DevFD: Developmental Face Forgery Detection by Learning Shared and Orthogonal LoRA Subspaces

Tianshuo Zhang[1,2] **Li Gao**[3] **Siran Peng** [1,2] **Xiangyu Zhu**[1,2*] **Zhen Lei**[1,2,4,5*]
[1] School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China
[2] MAIS, Institute of Automation, Chinese Academy of Sciences, Beijing, China
[3] China Mobile Financial Technology Co., Ltd., Beijing, China
[4] CAIR, HKSIS, Chinese Academy of Sciences, Hong Kong, China
[5] School of Computer Science and Engineering, the Faculty of Innovation Engineering,
M.U.S.T, Macau, China
tianshuo.zhang@nlpr.ia.ac.cn, gaolids@chinamobile.com
pengsiran2023@ia.ac.cn, xiangyu.zhu@ia.ac.cn, zhen.lei@ia.ac.cn

Abstract

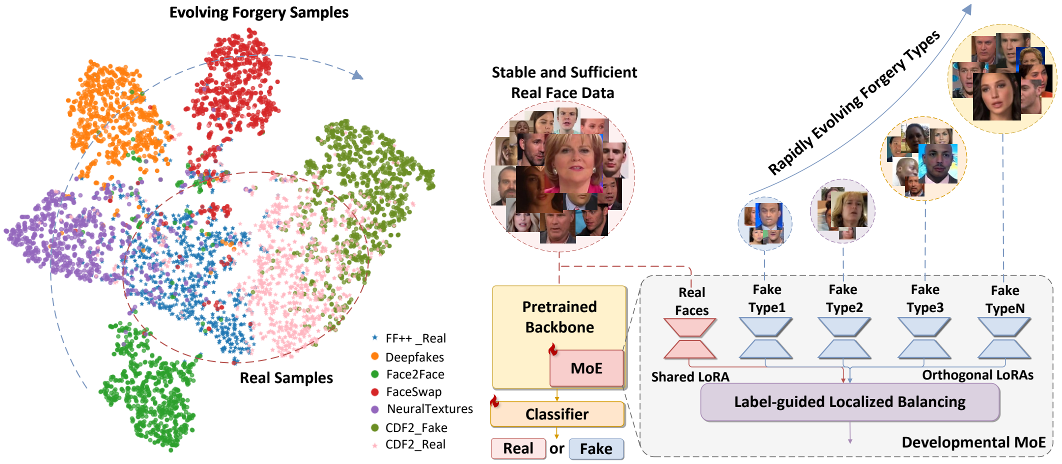
The rise of realistic digital face generation and manipulation poses significant social risks. The primary challenge lies in the rapid and diverse evolution of generation techniques, which often outstrip the detection capabilities of existing models. To defend against the ever-evolving new types of forgery, we need to enable our model to quickly adapt to new domains with limited computation and data while avoiding forgetting previously learned forgery types. In this work, we posit that genuine facial samples are abundant and relatively stable in acquisition methods, while forgery faces continuously evolve with the iteration of manipulation techniques. Given the practical infeasibility of exhaustively collecting all forgery variants, we frame face forgery detection as a continual learning problem and allow the model to develop as new forgery types emerge. Specifically, we employ a Developmental Mixture of Experts (MoE) architecture that uses LoRA models as its individual experts. These experts are organized into two groups: a Real-LoRA to learn and refine knowledge of real faces, and multiple Fake-LoRAs to capture incremental information from different forgery types. To prevent catastrophic forgetting, we ensure that the learning direction of Fake-LoRAs is orthogonal to the established subspace. Moreover, we integrate orthogonal gradients into the orthogonal loss of Fake-LoRAs, preventing gradient interference throughout the training process of each task. Experimental results under both the datasets and manipulation types incremental protocols demonstrate the effectiveness of our method.

1 Introduction

The swift evolution of generative models, including Generative Adversarial Networks (GANs) [2, 38, 62] and Diffusion models [17, 18, 61], along with the rise of large models [4, 25, 50, 63, 71], propels the rapid advancement of face forgery technology, which introduces significant security risks and contributes to a crisis of public trust. These manipulation methods leave unique traces of forgery. Existing detectors perform well with known forgery types but struggle to identify novel ones. When new manipulation types arise, a trivial method is to add new forgery data to the primary dataset and retrain a binary model. However, the rapid evolution of forgery techniques necessitates a new learning strategy that can quickly adapt to new forgeries using limited computation and data.

[*] Corresponding Authors.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).



(a) t-SNE visualization on FF++ and CDF2.

(b) Developmental MoE Forgery Detection Framework

Figure 1: (a) The t-SNE visualization of features extracted from the baseline on FF++ [48] and CDF2 [27] shows that real faces exhibit a close distribution, while fake faces form five distinct clusters. This observation inspires us to (b) propose a developmental mixture of experts to model the continuous emergence of unknown fake faces using a set of orthogonal LoRA subspaces, while concurrently employing a dedicated LoRA to preserve the commonalities of authentic faces. A label-guided localized balancing strategy employs the LoRA sequence to separately model the common real faces and the incremental fake types information.

In contrast to the rapidly evolving fake faces, real faces possess distinct characteristics. Genuine facial data is relatively abundant, stable, and acquired through a single modality, such as camera imaging. In the context of forgery detection, it typically does not introduce bias. These properties, which remain invariant as forgery techniques evolve, endow real faces with commonalities that may easily be overlooked. Fig. 1(a) presents a simple t-SNE [57] visualization experiment, using the baseline [8] model to extract features from both the FF++ [48] and CDF2 [27] datasets. The results reveal two key observations: (1) Real face samples from both datasets exhibit close distributions, while (2) forged samples form five distinct clusters. These observations inspire us to enable the model to adapt to emerging manipulation methods while learning the commonalities of real faces.

In this paper, we approach face forgery detection as a continual learning problem and make the model "developmental," enabling it to adapt to new forgery types continuously. As shown in Fig. 1(b), we propose a developmental Mixture of Experts (MoE) architecture, DevFD, which employs Low-Rank Adaptation (LoRA) models as individual experts to fine-tune pre-trained models within different subspaces. As illustrated in Fig. 2, the model expands a new LoRA branch when adapting to an unknown forgery type. However, adapting to new forgery types can disrupt the knowledge in established subspaces, leading to catastrophic forgetting. To address this, we use an orthogonal loss to constrain the learning direction of new LoRAs to be orthogonal to all previously established LoRAs, thereby preserving the learned knowledge. This part of the LoRAs ultimately forms an orthogonal sequence termed Fake-LoRAs. For real faces that are stable and exhibit common properties, modeling them with the same sequence overlooks their overall distribution. Therefore, we introduce a separate shared expert, termed Real-LoRA, to fine-tune and refine knowledge about real faces. To enable the Real-LoRA and Fake-LoRAs to align with their assigned roles and avoid disrupting the orthogonality of Fake-LoRAs, we design a label-guided localized balancing strategy that softly constrains the experts through a weighted response matrix. For unknown forgery images, the router aggregates the outputs of all LoRAs to make a joint decision, thereby maintaining collaboration among the experts.

Additionally, we theoretically analyze that the subspace orthogonal loss still causes forgetting due to the early stages of optimization not satisfying the orthogonality condition, where learning new tasks disrupts the established knowledge of other subspaces. To avoid this, we integrate orthogonal gradients into the orthogonal loss and propose a new integrated orthogonal loss to constrain the Fake-LoRAs. Experiments demonstrate that our method achieves state-of-the-art average accuracy and the lowest average forgetting rate in continual learning experiments on both the datasets incremental protocol and the manipulation types incremental protocol.

Contributions. We summarize our contributions as follows:

- We propose a developmental MoE architecture to address the continuous emergence of new forgery types. We use a label-guided localized balancing strategy to allocate Real-LoRA to model the real face information, while Fake-LoRAs capture the incremental fake face information.
- Based on theoretical analysis, we integrate orthogonal gradients into the subspace orthogonal loss and form a new integrated orthogonal loss to mitigate the interference of gradients on the knowledge of established subspaces during the entire stages of training.
- Extensive experiments demonstrate that our proposed method achieves the highest average scores and the lowest average forgetting rate among all comparison methods.

2 Related Works

2.1 Face Forgery Detection

Face forgery detection requires the model to classify input images or videos as either real or fake. Related methods [5, 6, 10, 33, 58, 67] focus on identifying forgery cues present in the forged media and can be categorized into two major groups. The first category designs models that incorporate additional information to aid in detection, such as frequency analysis [13, 24, 35, 42], wavelet [36], graphics [72, 73], language descriptions [54], and audio-video consistency [39]. The second category [23, 26, 51] is data-driven, utilizing only real faces and training models through a self-supervised approach. With the continuous evolution of forgery methods, continual learning is applied to forgery detection. DFIL [40] and HDP [55] treat the learning of forgery datasets as a sequence of sub-tasks, requiring the model to learn sequentially across these tasks, achieving satisfactory performance. In this work, we design a Developmental MoE to address the challenges of continual learning in face forgery detection.

2.2 Continual Learning

Continual learning typically models the task as a sequence of sub-tasks, requiring the model to adapt to dynamic data distributions incrementally. The goal is to achieve high performance on newly learned tasks while avoiding catastrophic forgetting of information from previous tasks. Related approaches can be broadly divided into three categories: Regularization-based Approaches [47, 49, 53, 59, 65, 70], Replay-based Approaches [22, 34, 45, 46, 52], and Optimization-based Approaches [11, 21, 31, 32, 41]. Among the latest methods, OrCo [1] enhances generalization in class-incremental learning through orthogonality and contrast. Meanwhile, MoECL [68] adapts to new tasks by introducing an MoE. DFIL [40] and SUR-LID [7] achieve satisfactory results using replay mechanisms. In this work, we treat the iteration of fake faces as an incremental learning problem and employ a developmental architecture to facilitate continual learning.

