Datasets used for the D-IDD and T-IDD protocols.

Season ID	D-IDD	T-IDD
S_1	FF++ [35]	DeepFakes
S_2	Celeb-DF [24]	StyleGAN2
S_3	DFDC-P [6]	FS-GAN
S_4	DFFD [4]	BlendFace
S_5	FFIW [50]	MaskGAN
S_6	OpenForensics [22]	SC-FEGAN
S_7	ForgeryNIR [42]	DF-StarGAN
S_8	ForgeryNet [13]	DiscoFaceGAN

subset contains one type of forged sample including both images and videos. More dataset details can be found in the supplementary[1].

Baselines. We compare the proposed methods against multiple SOTA EF-DIL methods, including four non-prompting methods **FT-seq**, **EWC** [20], SI [47], **LwF** [25], and four recent prompting methods **L2P** [45], **DualPrompt** [44], **S-iPrompt** [41] and **CODA-Prompt** [38]. Note that FT-seq is the naive sequential fine-tuning approach with pre-trained model weights. For EWC, SI and LwF, we use the public implementations from the Mammoth toolbox[2] with SwinT-B [26] backbone and the same hyper-parameters as in their original paper. For L2P[3], DualPrompt[3], S-iPrompt[4] and CODA-Prompt[5], we use the official implementations with tuned parameters for better performances.

Evaluation metrics. Following previous works [23, 38], we use the average accuracy (AA) and average forgetting degree [27] (AF) as the evaluation metrics. The former is the average final accuracy over all observed domains. The latter is the average performance drop across all domains.

Implementation details. The proposed method is implemented in PyTorch with NVIDIA A100 GPUs. We use the same image backbone SwinT-B [26] across all domains. RetinaFace [5] is employed to extract faces for all datasets. For video-level datasets, 50 frames were randomly selected from each video for testing and training. All the training images are resized to 384×384. For two protocols, we adopt an Stochastic Gradient Descent (SGD) optimizer with a momentum of 0.9, an initial learning rate of 0.01, and a batch size of 64. The number of epochs is set to 10 which is enough to fit all training sets. We adopt strong data augmentations only for ForgeryNet [13] dataset, while weak data augmentation are used for the rest datasets. For the parameters L and K, we set them to 32 and 50 respectively.

[1] https://github.com/woody-panda/EF-IDD/blob/main/Supplementary.pdf
[2] https://github.com/aimagelab/mammoth
[3] https://github.com/google-research/l2p,
[4] https://github.com/iamwangyabin/S-Prompts
[5] https://github.com/GT-RIPL/CODA-Prompt

4.2 Comparison with SOTA Methods

We compare our results with eight SOTA EF-DIL baselines. The results in Table 2 can be summarized in two points. 1). Non-prompting methods (the top part of the table) significantly underperform prompting methods (the bottom part of the table), even underperform the naive fine-tuning FT-seq. This shows that the prompting method can effectively learn new knowledge without forgetting the previous knowledge. 2). Our method significantly outperforms all SOTA methods, and the advantage can be consistently observed in the two metrics of AA and AF under both protocols. The largest performance improvement is achieved on T-IDD AA, i.e., 2.99% higher than the second-best. The results strongly support our claims that prompt-based approaches may not be the best option for highly homogeneous face data in the IDD task, while our UOS and DaM work better to improve the overall performance and mitigate forgetting even after eight sessions.

We then further compare the performance in each season with the existing EF-DIL method. As shown in Figure 3, we can observe that our method achieves state-of-the-art performance on each season for both D-IDD and T-IDD. The outstanding performance of the proposed method over all competing methods indicates that our proposed UOS and DaM successfully accumulate knowledge from experiences, thus it can overall improve the learning performance while mitigating catastrophic forgetting.

4.3 Ablation Studies

We carry out four ablation experiments under the D-IDD protocol using all eight datasets in Table 1.

Different backbones. One major advantage of the proposed method is that it can work with various models, *e.g.*, CNN-based and transformer-based models. Here we compare four different backbones, transformer-based SwinT-B [26] and ViT-B/16 [8], and two CNN-based Xception [3] and ResNet-50 [12]. Table 3 presents the

Table 2: Comparison to baseline methods under D-IDD and T-IDD protocols. Upper: non-prompt methods. Bottom: prompt methods.