2.3 Mixture of Experts

The Mixture of Experts (MoE) is a set of sparse models with multiple experts and a routing network. By training MoE in parallel, methods [14, 37, 74] enable efficient fine-tuning of large models with minimal computational cost. MoECL [68] introduces MoE to continual learning, employing an intuitive trainable-freezing strategy to reduce the model's forgetting rate. Low-rank Adaptation (LoRA) [15] further decreases parameter and computational cost by mapping inputs into a low-rank subspace. O-LoRA [60] utilizes LoRAs as experts and constrains the orthogonality of LoRA subspaces. InfLoRA [30] further applies orthogonal LoRA to vision tasks, improving the performance of multiple sub-tasks through pre-defined orthogonal subspaces. MoEFFD [20] integrates MoE and LoRA into face forgery detection, allowing each expert to specialize in extracting features related to different forgery techniques. We use the LoRA module to construct an MoE architecture to address continual face forgery detection.

3 Methodology

We cast forgery detection as a continual learning problem $\mathcal{D} = \{\mathcal{D}_1, \dots, \mathcal{D}_T\}$, where T is the total number of tasks. For any task $\mathcal{D}_t$, let $\mathbf{x}_{i,t}$ be the sample and $y_{i,t}$ its corresponding label, and n_t represent the number of samples in the current task. The model sequentially learns each task

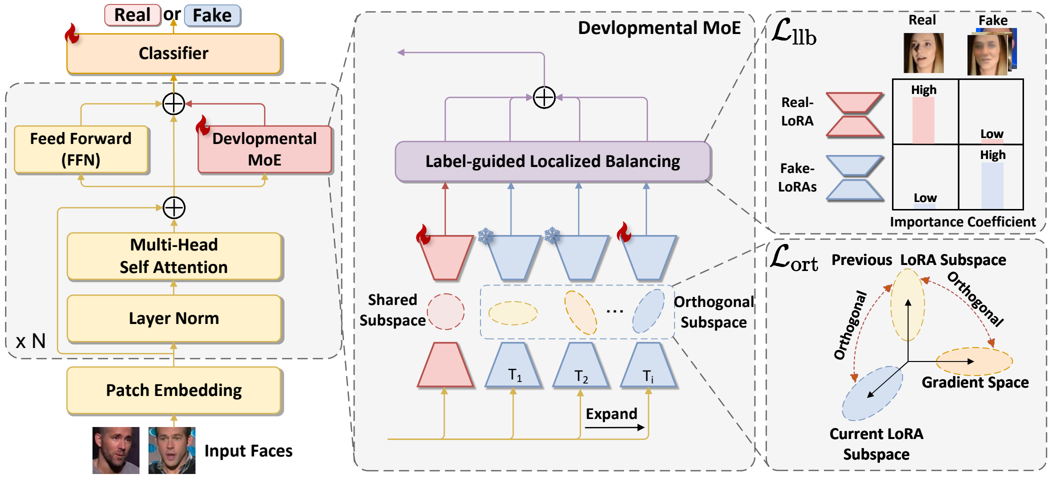


Figure 2: The proposed DevFD framework employs a Developmental MoE architecture to fine-tune the FFN layer in each transformer block. The architecture establishes a developmental LoRA sequence, which adds new branches as the number of tasks increases, enabling the model to handle emerging new types of forgeries. A new label-guided localized balancing strategy allocates the LoRA sequence into two purposes: the Real-LoRA fine-tune and refine knowledge about real faces, while the Fake-LoRAs compose an orthogonal sequence to model the unique cues of fake faces. We integrate orthogonal gradients into the orthogonal loss to alleviate the interference of gradients on previously learned tasks during the training phase when orthogonality is not yet achieved, thereby achieving a lower rate of forgetting.

$\mathcal{D}_t = \{(\mathbf{x}_{i,t}, y_{i,t})\}_{i=1}^{n_t} = \{\mathcal{X}_t, \mathcal{Y}_t\}$ in the task sequence $\mathcal{D}$ to adapt to new types of forgery. Based on the characteristics of the face forgery detection task, we provide the following descriptions:

$$\begin{cases} p\left(\mathcal{X}_i\right) \neq p\left(\mathcal{X}_j\right), \, \mathcal{Y}_i = \mathcal{Y}_j = \text{Fake for } i \neq j, \\ p\left(\mathcal{X}_i\right) = p\left(\mathcal{X}_j\right), \, \mathcal{Y}_i = \mathcal{Y}_j = \text{Real for } i \neq j, \end{cases} \tag{1}$$

where $p\left(\cdot\right)$ is the sample distribution, $\mathcal{X}$ is the samples and $\mathcal{Y}$ is the labels. For fake faces, different forgery types belong to different domains but share the same label $\mathcal{Y} =$ "Fake", presenting a domain incremental problem. In contrast, due to the uniform acquisition methods and the stable and abundant samples of real faces, real faces across datasets exhibit similarity. We aim to model and align them into a unified distribution, i.e., $p\left(\mathcal{X}_i\right) = p\left(\mathcal{X}_j\right)$. We strictly isolate training data across tasks.

3.1 Developmental MoE Architecture

We propose a novel developmental MoE architecture using LoRAs as individual experts to fine-tune pre-trained models within different subspaces; the overview is shown in Fig. 2. First, we employ a Vision Transformer (ViT) pre-trained on real faces as the backbone, denoted as $g_{\mathbf{\Phi}}\left(f_{\mathbf{\Theta}}\left(\cdot\right)\right)$, where $f_{\mathbf{\Theta}}$ represents the pre-trained ViT network, and $g_{\mathbf{\Phi}}$ denotes the classifier. We freeze the parameter $\mathbf{\Theta}$ of the ViT backbone f and introduce LoRA to fine-tune the Feed Forward Network (FFN) weights in each transformer block. For a parameter matrix of any FFN layer $\mathbf{W} \in \mathbb{R}^{d_O \times d_I}$ with input dimension d_I and output dimension d_O, LoRA decomposes the parameter change matrix ΔW into the multiplication of two low-rank matrices:

$$\Delta \mathbf{W} = \mathbf{A}\mathbf{B}, \tag{2}$$

where $\mathbf{B} \in \mathbb{R}^{d_I \times r}$ is the dimensionality reduction matrix and $\mathbf{A} \in \mathbb{R}^{r \times d_O}$ is the dimensionality expansion matrix. Generally, the ranks r of the two matrices are equal and satisfy the condition: $r \ll \min(d_O, d_I)$. To enable the model to adapt to each task, as illustrated in Fig. 2, we design a developmental LoRA sequence as MoE adapters:

$$\begin{aligned} \mathbf{e} &= \text{FFN}(\mathbf{x}) + \text{MoE}(\mathbf{x}) \\ &= \text{FFN}(\mathbf{x}) + \sum_{j=0}^{t} \text{LLB}(\mathbf{A}_j \mathbf{B}_j \mathbf{x}). \end{aligned} \tag{3}$$

Here, $\mathbf{x}$ is the input, while $\mathbf{e}$ is the output. The LoRA modules in the sequence $\{\mathbf{A}_j\mathbf{B}_j\}_{j=0}^{t}$ are assigned by a label-guided localized balancing strategy (LLB) for two purposes: the Real-LoRA $\mathbf{A}_0\mathbf{B}_0$ is designed to learn and refine the common information of real faces modeled by the backbone, while the Fake-LoRAs $\{\mathbf{A}_j\mathbf{B}_j\}_{j=1}^{t}$ compose an orthogonal sequence for learning fake face information specific to each task.

3.2 Learning LoRA Subspaces enhanced by Orthogonal Gradients

The reason for catastrophic forgetting is that parameter adjustments while learning the current task disrupt the knowledge acquired from previous tasks [30]. The subspace established by LoRA can approximately represent the information acquired for a new task [60]. As a developmental structure, DevFD constrains the learning direction of the expanded branches for the new task to prevent interference with previously established subspaces by using an integrated orthogonal loss enhanced with orthogonal gradients. First, the learning of the t-th task can be modeled as follows:

$$\mathbf{e} = \mathbf{W}\mathbf{x} + \sum_{j=1}^{t} \mathbf{A}_j\mathbf{B}_j\mathbf{x} = \mathbf{W}_{t-1}\mathbf{x} + \mathbf{A}_t\mathbf{B}_t\mathbf{x} = \mathbf{W}_t\mathbf{x}, \tag{4}$$

where $\mathbf{W}$ is the parameter matrix of the FFN layer, $\mathbf{e}$ and $\mathbf{x}$ represent the output and input of the network, respectively. Existing methods attempt to use orthogonal loss to control the learning direction of LoRA to be orthogonal to the established subspaces in order to prevent forgetting. However, the forgetting remains severe. We demonstrate that relying solely on the subspace orthogonal loss is insufficient and present an orthogonal gradient solution to further mitigate forgetting.

Why Orthogonal LoRA Subspaces are Still Forgetful? The adaptation of LoRA to new tasks involves fine-tuning the weight matrices within their respective subspaces. LoRA maps the input to the subspace : $\operatorname{span}\{\mathbf{b}_1^t, \ldots, \mathbf{b}_r^t\}$, where the row vectors $\{\mathbf{b}_1^t, \ldots, \mathbf{b}_r^t\}$ of the dimensionality reduction matrix $\mathbf{B}_t$ serve as a basis for this subspace. O-LoRA [60] enforces the orthogonality between the basis $\{\mathbf{b}_1^t, \ldots, \mathbf{b}_r^t\}$ and the basis $\left\{\mathbf{b}_1^i, \ldots, \mathbf{b}_r^i\right\}$ of the forward LoRAs' subspace, ensuring that the subspaces of the LoRA sequence are mutually orthogonal. Let $\mathbf{O_{i,t}}$ be the dot product of two bases of these subspaces. The orthogonality optimization objective can then be expressed as:

$$\underset{\left\{\mathbf{b}_1^t,\ldots,\mathbf{b}_r^t\right\}}{\arg\min} \; \left\|\mathbf{O}_{i,t}\right\|^2 = \underset{\left\{\mathbf{b}_1^t,\ldots,\mathbf{b}_r^t\right\}}{\arg\min} \; \left\|\mathbf{B}_t^T\mathbf{B}_i\right\|^2, \; i = 1, ..., t-1. \tag{5}$$

Notably, the orthogonality condition exists in the form of an optimization objective. We will demonstrate that preventing the disruption of the information in the previous subspaces depends on the strict $\mathbf{B}_t^T\mathbf{B}_i = 0$ throughout the training process. We begin with a proven proposition [30]: when the model learns the t-th task, fine-tuning $\mathbf{A}_t$ is equivalent to fine-tuning the original parameter matrix $\mathbf{W}$ within the subspace $\operatorname{span}\{\mathbf{b}_1^t, \ldots, \mathbf{b}_r^t\}$, denoted as:

$$\Delta_{\mathbf{A}_t}\mathbf{W}_t = \Delta_{\mathbf{W}}\mathbf{W}_t\mathbf{B}_t^T\mathbf{B}_t, \tag{6}$$

where $\Delta_{\mathbf{A}_t}\mathbf{W}_t$ is the increment of the composed matrix $\mathbf{W}_t$ caused by the change of $\mathbf{A}_t$, and $\Delta_{\mathbf{W}}\mathbf{W}_t$ is the increment of the composed matrix $\mathbf{W}_t$ caused by the change of the original parameter $\mathbf{W}$. Thus, a projection matrix $\mathbf{B}_t^T\mathbf{B}_t$ projects each row vector of $\Delta_{\mathbf{W}}\mathbf{W}_t$ into the subspace $\operatorname{span}\{\mathbf{b}_1^t, \ldots, \mathbf{b}_r^t\}$, it follows that the increment $\Delta_{\mathbf{A}_t}\mathbf{W}_t$ due to the fine-tuning of $\mathbf{A}_t$ is equivalent to mapping $\Delta_{\mathbf{W}}\mathbf{W}_t$ into $\operatorname{span}\{\mathbf{b}_1^t, \ldots, \mathbf{b}_r^t\}$. In summary, the increment $\Delta_{\mathbf{A}_t}\mathbf{W}_t$ will not affect the subspace established by previous LoRA, under the assumption that $\operatorname{span}\{\mathbf{b}_1^t, \ldots, \mathbf{b}_r^t\}$ is orthogonal to these subspaces, meaning that the orthogonality condition $\mathbf{B}_t^T\mathbf{B}_i = 0$ needs to hold strictly during the entire training process. Therefore, we argue that catastrophic forgetting occurs in the early stages of training on new tasks when the bases $\mathbf{B}_t$ and $\mathbf{B}_i$ are not perfectly orthogonal. The gradients of the new tasks can disrupt the established subspaces.

Integrate Orthogonal Gradients into Orthogonal Loss. To address the issues discussed above, we choose to directly constrain the gradient space while optimizing the orthogonality of subspaces, resulting in a new integrated orthogonal loss. First, it is proven that the gradient update of a linear or convolution layer lies within the span of the input vectors [29]. We place the proofs in the Appendix B. This proposition allows us to transform the constraints on the gradient space into one on the input space. Let the input matrix be $\mathbf{H} \in \mathbb{R}^{d_I \times n}$, where n is the batch size. Due to the inconsistency in matrix dimensions, we first perform Singular Value Decomposition (SVD) on $\mathbf{H}_t^T$:

$$\mathbf{H}_t^T = \mathbf{U}_t\mathbf{\Sigma}_t\mathbf{V}_t^T. \tag{7}$$

We then choose the rows of $\mathbf{V}_t^T$ corresponding to the top-r singular values and compose $(\mathbf{V}_t^T)_r \in \mathbb{R}^{r\times d_I}$. We use $(\mathbf{V}_t^T)_r$ as an estimate of the gradient space and constrain it to be orthogonal to the previously established subspace. Let $\mathbf{G_{i,t}}$ represent the dot product between the estimated gradient space and the previous subspace basis. The gradient constraint can then be expressed as:

$$\underset{\{\mathbf{b}_1^t,...,\mathbf{b}_r^t\}}{\arg\min} \ \left\|\mathbf{G_{i,t}}\right\|^2 = \underset{\{\mathbf{b}_1^t,...,\mathbf{b}_r^t\}}{\arg\min} \ \left\|(\mathbf{V}_t)_r\mathbf{B}_i\right\|^2, \ i = 1, ..., t-1. \tag{8}$$

In summary, we integrate the orthogonality of LoRA's subspace and the gradient space, designing the integrated orthogonal loss as follows:

$$\mathcal{L}_{\text{ort}} = \frac{1}{t-1}\sum_{i=1}^{t-1}\left(\lambda_1 \sum \left\|\mathbf{O}_{i,t}\right\|^2 + \lambda_2 \sum \left\|\mathbf{G}_{i,t}\right\|^2\right). \tag{9}$$

Here, $\sum\left\|\cdot\right\|^2$ represents the sum of the squares of the matrix elements, λ_1 and λ_2 are hyperparameters. $\mathcal{L}_{\text{ort}}$ constrains the subspace and the gradient space of the current LoRA to be orthogonal to those of the previous LoRAs, thereby alleviating forgetting.

3.3 Label-guided Localized Balancing Strategy

To model the commonalities of real faces and continuously refine the real face knowledge represented by the backbone in a continual learning sequence, we establish an additional LoRA Expert $\mathbf{A}_0\mathbf{B}_0$, referred to as Real-LoRA, and make it learnable across all tasks. However, the Real-LoRA introduces two issues. First, the additional Real-LoRA disrupts the orthogonality of the LoRA sequence, leading to interference with the Fake-LoRA sequence. Second, it cannot be guaranteed that the Real-LoRA will learn real-face information. To address these issues, we propose a label-guided localized balancing strategy. This strategy dynamically adjusts the responses of experts to different types of samples based on the labels during training. Specifically, it enables Real-LoRA to focus more on real faces, while Fake-LoRAs concentrate on learning forged faces in an orthogonal manner, preventing Real-LoRA from disrupting the orthogonality of Fake-LoRAs. The proposed strategy first takes the inputs from all LoRAs and sets a response matrix $\mathbf{I} \in \mathbb{R}^{(t+1)\times n}$, where $t+1$ indicates the total number of experts. An element $\mathbf{I}_{k,l}$ indicates the response of the k-th expert to the l-th sample in the batch. It is defined as follows:

$$\mathbf{I}_{k,l} = \sum_{j=1}^{T_m} \frac{\exp\left(\omega_{k,l}^j/\tau\right)}{\sum_{i=0}^{t}\exp\left(\omega_{i,l}^j/\tau\right)}. \tag{10}$$

Here, $\omega_{k,l}^j$ represents the output of the k-th expert for the j-th token in the l-th sample, T_m denotes the number of tokens in the l-th sample. All outputs of LoRA can be defined as a vector matrix $\mathbf{O} \in \mathbb{R}^{(t+1)\times n\times d_o}$, where each element is defined as the output of the k-th expert for the l-th sample. Based on the response matrix $\mathbf{I}$ and the output matrix $\mathbf{O}$, the forward process computes the final output $\mathbf{e} \in \mathbb{R}^{n\times d_o}$, where the output vector $\mathbf{e}_l \in \mathbb{R}^{d_o}$ for the l-th sample is defined as:

$$\mathbf{e}_l = \sum_{k=0}^{t} \mathbf{I}_{k,l} \cdot \mathbf{O}_{k,l} \tag{11}$$

To allocate Real-LoRA and Fake-LoRAs for modeling the information of real and fake faces, respectively, we utilize labels to define a balancing coefficient matrix $\mathbf{C} \in \mathbb{R}^{(t+1)\times n}$. An element $\mathbf{C}_{k,l}$ represents the balancing weight applied to the response at the corresponding position $\mathbf{I}_{k,l}$:

$$\mathbf{C}_{k,l} = \left\{\begin{array}{ll} 1-\delta, & \text{Type}_e(k) = \text{Type}_h(l) \\ 1+\delta, & \text{Type}_e(k) \neq \text{Type}_h(l) \end{array}\right., \tag{12}$$

where $\text{Type}_e(k) = \text{Type}_h(l)$ indicates that the expert matches the category label corresponding to the sample, such as Real-LoRA for real faces, the coefficient matrix $\mathbf{C}$ decreases the balancing coefficient for this item; otherwise, $\mathbf{C}$ increases the balancing coefficient. The increment δ of the coefficient falls within the range of $[0, 1]$. Based on the response matrix $\mathbf{I}$ and the balancing coefficient matrix $\mathbf{C}$, the LLB loss constrains the dispersion of the weighted response matrix $\mathbf{I} \circ \mathbf{C}$:

$$\mathcal{L}_{\text{llb}} = \frac{\sigma^2(\mathbf{I}\circ\mathbf{C})}{\mu(\mathbf{I}\circ\mathbf{C})}, \tag{13}$$

where $\sigma^2(\cdot)$ and $\mu(\cdot)$ represent the variance and mean, respectively, and $\circ$ is element-wise multiplication. This loss function forces the weighted responses ($\mathbf{I} \circ \mathbf{C}$) to become as balanced as possible. By optimizing this loss function, the model is thus compelled to generate a greater original response $\mathbf{I}_{k,l}$ for the matching pairs (to compensate for their small $1-\delta$ balancing coefficients) and a smaller original response for the mismatched pairs. Additionally, this router does not select specific LoRAs but encourages collaboration among all LoRAs.