Method	D-IDD		T-IDD	
	AA (↑)	AF (↓)	AA (↑)	AF (↓)
FT-seq	55.81	30.21	60.28	25.56
EWC [20]	51.48	39.43	52.29	20.34
SI [47]	52.34	26.26	54.35	16.52
LwF [25]	53.29	19.32	53.77	14.34
L2P [45]	61.21	16.48	63.76	12.29
DualPrompt [44]	65.21	13.75	66.16	11.84
S-iPrompt [41]	67.35	9.25	66.84	10.37
CODA-Prompt [38]	68.71	8.64	65.23	11.25
Ours	**71.59**	**7.12**	**69.83**	**8.54**

Table 3: Ablations of four different backbones.

Backbones	ResNet-50	ViT-B/16	Xception	SwinT-B
AA (↑)	64.31	68.94	70.36	**71.59**
AF (↓)	11.52	8.57	8.16	**7.12**

results. All four backbones achieve good performance compared to baseline results in Table 2. This confirms the robustness of our proposed framework which works well with various models in the exemplar-free setting. Moreover, SwinT-B achieves the best performance among all four backbones, which is adopted for the following experiments.

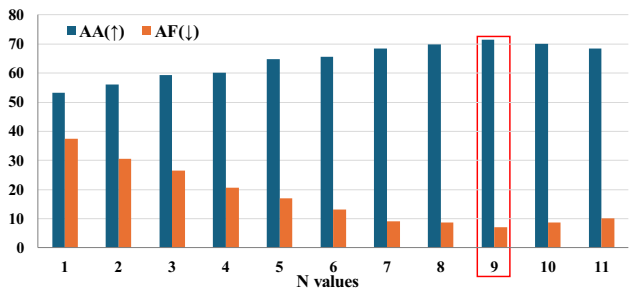

Figure 4: Performance with different N values. ($N + M = 12$)

Impact of N values. We divide the backbone network into the first N layers and the last M layers, as F_e and F_s respectively. Note that we use SwinT-B as the backbone, which consists of 12 ($N + M = 12$) sub-layers. Here we evaluate the impact of different N and M values on performance. In Figure 4 it shows that our model achieves the best performance when $N = 9$ and $M = 3$. The main reason is that a deeper F_e brings more compact features to each domain of UOS, while a deeper F_s provides more efficient features for separately learning domain-specific knowledge and sequentially learning domain-invariant knowledge. Therefore, we set N and M as 9 and 3, respectively in all experiments.

Effectiveness of UOS and DaM. We evaluate the two key components of the framework, *i.e.*, UOS and DaM, to demonstrate their effectiveness. The results are listed in Table 4. "Baseline" indicates the pre-trained SwinT-B backbone with independent classifiers for each domain. Results show that employing only UOS or DaM both improves the performance, and the improvement brought by UOS is larger than DaM. The best performance is achieved when both UOS and DaM are employed at the same time. These results demonstrate

Table 4: Effectiveness of UOS and DaM.

Method	AA (↑)	AF (↓)
Baseline	61.37	16.54
Baseline + DaM	67.28	10.34
Baseline + UOS	68.64	8.91
Baseline + UOS + DaM	**71.59**	**7.12**

the effectiveness of UOS and DaM. The advantage of UOS indicates that effective optimization strategies are very important in the EF-IDD task.

Different UOS methods. The proposed UOS can be equipped with different uncertainty estimation methods, and here we evaluate three optional methods, *i.e.*, Fully-Connected layer (FC), 1×1 Convolutional Neural Network (Conv) and Graph Convolutional Network (GCN). Results in Table 5 show that the GCN-based method works the best among all, which is employed in our experiments. This demonstrates that information passing across different similarity pairs is beneficial for joint optimization.

Table 5: Ablations of different UOS methods. "-" denotes that the model without uncertainty optimization.

U-Estimation	-	FC	Conv	GCN
AA (↑)	67.28	68.16	70.23	**71.59**
AF (↓)	10.34	9.74	8.22	**7.12**

4.4 *Generalization.*

Generalization to unseen domains is a major challenge for most tasks and especially for Deepfake detection. Newly forged samples are not always available for training (not even noticed as new). It is important that an IDD model can generalize well to unseen data. Here we test our previously trained models (under both D-IDD and T-IDD protocols) on four extra Deepfake datasets and compare them with baseline methods. For D-IDD, we use DFD [29] and Kodf [21]. For T-IDD, we use FaceShifter and StarGAN2 from ForgeryNet [13] dataset. Results in Table 6 show that our approach achieves the best performance among all methods. Compared with the second-best method S-iPrompt, the improvement is stable across all four datasets under both protocols, ranging from 1% to 2%. This shows that the proposed method generalizes better to unseen domain data than the current EF-DIL methods.