When the model begins to learn a new task, DevFD establishes a new LoRA branch and freeze the other Fake-LoRAs in the orthogonal sequence $\{\mathbf{A}_j\mathbf{B}_j\}_{j=1}^{t}$, allowing only the newly added LoRA $\mathbf{A}_t\mathbf{B}_t$ and the Real-LoRA $\mathbf{A}_0\mathbf{B}_0$ to be trainable. During inference, it aggregates the outputs of all LoRA experts to make a joint decision, fostering collaboration among them.

In summary, we define the total loss as the weighted sum of the binary cross-entropy loss $\mathcal{L}_{\text{cls}}$, the orthogonal loss $\mathcal{L}_{\text{ort}}$, and the label-guided localized balancing loss $\mathcal{L}_{\text{llb}}$:

$$\mathcal{L} = \mathcal{L}_{\text{cls}} + \mathcal{L}_{\text{ort}} + \lambda_3 \mathcal{L}_{\text{llb}}, \tag{14}$$

where λ_3 is a hyperparameter. This loss allows real faces and fake faces from different forgery manipulation types to be modeled by distinct LoRAs without mutual interference and encourages collaboration among all LoRAs in this MoE architecture.

4 Experiments

4.1 Experimental Setup

Datasets. To facilitate comparisons, we select multiple datasets, employing different forgery methods to manipulate and generate faces from distinct domains. FaceForensics++ (FF++) [48] contains 1K real face videos. These faces are manipulated by four forgery methods, resulting in a total of 4K fake videos. The Deepfake Detection (DFD) [12] dataset comprises over a hundred real face samples and more than 1K forged face samples. For the Deepfake Detection Challenge (DFDC) [9] dataset, we utilized the Preview version (DFDC-P), which includes 5K videos and two forgery methods. Celeb-DF v2 (CDF2) [27] comprises 590 real and 5,639 fake videos. DF40 [66] is a large-scale dataset comprising 40 types of forgery methods and over 100K fake videos. We select [FF++, DFDC-P, DFD, CDF2] as our dataset-incremental testing protocol. From DF40, we choose three forgery methods corresponding to different forgery types: Face-Swapping (FS): BlendFace, Face-Reenactment (FR): MCNet, and Entire Face Synthesis (EFS): StyleGAN3. The datasets of selected forgery methods and a base dataset with hybrid forgery types FF++ jointly compose our forgery-type-incremental testing protocol: [Hybrid, FR, FS, EFS].

Implementation Details and Metrics. We utilize a pre-trained ViT model as the initial backbone. The pre-training data is sourced from real faces in the FF++ dataset. We conduct self-supervised pre-training through a data augmentation approach [51]. Regarding data utilization, we strictly isolate data across tasks, including real and fake faces. Similar to DFIL [40], we randomly sample 100 videos from the current dataset and train for 20 epochs in each task, maintaining a balance between real and fake samples. The hyperparameter settings are adjusted according to the number of training epochs, and detailed information is provided in Appendix C.2. We employed the Adam optimizer with parameters set to $\beta_1 = 0.9, \beta_2 = 0.999$. The learning rate is set to $1e-4$, and the batch size is set to 128. The rank (subspace dimension) for LoRA was determined using a grid optimization algorithm similar to InfLoRA [30]. We utilize Accuracy (Acc) and Area Under the Curve (AUC) to evaluate the overall performance of the model. Average Forgetting (AF) indicates the model's ability to retain information from previously learned tasks. Let $a_{t,i}$ denote the evaluation $Score$ (Acc or AUC) on task i after the model has been trained up to task t. The AF_T (for $T > 1$) is defined as:

$$AF_T = \frac{1}{T-1} \sum_{i=1}^{T-1} \left(a_{i,i} - a_{T,i} \right), \tag{15}$$

where $a_{T,i}$ is the score on task i after learning the final task T.

4.2 Comparison Experiments

Experiments on Dataset Incremental Protocol. To comprehensively evaluate the performance of the proposed method, we conduct comparative experiments with general continual learning

Table 1: **Left:** Experiments on dataset incremental protocol. **Right:** Experiments on manipulation types incremental protocol. The best performer is highlighted in boldface, while the second-best result is underlined. Shadowed lines indicate the results from our method.

Dataset Incremental Protocol

Method	Dataset	Acc(%)↑				Avg↑	AF↓
		FF++	DFDCP	DFD	CDF2		
LWF [28]	FF++	95.52	-	-	-	95.52	-
	DFDCP	87.83	81.57	-	-	84.70	7.69
	DFD	76.16	41.78	96.36	-	71.43	29.58
	CDF2	67.34	67.43	84.05	87.90	76.68	18.21
ER [3]	FF++	95.52	-	-	-	95.52	-
	DFDCP	92.25	88.53	-	-	90.39	3.27
	DFD	84.69	80.59	94.00	-	86.42	9.39
	CDF2	71.65	69.40	92.98	83.26	79.32	14.67
SI [69]	FF++	95.51	-	-	-	95.51	-
	DFDCP	90.88	88.38	-	-	89.63	4.63
	DFD	45.46	27.02	96.67	-	56.38	55.71
	CDF2	36.67	42.04	44.83	78.88	50.60	52.34
CoReD [19]	FF++	95.50	-	-	-	95.50	-
	DFDCP	92.94	87.61	-	-	90.28	2.56
	DFD	86.84	81.07	95.22	-	87.71	7.60
	CDF2	74.08	76.59	93.41	80.78	81.22	11.42
DFIL [40]	FF++	95.67	-	-	-	95.67	-
	DFDCP	93.15	88.87	-	-	91.01	2.52
	DFD	90.30	85.42	94.67	-	90.13	4.41
	CDF2	86.28	79.53	92.36	83.81	85.49	7.01
DMP [56]	FF++	95.96	-	-	-	95.96	-
	DFDCP	92.71	89.72	-	-	91.22	3.25
	DFD	92.64	86.09	94.84	-	91.19	3.48
	CDF2	91.61	84.86	91.81	91.67	**89.99**	4.08
DevFD (Ours)	FF++	98.41	-	-	-	**98.41**	-
	DFDC-P	97.06	89.90	-	-	**93.48**	**1.35**
	DFD	92.44	89.07	97.91	-	**93.14**	**3.40**
	CDF2	90.71	90.31	93.12	85.15	89.82	**4.03**

Manipulation Types Incremental Protocol

Method	Mani. Type	AUC(%)↑				Avg↑	AF↓
		Hybrid	FR	FS	EFS		
iCaRL [44]	Hybrid	96.53	-	-	-	96.53	-
	FR	67.36	99.89	-	-	83.63	29.17
	FS	73.79	66.24	97.54	-	79.19	28.20
	EFS	52.98	55.38	64.74	100.0	68.28	40.29
DER [64]	Hybrid	97.00	-	-	-	97.00	-
	FR	59.03	99.73	-	-	79.38	37.97
	FS	68.15	19.68	97.94	-	61.93	54.45
	EFS	56.79	59.83	65.36	100.0	70.49	37.56
CoReD [19]	Hybrid	96.65	-	-	-	96.65	-
	FR	93.55	79.88	-	-	86.72	3.10
	FS	89.07	79.29	86.05	-	84.80	4.09
	EFS	84.54	64.29	84.17	92.63	81.41	9.86
DFIL [40]	Hybrid	96.46	-	-	-	96.46	-
	FR	55.74	99.75	-	-	77.75	40.72
	FS	60.71	66.49	99.03	-	75.41	34.51
	EFS	50.83	95.56	70.81	99.96	79.29	26.01
HDP [55]	Hybrid	96.71	-	-	-	96.71	-
	FR	67.41	95.45	-	-	81.43	29.30
	FS	63.00	71.35	95.09	-	76.48	28.91
	EFS	59.89	70.06	89.34	93.73	78.26	22.65
SUR-LID [7]	Hybrid	96.85	-	-	-	96.85	-
	FR	82.91	92.42	-	-	87.66	13.94
	FS	90.50	96.26	97.94	-	**94.90**	**1.26**
	EFS	87.90	96.79	93.56	99.07	94.33	2.99
DevFD (Ours)	Hybrid	97.63	-	-	-	**97.63**	-
	FR	94.69	93.07	-	-	**93.88**	**2.94**
	FS	90.97	92.76	97.05	-	93.59	3.49
	EFS	90.86	94.35	95.23	99.16	**94.90**	**2.44**

approaches (e.g., LwF [28], ER [3], and SI [69]) and specialized continual face forgery detection methods (e.g., CoReD [19], DFIL [40], and DMP [56]). We ensure that the improvements stem from the new strategy and maintain the fairness of the experiments, as detailed in Appendix C.1. Under the dataset incremental protocol, the model is sequentially trained on task sequences constructed from [FF++, DFDC-P, DFD, CDF2]. After completing each task's training phase, we evaluate model accuracy on both the current and previously learned tasks. Quantitative results are organized in a lower triangular accuracy matrix, accompanied by two key metrics: average accuracy (Avg) and average forgetting (AF), as detailed in Table 1-Left. The results reveal that general continual learning methods initially achieve satisfactory accuracy but suffer severe performance degradation on previous tasks as training progresses, resulting in higher forgetting rates. In contrast, continual forgery detection approaches exhibit more stable performance but still suffer from forgetting. It is worth noting that many of the compared methods [7, 40] utilize a replay set, while our method strictly isolates all subtask data and still achieves superior anti-forgetting and state-of-the-art performance.

Experiments on Manipulation Types Incremental Protocol. The dataset incremental protocol tested above demonstrates our method's superior performance and anti-forgetting capability. Considering potential issues in the dataset incremental protocol, such as the domain gaps between different datasets and the possibility of shared manipulation methods across datasets, we perform experiments under the manipulation types incremental protocol [Hybrid, FR, FS, EFS], employing AUC as the primary metric for comparison. The experimental results are presented in Table 1-Right. The results show general continual learning approaches (LwF [28], iCaRL [44], DER [64]) exhibit unstable performance with high forgetting rates. Even specialized forgery detection methods demonstrate initial training instability – DFIL [40] shows a 40.72% performance drop on previous tasks during FR-type training. Our method maintains stable learning throughout the entire training sequence, achieving both the highest final performance and the lowest forgetting rate. Comparative analysis reveals significant advantages: our approach surpasses SUR-LID [7] by 0.57% in average AUC, with

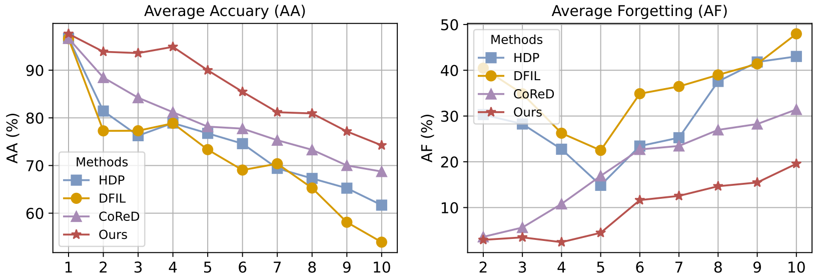


Figure 3: The Task10 long-sequence continual learning experiment based on DF40, the proposed method achieves the highest average accuracy and the lowest forgetting rate.

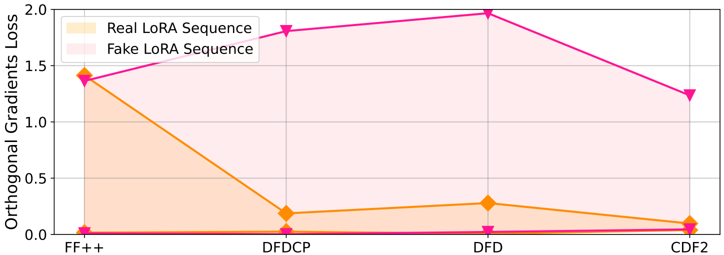


Figure 4: Motivation Validation. We track the variation range of the orthogonal gradient loss during the training process for two orthogonal sequences, one modeling real faces and the other modeling forged faces. For each sequence, the two lines represent the upper and lower bounds of the loss, and the shaded area between them indicates its range of variation. We observe that real faces exhibit an obvious smaller orthogonal gradient loss.

0.55% lower forgetting rate, and outperforms HDP [55] by 16.64% AUC margin while reducing forgetting rate by 20.21%. These results validate our method's performance against manipulation-type shifts in continual learning scenarios.

4.3 Long-sequence Continual Learning Experiments

To further validate the effectiveness of our proposed method on more manipulation types and extended task sequences, we selected 10 datasets from DF40 and completed long-sequence continual learning experiments under the Task10 protocol. We provide more details in the Appendix C.3. The average AUC metric and forgetting rate are plotted as shown in Fig. 3. The experimental results indicate that in long-sequence learning, both DFIL [40] and HDP [55] exhibit unstable catastrophic forgetting. At the same time, our method achieves the highest average AUC and the lowest forgetting rates.

4.4 Motivation Validation Experiments

Our proposed method is based on a key observation: abundant and unbiased real faces exhibit a more compact distribution across datasets than forged faces created by various methods. Building upon a t-SNE experiment that validates this observation, we further confirm that real faces occupy a more similar input space than forged faces within our proposed framework. We first theoretically demonstrate that a more similar input space corresponds to a smaller orthogonal gradient loss. Specifically, when a new orthogonal LoRA is configured to model a highly similar domain, this domain will possess a highly similar input space $\mathbf{H}_t \in \mathbb{R}^{d_I \times n}$. From Eqn. 7, it can be deduced that the resulting $\mathbf{V}_t^T$ will also be highly similar. Consequently, in Eqn. 8 and Eqn. 9, the term $\sum \|\mathbf{G}_{i,t}\|^2$ yields a near-zero value because its components are approximately orthogonal.

Based on this, we design a new experiment consisting of two separate orthogonal LoRA sequences: one for modeling real faces and the other for modeling forged faces. These sequences learn from the task series [FF++, DFDCP, DFD, CDF2]. We record the range of orthogonal gradient loss values for the two sequences during the training process to investigate the cross-dataset similarity of their respective input spaces. The experimental results are presented in Fig. 4. Starting from the second task, the orthogonal gradient induced by the real face data for a new LoRA is smaller than that induced by the forged face data. The results suggest that real faces from different datasets share a

Table 2: Ablation study on the loss functions. AA represents the average accuracy, while AF indicates the average forgetting.

Loss	FF++	DFDC-P		DFD		CDF2	
	AA↑	AA↑	AF↓	AA↑	AF↓	AA↑	AF↓
$\mathcal{L}_{\text{cls}}$	98.18	90.49	9.33	83.86	15.79	78.82	23.93
$\mathcal{L}_{\text{cls}} + \mathcal{L}_{\text{olora}}$	97.30	91.08	4.53	89.71	9.01	87.66	10.06
$\mathcal{L}_{\text{cls}} + \mathcal{L}_{\text{ort}}$	97.35	93.59	2.68	92.08	6.82	**90.66**	7.91
$\mathcal{L}_{\text{cls}} + \mathcal{L}_{\text{llb}}$	98.09	**94.20**	4.45	86.75	11.19	83.07	19.31
All	**98.41**	93.48	**1.35**	**93.14**	**3.40**	89.82	**4.03**

Table 3: Ablation study on the LoRA modules. The best performer is highlighted in boldface. Shadowed lines indicate the results from our method.

Module		FF++	DFDC-P		DFD		CDF2	
Real-LoRA Num.	Fake-LoRA Num.	AA↑	AA↑	AF↓	AA↑	AF↓	AA↑	AF↓
1	-	94.02	88.94	11.44	79.05	19.51	75.69	30.80
-	4	97.06	93.07	2.54	91.31	4.20	88.39	6.42
1	4	**98.41**	**93.48**	1.35	**93.14**	**3.40**	89.82	4.03
4	4	98.22	93.25	**1.33**	93.00	3.45	**90.58**	**3.72**

similar input space $\mathbf{H}_t$, leading to an orthogonal gradient loss that is an order of magnitude smaller. Moreover, the orthogonal LoRA sequence introduces additional computational overhead compared to a single Real-LoRA. This experiment mutually validates our initial t-SNE results, demonstrating that using a single, shared Real-LoRA to model real faces is both reasonable and necessary.

4.5 Ablation Study

Effect of $\mathcal{L}_{\text{ort}}$ and $\mathcal{L}_{\text{llb}}$. We employ orthogonal loss $\mathcal{L}_{\text{ort}}$ to constrain both the subspace and gradient space of the currently learned LoRA and use $\mathcal{L}_{\text{llb}}$ to allocate the Real-LoRA and Fake-LoRAs. We conduct ablation studies on both losses, with the experimental results presented in Table 2. $\mathcal{L}_{\text{olora}}$ denotes the use of orthogonal subspaces without orthogonal gradients to validate the effectiveness of our orthogonal gradients; "All" represents the model's performance incorporating all three losses. The results reveal that the $\mathcal{L}_{\text{ort}}$ significantly enhance the model's anti-forgetting capability. Meanwhile, $\mathcal{L}_{\text{llb}}$ primarily improves the model's Average Accuracy (AA) by facilitating task-specific allocations among LoRAs and maintaining a weighted response matrix for collaboration.

Effect of Real-LoRA and Fake-LoRAs. To study the effectiveness of the designed LoRA modules, we separately isolate the two types of LoRA and report the results in Table 3. We differentiate our models based on the number of different types of LoRAs used. Specifically, '4' indicates the use of an orthogonal LoRA sequence (with a sequence length of 4 tasks), '1' indicates the use of a single global LoRA, and '-' indicates that no LoRA module is used. The $\mathcal{L}_{\text{llb}}$ is introduced only when both types of LoRAs are used simultaneously. To investigate whether the global Real-LoRA would lead to catastrophic forgetting of real faces, we add an additional experiment shown in the last row by expanding the shared Real-LoRA into a sequence identical to that of Fake-LoRAs, i.e., using two orthogonal LoRA sequences to learn real and fake samples separately. The experimental results show that Fake-LoRAs have a significant effect on mitigating catastrophic forgetting. After incorporating both types of LoRA, the addition of Real-LoRA improves both accuracy and the forgetting rate. Moreover, replacing Real-LoRA with an orthogonal sequence does not bring significant improvements in forgetting rate or accuracy. Therefore, the Real-LoRA does not cause catastrophic forgetting.

5 Conclusion

In this paper, we propose DevFD, a Developmental MoE architecture for continual face forgery detection. By utilizing LoRA models as experts, DevFD expands with new LoRA branches to adapt to emerging types of forgery. We employ a label-guided localized balancing strategy to allocate all experts for two purposes: the Real-LoRA refines the real face knowledge modeled by the backbone continuously, while the Fake-LoRAs capture incremental forgery cues from different types. To prevent catastrophic forgetting, we use an orthogonality loss to constrain the learning direction of the learning LoRA to be orthogonal to the existing subspaces. Additionally, we integrate orthogonal gradients into the subspace orthogonal loss to mitigate the interference of gradients on the established subspace knowledge during the entire training phase. Experimental results demonstrate that our approach achieves the best performance.