Table 6: Performance on unseen domains as Accuracy (%).

Method	D-IDD		T-IDD	
	DFD	Kodf	FaceShifter	StarGAN2
FT-seq	53.11	52.43	52.45	54.37
EWC [20]	55.14	56.32	54.21	53.86
SI [47]	55.94	51.25	53.72	52.36
LwF [25]	57.25	53.29	55.84	54.90
L2P [45]	66.23	62.12	59.67	57.82
DualPrompt [44]	69.97	64.48	63.61	64.59
CODA-Prompt [41]	69.16	63.27	63.19	65.83
S-iPrompt [38]	70.20	65.42	64.17	66.87
Ours	**71.28**	**66.74**	**65.53**	**68.76**

From another perspective of generalization, our EF-IDD framework is not constrained to face inputs but also works for detecting forged images on a general level. We evaluate our approach on one latest benchmark CDDB-long [23], which contains 842K images of diverse contents (scenes, animals, cars, *etc.*), and synthesized fake

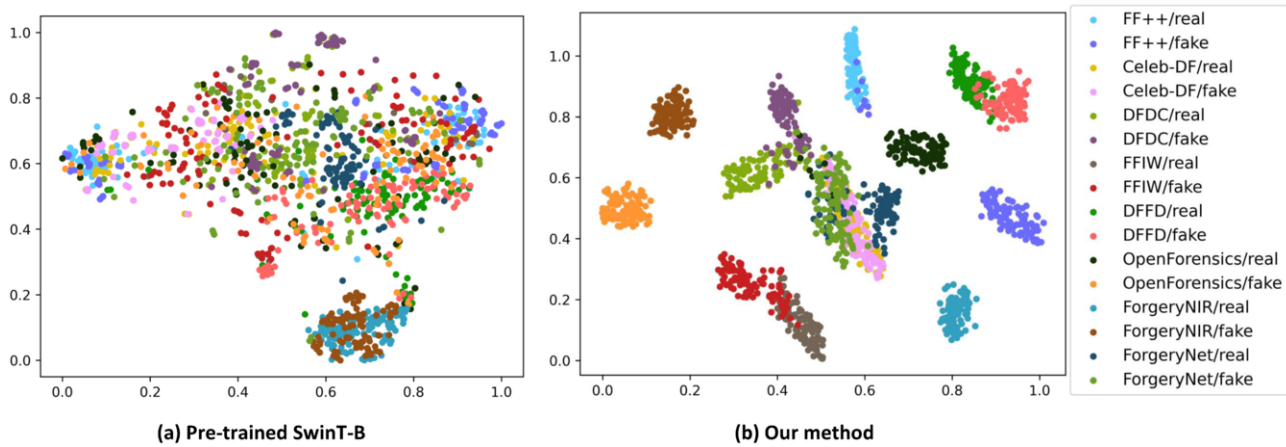

Figure 5: t-SNE visualization on the resulting feature spaces of pre-trained SwinT-B and our method on D-IDD.

samples from 13 deepfake sources. The model was continuously trained in 13 sessions and we followed the same protocol as the original paper. Results in Table 7 show that non-prompting methods suffer catastrophic accuracy drops of over 30%, due to the highly diverse data contents and forgery sources, while our method achieves the best performance, with the forgetting rate significantly reduced by 1.89% compared to the second best method S-iPrompt. This result demonstrates that our method is also effective for detecting other categories of forged images.

Table 7: Performance on general forgery image detection. Results on the CDDB-Long benchmark.

Method	AA (↑)	AF (↓)
FT-seq	54.28	42.39
EWC [20]	50.36	41.52
SI [47]	51.48	38.46
LwF [25]	58.86	17.38
L2P [45]	60.21	14.58
DualPrompt [44]	64.38	10.57
CODA-Prompt [41]	65.94	9.32
S-iPrompt [38]	66.39	8.47
Ours	**67.84**	**6.58**

4.5 Visualization

We use the popular scheme t-SNE [39] for the visualization under the D-IDD protocol. As shown in Figure 5, compared with pre-trained backbones, our proposed method has significant advantages in inter-domain separation and inner-domain clustering. This confirms the validity of UOS and DaM, which can effectively map instance features into various domains to facilitate the IDD task.

5 Conclusion

To address the evolving deepfakes in the real world, this study focuses on the Exemplar-Free Incremental Deepfake Detection (EF-IDD) problem. Compared with existing works, EF-IDD requires the model to mitigate the catastrophic forgetting problem without accessing any data from previous sessions, which is challenging but fits practical needs. The Domain-adaptive module (DaM) and the uncertainty optimization strategy (UOS) are proposed to counter the challenge. DaM can learn domain-specific prompts independently, avoiding using old examples. UOS employs uncertainty optimization to alleviate the influence of observation noises. Extensive experiments were carried out on multiple datasets under two incremental protocols, and the results demonstrate that our method significantly outperforms existing methods. In the future, we will consider other challenging scenarios of the EF-IDD task, $e.g.$, in the current session only limited samples are available for training, or data with no labels.

Acknowledgement

This work was supported by the Research Council of Finland (former Academy of Finland) ICT 2023 project TrustFace (grant 345948), Academy Professor project EmotionAI (grants 336116, 345122, 359854), the University of Oulu & Research Council of Finland Profi 7 (grant 352788), and by Infotech Oulu. As well, the authors wish to acknowledge CSC-IT Center for Science, Finland, for computational resources.

References

[1] J. Cao, C. Ma, T. Yao, S. Chen, S. Ding, and X. Yang. End-to-end reconstruction-classification learning for face forgery detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4113–4122, 2022.

[2] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8789–8797, 2018.

[3] F. Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1251–1258, 2017.

[4] H. Dang, F. Liu, J. Stehouwer, X. Liu, and A. K. Jain. On the detection of digital face manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5781–5790, 2020.

[5] J. Deng, J. Guo, E. Ververas, I. Kotsia, and S. Zafeiriou. Retinaface: Single-shot multi-level face localisation in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5203–5212, 2020.

[6] B. Dolhansky, J. Bitton, B. Pflaum, J. Lu, R. Howes, M. Wang, and C. C. Ferrer. The deepfake detection challenge (dfdc) preview dataset. *ArXiv*, 2019.

[7] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2020.

[8] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

[9] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014.

[10] Q. Gu, S. Chen, T. Yao, Y. Chen, S. Ding, and R. Yi. Exploiting fine-grained face forgery clues via progressive enhancement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 735–743, 2022.

[11] A. Harakeh and S. L. Waslander. Estimating and evaluating regression predictive uncertainty in deep object detectors. In *International Conference on Learning Representations*, 2020.

[12] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[13] Y. He, B. Gan, S. Chen, Y. Zhou, G. Yin, L. Song, L. Sheng, J. Shao, and Z. Liu. Forgerynet: A versatile benchmark for comprehensive forgery analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4360–4369, 2021.

[14] N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. De Laroussilhe, A. Gesmundo, M. Attariyan, and S. Gelly. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, pages 2790–2799. PMLR, 2019.

[15] B. Huang, Z. Wang, J. Yang, J. Ai, Q. Zou, Q. Wang, and D. Ye. Implicit identity driven deepfake face swapping detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4490–4499, 2023.

[16] T. Karras, S. Laine, and T. Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2019.

[17] A. Kendall and Y. Gal. What uncertainties do we need in bayesian deep learning for computer vision? *Advances in Neural Information Processing Systems*, 30, 2017.

[18] A. Khosla, T. Zhou, T. Malisiewicz, A. A. Efros, and A. Torralba. Undoing the damage of dataset bias. In *European Conference on Computer Vision*, pages 158–171. Springer, 2012.

[19] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2016.

[20] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017.

[21] P. Kwon, J. You, G. Nam, S. Park, and G. Chae. Kodf: A large-scale korean deepfake detection dataset. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10744–10753, 2021.

[22] T.-N. Le, H. H. Nguyen, J. Yamagishi, and I. Echizen. Openforensics: Large-scale challenging dataset for multi-face forgery detection and segmentation in-the-wild. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10117–10127, 2021.

[23] C. Li, Z. Huang, D. P. Paudel, Y. Wang, M. Shahbazi, X. Hong, and L. Van Gool. A continual deepfake detection benchmark: Dataset, methods, and essentials. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1339–1349, 2023.

[24] Y. Li, X. Yang, P. Sun, H. Qi, and S. Lyu. Celeb-df: A large-scale challenging dataset for deepfake forensics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3207–3216, 2020.

[25] Z. Li and D. Hoiem. Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(12):2935–2947, 2017.

[26] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021.

[27] D. Lopez-Paz and M. Ranzato. Gradient episodic memory for continual learning. *Advances in Neural Information Processing Systems*, 30, 2017.