Acknowledgement

This work was supported in part by Chinese National Natural Science Foundation Projects U23B2054, 62276254, 62176256, Beijing Natural Science Foundation L242092, the Science and Technology Development Fund of Macau Project 0140/2024/AGJ.

References

[1] Noor Ahmed, Anna Kukleva, and Bernt Schiele. Orco: Towards better generalization via orthogonality and contrast for few-shot class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 28762–28771, 2024.

[2] Stella Bounareli, Christos Tzelepis, Vasileios Argyriou, Ioannis Patras, and Georgios Tzimiropoulos. Stylemask: Disentangling the style space of stylegan2 for neural face reenactment. In *2023 IEEE 17th international conference on automatic face and gesture recognition (FG)*, pages 1–8. IEEE, 2023.

[3] Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet K Dokania, Philip HS Torr, and Marc'Aurelio Ranzato. On tiny episodic memories in continual learning. *arXiv preprint arXiv:1902.10486*, 2019.

[4] Jun Chen, Deyao Zhu, Xiaoqian Shen, Xiang Li, Zechu Liu, Pengchuan Zhang, Raghuraman Krishnamoorthi, Vikas Chandra, Yunyang Xiong, and Mohamed Elhoseiny. Minigpt-v2: large language model as a unified interface for vision-language multi-task learning. *arXiv preprint arXiv:2310.09478*, 2023.

[5] Liang Chen, Yong Zhang, Yibing Song, Lingqiao Liu, and Jue Wang. Self-supervised learning of adversarial example: Towards good generalizations for deepfake detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 18710–18719, 2022.

[6] Haonan Cheng, Hanyue Liu, Juanjuan Cai, and Long Ye. Clformer: a cross-lingual transformer framework for temporal forgery localization. *Visual Intelligence*, 3(1):1–13, 2025.

[7] Jikang Cheng, Zhiyuan Yan, Ying Zhang, Li Hao, Jiaxin Ai, Qin Zou, Chen Li, and Zhongyuan Wang. Stacking brick by brick: Aligned feature isolation for incremental face forgery detection. *arXiv preprint arXiv:2411.11396*, 2024.

[8] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258, 2017.

[9] 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.

[10] Xiaoyi Dong, Jianmin Bao, Dongdong Chen, Ting Zhang, Weiming Zhang, Nenghai Yu, Dong Chen, Fang Wen, and Baining Guo. Protecting celebrities from deepfake with identity consistency transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9468–9478, 2022.

[11] Arthur Douillard, Matthieu Cord, Charles Ollion, Thomas Robert, and Eduardo Valle. Podnet: Pooled outputs distillation for small-tasks incremental learning. In *Computer vision–ECCV 2020: 16th European conference, Glasgow, UK, August 23–28, 2020, proceedings, part XX 16*, pages 86–102. Springer, 2020.

[12] Nick Dufour and Andrew Gully. Contributing data to deepfake detection research. *Google AI Blog*, 1(2):3, 2019.

[13] Joel Frank, Thorsten Eisenhofer, Lea Schönherr, Asja Fischer, Dorothea Kolossa, and Thorsten Holz. Leveraging frequency analysis for deep fake image recognition. In *International conference on machine learning*, pages 3247–3258. PMLR, 2020.

[14] Jiaao He, Jidong Zhai, Tiago Antunes, Haojie Wang, Fuwen Luo, Shangfeng Shi, and Qin Li. Fastermoe: modeling and optimizing training of large-scale dynamic pre-trained models. In *Proceedings of the 27th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming*, pages 120–134, 2022.

[15] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

[16] Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. Openclip, 2021.

[17] Gyeongman Kim, Hajin Shim, Hyunsu Kim, Yunjey Choi, Junho Kim, and Eunho Yang. Diffusion video autoencoders: Toward temporally consistent face video editing via disentangled video encoding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6091–6100, 2023.

[18] Kihong Kim, Yunho Kim, Seokju Cho, Junyoung Seo, Jisu Nam, Kychul Lee, Seungryong Kim, and KwangHee Lee. Diffface: Diffusion-based face swapping with facial guidance. *arXiv preprint arXiv:2212.13344*, 2022.

[19] Minha Kim, Shahroz Tariq, and Simon S Woo. Cored: Generalizing fake media detection with continual representation using distillation. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 337–346, 2021.

[20] Chenqi Kong, Anwei Luo, Song Xia, Yi Yu, Haoliang Li, and Alex C Kot. Moe-ffd: Mixture of experts for generalized and parameter-efficient face forgery detection. *arXiv preprint arXiv:2404.08452*, 2024.

[21] Yajing Kong, Liu Liu, Zhen Wang, and Dacheng Tao. Balancing stability and plasticity through advanced null space in continual learning. In *European Conference on Computer Vision*, pages 219–236. Springer, 2022.

[22] Vinod K Kurmi, Badri N Patro, Venkatesh K Subramanian, and Vinay P Namboodiri. Do not forget to attend to uncertainty while mitigating catastrophic forgetting. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 736–745, 2021.

[23] Nicolas Larue, Ngoc-Son Vu, Vitomir Struc, Peter Peer, and Vassilis Christophides. Seeable: Soft discrepancies and bounded contrastive learning for exposing deepfakes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 21011–21021, 2023.

[24] Jiaming Li, Hongtao Xie, Jiahong Li, Zhongyuan Wang, and Yongdong Zhang. Frequency-aware discriminative feature learning supervised by single-center loss for face forgery detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6458–6467, 2021.

[25] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR, 2023.

[26] Lingzhi Li, Jianmin Bao, Ting Zhang, Hao Yang, Dong Chen, Fang Wen, and Baining Guo. Face x-ray for more general face forgery detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5001–5010, 2020.

[27] Yuezun Li, Xin Yang, Pu Sun, Honggang Qi, and Siwei 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.

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

[29] Yan-Shuo Liang and Wu-Jun Li. Adaptive plasticity improvement for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7816–7825, 2023.

[30] Yan-Shuo Liang and Wu-Jun Li. Inflora: Interference-free low-rank adaptation for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23638–23647, 2024.

[31] Sen Lin, Li Yang, Deliang Fan, and Junshan Zhang. Trgp: Trust region gradient projection for continual learning. *arXiv preprint arXiv:2202.02931*, 2022.

[32] Hao Liu and Huaping Liu. Continual learning with recursive gradient optimization. *arXiv preprint arXiv:2201.12522*, 2022.

[33] Honggu Liu, Xiaodan Li, Wenbo Zhou, Yuefeng Chen, Yuan He, Hui Xue, Weiming Zhang, and Nenghai Yu. Spatial-phase shallow learning: rethinking face forgery detection in frequency domain. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 772–781, 2021.

[34] David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. *Advances in neural information processing systems*, 30, 2017.

[35] Yuchen Luo, Yong Zhang, Junchi Yan, and Wei Liu. Generalizing face forgery detection with high-frequency features. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16317–16326, 2021.

[36] Changtao Miao, Zichang Tan, Qi Chu, Huan Liu, Honggang Hu, and Nenghai Yu. F 2 trans: High-frequency fine-grained transformer for face forgery detection. *IEEE Transactions on Information Forensics and Security*, 18:1039–1051, 2023.

[37] Basil Mustafa, Carlos Riquelme, Joan Puigcerver, Rodolphe Jenatton, and Neil Houlsby. Multimodal contrastive learning with limoe: the language-image mixture of experts. *Advances in Neural Information Processing Systems*, 35:9564–9576, 2022.

[38] Trevine Oorloff and Yaser Yacoob. Robust one-shot face video re-enactment using hybrid latent spaces of stylegan2. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 20947–20957, 2023.

[39] Trevine Oorloff, Surya Koppisetti, Nicolò Bonettini, Divyaraj Solanki, Ben Colman, Yaser Yacoob, Ali Shahriyari, and Gaurav Bharaj. Avff: Audio-visual feature fusion for video deepfake detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 27102–27112, 2024.

[40] Kun Pan, Yifang Yin, Yao Wei, Feng Lin, Zhongjie Ba, Zhenguang Liu, Zhibo Wang, Lorenzo Cavallaro, and Kui Ren. Dfil: Deepfake incremental learning by exploiting domain-invariant forgery clues. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 8035–8046, 2023.

[41] Pingbo Pan, Siddharth Swaroop, Alexander Immer, Runa Eschenhagen, Richard Turner, and Mohammad Emtiyaz E Khan. Continual deep learning by functional regularisation of memorable past. *Advances in neural information processing systems*, 33:4453–4464, 2020.

[42] Yuyang Qian, Guojun Yin, Lu Sheng, Zixuan Chen, and Jing Shao. Thinking in frequency: Face forgery detection by mining frequency-aware clues. In *European conference on computer vision*, pages 86–103. Springer, 2020.

[43] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[44] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2001–2010, 2017.

[45] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2001–2010, 2017.

[46] 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.

[47] Hippolyt Ritter, Aleksandar Botev, and David Barber. Online structured laplace approximations for overcoming catastrophic forgetting. *Advances in Neural Information Processing Systems*, 31, 2018.

[48] Andreas Rossler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Nießner. Faceforensics++: Learning to detect manipulated facial images. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1–11, 2019.

[49] Jonathan Schwarz, Wojciech Czarnecki, Jelena Luketina, Agnieszka Grabska-Barwinska, Yee Whye Teh, Razvan Pascanu, and Raia Hadsell. Progress & compress: A scalable framework for continual learning. In *International conference on machine learning*, pages 4528–4537. PMLR, 2018.