[28] F. Marra, C. Saltori, G. Boato, and L. Verdoliva. Incremental learning for the detection and classification of gan-generated images. In *IEEE International Workshop on Information Forensics and Security*, pages 1–6. IEEE, 2019.

[29] D. Nick and G. Andrew. Contributing data to deepfake detection research, 2019.

[30] K. Pan, Y. Yin, Y. Wei, F. Lin, Z. Ba, Z. Liu, Z. Wang, L. Cavallaro, and K. Ren. Dfil: Deepfake incremental learning by exploiting domain-invariant forgery clues. In *Proceedings of the ACM International Conference on Multimedia*, pages 8035–8046, 2023.

[31] Y. Qian, G. Yin, L. Sheng, Z. Chen, and J. Shao. Thinking in frequency: Face forgery detection by mining frequency-aware clues. In *European Conference on Computer Vision*, pages 86–103. Springer, 2020.

[32] S.-A. Rebuffi, H. Bilen, and A. Vedaldi. Learning multiple visual domains with residual adapters. *Advances in Neural Information Processing Systems*, 30, 2017.

[33] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2001–2010, 2017.

[34] S.-A. Rebuffi, H. Bilen, and A. Vedaldi. Efficient parametrization of multi-domain deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8119–8127, 2018.

[35] A. Rossler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner. Faceforensics++: Learning to detect manipulated facial images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1–11, 2019.

[36] A. Rücklé, G. Geigle, M. Glockner, T. Beck, J. Pfeiffer, N. Reimers, and I. Gurevych. Adapterdrop: On the efficiency of adapters in transformers. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 7930–7946. Association for Computational Linguistics, 2021.

[37] H. Shin, J. K. Lee, J. Kim, and J. Kim. Continual learning with deep generative replay. *Advances in Neural Information Processing Systems*, 30, 2017.

[38] J. S. Smith, L. Karlinsky, V. Gutta, P. Cascante-Bonilla, D. Kim, A. Arbelle, R. Panda, R. Feris, and Z. Kira. Coda-prompt: Continual decomposed attention-based prompting for rehearsal-free continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11909–11919, 2023.

[39] L. Van der Maaten and G. Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(11), 2008.

[40] L. Wang, J. Xie, X. Zhang, M. Huang, H. Su, and J. Zhu. Hierarchical decomposition of prompt-based continual learning: Rethinking obscured sub-optimality. *Advances in Neural Information Processing Systems*, 36, 2023.

[41] Y. Wang, Z. Huang, and X. Hong. S-prompts learning with pre-trained transformers: An occam's razor for domain incremental learning. *Advances in Neural Information Processing Systems*, 35:5682–5695, 2022.

[42] Y. Wang, C. Peng, D. Liu, N. Wang, and X. Gao. Forgerynir: deep face forgery and detection in near-infrared scenario. *IEEE Transactions on Information Forensics and Security*, 17:500–515, 2022.

[43] Y. Wang, K. Yu, C. Chen, X. Hu, and S. Peng. Dynamic graph learning with content-guided spatial-frequency relation reasoning for deepfake detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7278–7287, 2023.

[44] Z. Wang, Z. Zhang, S. Ebrahimi, R. Sun, H. Zhang, C.-Y. Lee, X. Ren, G. Su, V. Perot, J. Dy, et al. Dualprompt: Complementary prompting for rehearsal-free continual learning. In *European Conference on Computer Vision*, pages 631–648. Springer, 2022.

[45] Z. Wang, Z. Zhang, C.-Y. Lee, H. Zhang, R. Sun, X. Ren, G. Su, V. Perot, J. Dy, and T. Pfister. Learning to prompt for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 139–149, 2022.

[46] L. Yu, S. Wang, X. Li, C.-W. Fu, and P.-A. Heng. Uncertainty-aware self-ensembling model for semi-supervised 3d left atrium segmentation. In *Medical Image Computing and Computer Assisted Intervention*, pages 605–613. Springer, 2019.

[47] F. Zenke, B. Poole, and S. Ganguli. Continual learning through synaptic intelligence. In *International Conference on Machine Learning*, pages 3987–3995. PMLR, 2017.

[48] H. Zhao, W. Zhou, D. Chen, T. Wei, W. Zhang, and N. Yu. Multi-attentional deepfake detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2185–2194, 2021.

[49] T. Zhao, X. Xu, M. Xu, H. Ding, Y. Xiong, and W. Xia. Learning self-consistency for deepfake detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15023–15033, 2021.

[50] T. Zhou, W. Wang, Z. Liang, and J. Shen. Face forensics in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5778–5788, 2021.