[50] Yichen Shi, Yuhao Gao, Yingxin Lai, Hongyang Wang, Jun Feng, Lei He, Jun Wan, Changsheng Chen, Zitong Yu, and Xiaochun Cao. Shield: An evaluation benchmark for face spoofing and forgery detection with multimodal large language models. *Visual Intelligence*, 3(1):9, 2025.

[51] Kaede Shiohara and Toshihiko Yamasaki. Detecting deepfakes with self-blended images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18720–18729, 2022.

[52] Christian Simon, Piotr Koniusz, and Mehrtash Harandi. On learning the geodesic path for incremental learning. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pages 1591–1600, 2021.

[53] Pravendra Singh, Pratik Mazumder, Piyush Rai, and Vinay P Namboodiri. Rectification-based knowledge retention for continual learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15282–15291, 2021.

[54] Ke Sun, Shen Chen, Taiping Yao, Haozhe Yang, Xiaoshuai Sun, Shouhong Ding, and Rongrong Ji. Towards general visual-linguistic face forgery detection. *arXiv preprint arXiv:2307.16545*, 2023.

[55] Ke Sun, Shen Chen, Taiping Yao, Xiaoshuai Sun, Shouhong Ding, and Rongrong Ji. Continual face forgery detection via historical distribution preserving. *International Journal of Computer Vision*, pages 1–18, 2024.

[56] Jiahe Tian, Cai Yu, Xi Wang, Peng Chen, Zihao Xiao, Jizhong Han, and Yesheng Chai. Dynamic mixed-prototype model for incremental deepfake detection. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 8129–8138, 2024.

[57] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[58] Lantao Wang and Chao Ma. Adapting pretrained large-scale vision models for face forgery detection. In *International Conference on Multimedia Modeling*, pages 71–85. Springer, 2024.

[59] Liyuan Wang, Mingtian Zhang, Zhongfan Jia, Qian Li, Chenglong Bao, Kaisheng Ma, Jun Zhu, and Yi Zhong. Afec: Active forgetting of negative transfer in continual learning. *Advances in Neural Information Processing Systems*, 34:22379–22391, 2021.

[60] Xiao Wang, Tianze Chen, Qiming Ge, Han Xia, Rong Bao, Rui Zheng, Qi Zhang, Tao Gui, and Xuanjing Huang. Orthogonal subspace learning for language model continual learning. *arXiv preprint arXiv:2310.14152*, 2023.

[61] Mingwang Xu, Hui Li, Qingkun Su, Hanlin Shang, Liwei Zhang, Ce Liu, Jingdong Wang, Luc Van Gool, Yao Yao, and Siyu Zhu. Hallo: Hierarchical audio-driven visual synthesis for portrait image animation. *arXiv preprint arXiv:2406.08801*, 2024.

[62] Zhiliang Xu, Hang Zhou, Zhibin Hong, Ziwei Liu, Jiaming Liu, Zhizhi Guo, Junyu Han, Jingtuo Liu, Errui Ding, and Jingdong Wang. Styleswap: Style-based generator empowers robust face swapping. In *European Conference on Computer Vision*, pages 661–677. Springer, 2022.

[63] Le Xue, Manli Shu, Anas Awadalla, Jun Wang, An Yan, Senthil Purushwalkam, Honglu Zhou, Viraj Prabhu, Yutong Dai, Michael S Ryoo, et al. xgen-mm (blip-3): A family of open large multimodal models. *arXiv preprint arXiv:2408.08872*, 2024.

[64] Shipeng Yan, Jiangwei Xie, and Xuming He. Der: Dynamically expandable representation for class incremental learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3014–3023, 2021.

[65] Shipeng Yan, Jiangwei Xie, and Xuming He. Der: Dynamically expandable representation for class incremental learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3014–3023, 2021.

[66] Zhiyuan Yan, Taiping Yao, Shen Chen, Yandan Zhao, Xinghe Fu, Junwei Zhu, Donghao Luo, Li Yuan, Chengjie Wang, Shouhong Ding, et al. Df40: Toward next-generation deepfake detection. *arXiv preprint arXiv:2406.13495*, 2024.

[67] Zhiyuan Yan, Yandan Zhao, Shen Chen, Mingyi Guo, Xinghe Fu, Taiping Yao, Shouhong Ding, Yunsheng Wu, and Li Yuan. Generalizing deepfake video detection with plug-and-play: Video-level blending and spatiotemporal adapter tuning. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 12615–12625, 2025.

[68] Jiazuo Yu, Yunzhi Zhuge, Lu Zhang, Ping Hu, Dong Wang, Huchuan Lu, and You He. Boosting continual learning of vision-language models via mixture-of-experts adapters. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23219–23230, 2024.

[69] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *International conference on machine learning*, pages 3987–3995. PMLR, 2017.

[70] Chi Zhang, Nan Song, Guosheng Lin, Yun Zheng, Pan Pan, and Yinghui Xu. Few-shot incremental learning with continually evolved classifiers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12455–12464, 2021.

[71] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*, 2023.

[72] Xiangyu Zhu, Hao Wang, Hongyan Fei, Zhen Lei, and Stan Z Li. Face forgery detection by 3d decomposition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2929–2939, 2021.

[73] Xiangyu Zhu, Hongyan Fei, Bin Zhang, Tianshuo Zhang, Xiaoyu Zhang, Stan Z Li, and Zhen Lei. Face forgery detection by 3d decomposition and composition search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(7):8342–8357, 2023.

[74] Barret Zoph, Irwan Bello, Sameer Kumar, Nan Du, Yanping Huang, Jeff Dean, Noam Shazeer, and William Fedus. St-moe: Designing stable and transferable sparse expert models. *arXiv preprint arXiv:2202.08906*, 2022.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We clearly state the proposed claims and motivations in the abstract and introduction, and list our contributions at the end of the introduction.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We provide a detailed discussion of the limitations of this work in Appendix Section: Limitations, including the potential increase in model scale with training.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: In Section: Methodology of the paper, we provide the full set of assumptions and a complete (and correct) proof for each theoretical result, such as the reasons why orthogonality constraints cause forgetting, the derivation of orthogonal gradients, and the derivation of the label-guided localized balancing strategy.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: In the Experimental Setup section of Section: Experiments, we present all experimental details for reproducibility, including the datasets used, the backbone network, training details, hyperparameters, and the unique task settings and task sequences specific to continual learning. In Section: Methodology, we provide a detailed description of all aspects of our method and the loss functions. Furthermore, we will make all training code and training details publicly available.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We will release all our training code, inference code, experimental settings, and trained models for public availability. The datasets used in the paper may require additional application and access, such as FaceForensics++. We will provide the official link to these datasets, as well as our preprocessing code for handling them, to ensure full reproducibility of the results.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: In the Experimental Setup section of Section: Experiments, we present all experimental details for reproducibility, including the datasets used, the backbone network, training details, hyperparameters, and the unique task settings and task sequences specific to continual learning.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: The experiments are based on two metrics: accuracy and AUC. First, due to our limited experimental resources, conducting repetitive experiments is impractical. Second, in the field of Face Forgery Detection, error bars or statistical significance tests are not required. None of the compared Face Forgery Detection methods provide statistical significance tests or error bars in their papers.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: In the Experimental Setup section of Section: Experiments, we provide the computational resources required for the experiments, and in the appendix, we present information related to computational resource consumption, including the total number of model parameters, the number of backbone parameters, and the trainable parameter count.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: In the paper, we carefully adhere to the NeurIPS Code of Ethics in every respect.

   Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Undoubtedly, the work presented in this paper has positive societal impacts, such as preventing fraud and identifying fake information, which we discuss in detail in the introduction.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper does not release any pretrained language models, image generators, or scraped datasets that could potentially be misused, thereby ensuring that there are no such risks associated with the paper.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We have provided citations and application conditions for the datasets and methods used.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We do not propose any new dataset-related assets, and the trained models as well as all code will be open-sourced.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: There are no crowdsourcing experiments or research involving human subjects in the paper.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.

- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: There are no crowdsourcing experiments or research involving human subjects in the paper.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: The core method in the paper does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (https://neurips.cc/Conferences/2025/LLM) for what should or should not be described.

Appendix

A Limitations

First, the first limitation lies in the linear increase of our method's model scale as the training sequence progresses. Although the number of parameters in LoRA is very small, this growth may have an impact on extremely long sequence continual learning. Second, our method fine-tunes the pre-trained weights within subspaces using LoRA, and thus may rely on a strong general pre-trained model that already possesses some general knowledge before training. We will address these limitations in our future research.

B Details of Orthogonal Gradient

To estimate the gradient space of LoRA and orthogonalize it with the subspace of LoRA, we rely on a key proposition [29] that the gradients of linear layers will lie within the input space. Here, we provide proof of this proposition. For a linear layer, let its weight matrix be denoted as $\mathbf{W} \in \mathbb{R}^{d_I \times d_O}$. Its forward process can be expressed as:

$$\mathbf{e} = \mathbf{W}\mathbf{x} + \mathbf{b}, \tag{16}$$

where $\mathbf{x} \in \mathbb{R}^{d_I}$ and $\mathbf{e} \in \mathbb{R}^{d_O}$ are the input and output vector, respectively. By applying the chain rule, we compute the gradient of $\mathbf{W}$:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{W}} = \frac{\partial \mathcal{L}}{\partial \mathbf{e}} \frac{\partial \mathbf{e}}{\mathbf{W}} = \frac{\partial \mathcal{L}}{\partial \mathbf{e}} \mathbf{x}^T. \tag{17}$$

Assuming the vector $\frac{\partial \mathcal{L}}{\partial \mathbf{e}}$ is given by: $[l_1, l_2, ..., l_{d_O}]^T$, then the gradient $\frac{\partial \mathcal{L}}{\partial \mathbf{e}} \mathbf{x}^T$ can be represented as:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{e}} \mathbf{x}^T = \begin{bmatrix} l_1, \\ l_2, \\ ..., \\ l_{d_O} \end{bmatrix} \mathbf{x}^T = \begin{bmatrix} l_1 \mathbf{x}^T, \\ l_2 \mathbf{x}^T, \\ ..., \\ l_{d_O} \mathbf{x}^T \end{bmatrix}. \tag{18}$$

It can be observed that each row of the gradient $\frac{\partial \mathcal{L}}{\partial \mathbf{W}}$ is a multiplication of the input vector $\mathbf{x}$ with a certain value l_k $(1 \leq k \leq d_O)$. Hence, the gradient shares the same subspace as the input vector $\mathbf{x}$, meaning the gradient lies within the input space.

C More Experimental Details

C.1 Experimental Fairness

First, at the data level, we use the same amount of subtask data as the compared methods, i.e., 100 videos with 20 sampled frames per video, which is identical to methods such as DFIL [40] and SUR-LID [7]. Second, at the model level, we employ a backbone with a similar order of magnitude of parameters as the compared methods: ViT-B/16, thus avoiding the use of a larger model to ensure experimental fairness. It is worth noting that our trainable parameters consist only of the parameters of two LoRAs, which is approximately 1/10 of the trainable parameters of the compared methods. Third, in terms of training strategies, many methods such as DFIL [40] and SUR-LID [7] utilize a Replay Set, which allows access to data from previous tasks—a practice that is prohibited in many continual learning scenarios. The proposed DevFD method completely isolates any data between subtasks. This not only demonstrates that we achieve the best results under stricter constraints but also indicates the broader applicability of our method. Finally, the comparative metric data used are the best results extracted directly from the original papers. We achieve an undeniably leading position in absolute performance.

C.2 More Implementation Details

We provide our detailed hyperparameter settings here. For the label-guided localized balancing strategy, we use a fixed hyperparameter setting. We set δ to 0.15 and λ_3 to 0.2. For the integrated

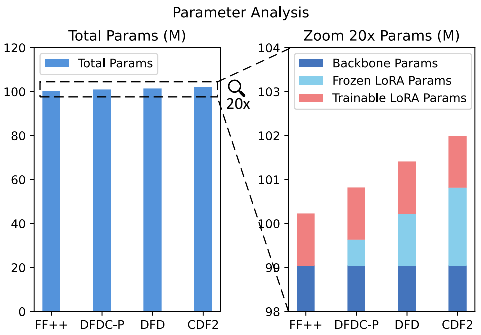


Figure 5: Expanded parameter analysis. The parameter expansion of the proposed developmental MoE is fully controllable, and the trainable parameters will remain constant throughout the task sequence.

orthogonality loss, since our goal is to prevent the gradients in the current LoRA from interfering with the subspaces established by previous tasks during the entire training phase, we dynamically adjust the hyperparameters of the orthogonal subspace loss to the orthogonal gradient loss for each epoch. Specifically, from epoch [0,5), we set λ_1 to 0.5 and λ_2 to 0.5. From epoch [5,10), we set λ_1 to 1 and λ_2 to 0.1. From epoch [10,20), we set λ_1 to 1 and λ_2 to 0.01.

C.3 More Details of Task10 Long Sequence Experiment

Existing continual learning methods for forgery detection typically employ task sequences with only four subtasks. We provide results for a continual learning task sequence with 10 subtasks based on DF40 [66] and reproduce three high-performing continual learning methods for comparison: CoReD [19], DFIL [40], and HDP [55]. We start with the original task sequence [Hybrid, FR, FS, EFS], where Face-Swapping (FS): BlendFace, Face-Reenactment (FR): MCNet, and Entire Face Synthesis (EFS): StyleGAN3 are used as our initial learning sequence and expand it to 10 tasks. We carefully select the remaining six methods while maintaining diversity. Specifically, for Face-Swapping (FS), we choose SimSwap and InSwapper; for Face-Reenactment (FR), we select Wav2Lip and SadTalker; and for Entire Face Synthesis (EFS), we choose StyleGAN2 and StyleGAN-XL. We then randomly shuffle the last six methods to form the complete task sequence with 10 subtasks: [FFpp, MCNet, BlendFace, StyleGAN3, StyleGAN-XL, Wav2Lip, SadTalker, StyleGAN2, SimSwap, InSwapper]. Analysis reveals that EFS methods may have significant differences from FS and FR methods, leading to sharp increases in forgetting rates during transitions between learning EFS methods and FS & FR methods, such as 5->6 and 7->8.

D More Experimental Results

D.1 Parameter Analysis

We introduce a Developmental Mixture of Experts: DevFD to tackle the challenge of detecting new types of face forgeries. As new tasks are integrated, the parameters of the DevFD model expand in a specific manner. To illustrate that our developmental MoE can handle long-sequence tasks without suffering from parameter explosion, we perform experiments to track the expansion in parameter volume, as depicted in Fig. 5. By employing Parameter-Efficient Fine-Tuning (PEFT) technology, our DevFD model can learn in long-sequence tasks with a minimal increase in parameter volume. The experimental results show that when the model is fine-tuned for the first task in the sequence, the trainable parameters constitute only 1.18% of the total parameters. Even after training on all four tasks, the additional parameters account for just 2.89% of the total. Furthermore, since only the Real-LoRA and the final LoRA in the Fake-LoRA sequence are trainable, the number of trainable parameters remains stable throughout the task sequence and does not expand with the overall parameter volume. Consequently, the parameters of our proposed developmental MoE are fully manageable, making it well-suited for learning in long-sequence tasks.

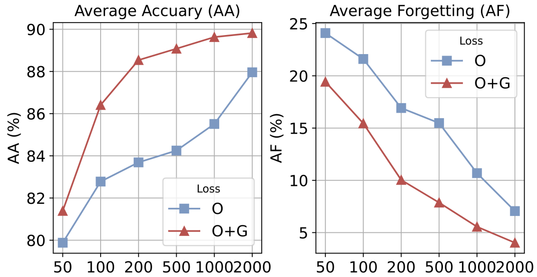


Figure 6: Effect of subspace and gradient constraint. O represents the scenario with only subspace constraints, while O+G indicates the presence of both subspace and gradient constraints. The gradient constraints improve in both accuracy and forgetting rate.

D.2 Effect of orthogonal Subspace and Orthogonal Gradients

The proposed method introduces additional gradient constraints to ensure that even when the subspaces are not entirely orthogonal, the gradient space remains orthogonal to the established subspaces. This prevents the gradient from interfering with the established subspaces. We simulate the early stages of the training process using fewer training samples to evaluate whether these constraints on the gradient space can prevent interference and the results shown in Fig. 6. Here, O represents the scenario with only subspace constraints, while O+G indicates the presence of both subspace and gradient space constraints. The experimental results demonstrate that the orthogonal gradients significantly reduces the average forgetting in the early stages of training. Additionally, it can improve the average accuracy by constraining the early learning direction, confirming the effectiveness of orthogonal gradients.

D.3 Effect of ViT Backbone

We employ a Vision Transformer (ViT) pre-trained by CLIP [43] as our initial backbone for training. To explore how different backbones affect model performance, we test various ViT models [16] with varying sizes of parameters. The results of these experiments are detailed in Table 4.

Table 4: Effect of backbones in the proposed method.Shadowed lines indicate the backbone used in our method.

Backbone	Params	FF++	DFDC-P		DFD		CDF2	
		AA	AA	AF	AA	AF	AA	AF
ViT-B/16	86M	98.41	93.48	1.35	93.14	3.40	89.82	4.03
ViT-B/32	86M	98.03	91.24	2.88	93.29	4.52	88.25	7.63
ViT-L/14	304M	99.07	93.75	1.36	93.37	3.79	90.55	5.86

Our experiments reveal that a patch size of 16x16 performs better than 32x32, with the ViT-B/16 model demonstrating superior overall performance compared to the ViT-B/32 model. When considering the effect of increasing the parameter scale, the ViT-L/14 model, which adds 218M parameters to the

Table 5: Performance comparison of different task orders.

Dataset	FF++	DFDC-P	CDF2	DFD	AA	AF
FF++	98.41	-	-	-	98.41	-
DFDC-P	96.95	90.88	-	-	93.92	1.46
CDF2	93.42	88.37	92.33	-	91.37	3.75
DFD	90.86	84.35	89.81	95.91	90.23	4.15
Dataset	FF++	DFD	DFDC-P	CDF2	AA	AF
FF++	98.41	-	-	-	98.41	-
DFD	94.13	97.25	-	-	95.69	4.28
DFDC-P	92.88	94.91	85.94	-	91.24	3.94
CDF2	90.05	94.39	87.28	90.43	90.54	3.29

ViT-B/16, only saw a modest improvement in average accuracy but a slight increase in the average forgetting rate when trained on all four tasks. We attribute this marginal gain to the limited dataset size, which makes model performance less sensitive to parameter scale increase. Based on these findings, we choose the ViT-B/16 as our backbone model for its optimal balance of performance and efficiency.

D.4 Effect of Task Orders

To determine if the high accuracy and resistance to forgetting in our proposed method are independent of the specific order in which tasks are presented, we conducted a series of experiments varying the task sequences. The outcomes of these experiments are detailed in Table 5. Our findings indicate that our method consistently delivers high detection accuracy and resistance to forgetting across different task sequences. For instance, when the tasks are sequenced as [FF++, DFDC-P, CDF2, DFD], the model achieves an average accuracy of 90.23% and an average forgetting rate of 4.15%. In another sequence, [FF++, DFD, DFDC-P, CDF2], the model's average accuracy is 90.54% with an average forgetting rate of 3.29%. These results suggest that the order of tasks does not significantly impact the model's performance.