

Suppressing Gradient Conflict for Generalizable Deepfake Detection

Ming-Hui Liu, Harry Cheng, Xin Luo, Xin-Shun Xu* *Senior Member, IEEE*

arXiv:2507.21530v1 [cs.CV] 29 Jul 2025

Abstract—Robust deepfake detection models must be capable of generalizing to ever-evolving manipulation techniques beyond training data. A promising strategy is to augment the training data with online synthesized fake images containing broadly generalizable artifacts. However, in the context of deepfake detection, it is surprising that jointly training on both original and online synthesized forgeries may result in degraded performance. This contradicts the common belief that incorporating more source-domain data should enhance detection accuracy. Through empirical analysis, we trace this degradation to gradient conflicts during backpropagation which force a trade-off between source domain accuracy and target domain generalization. To overcome this issue, we propose a Conflict-Suppressed Deepfake Detection (CS-DFD) framework that explicitly mitigates the gradient conflict via two synergistic modules. First, an Update Vector Search (UVS) module searches for an alternative update vector near the initial gradient vector to reconcile the disparities of the original and online synthesized forgeries. By further transforming the search process into an extremum optimization problem, UVS yields the uniquely update vector, which maximizes the simultaneous loss reductions for each data type. Second, a Conflict Gradient Reduction (CGR) module enforces a low-conflict feature embedding space through a novel Conflict Descent Loss. This loss penalizes misaligned gradient directions and guides the learning of representations with aligned, non-conflicting gradients. The synergy of UVS and CGR alleviates gradient interference in both parameter optimization and representation learning. Experiments on multiple deepfake benchmarks demonstrate that CS-DFD achieves state-of-the-art performance in both in-domain detection accuracy and cross-domain generalization.

Index Terms—Deepfake Detection, Conflicting Gradients, Generalization.

I. INTRODUCTION

WITH the rapid development of multimodal large-scale models [1]–[5], the technical barriers to tampering or synthesizing facial data have been significantly lowered [6]–[9]. The proliferation of such deepfake content continues to disrupt the order of internet finance, social discourse, and ethical norms. Therefore, deepfake detection, which is dedicated to distinguishing authentic data from fake content, has gained attention from the research community [10]–[17]. As illustrated in Fig. 1(a), traditional deepfake detectors are trained with a fixed set of real and fake data to perform a binary classification [18]–[20]. While this paradigm delivers strong

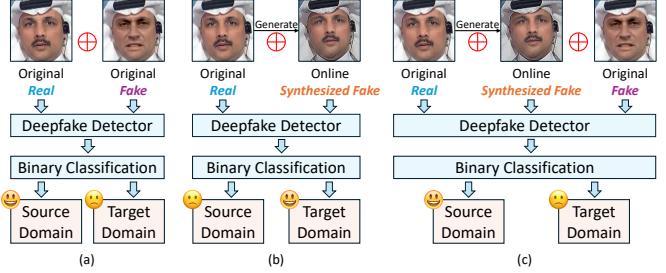


Fig. 1. Different training strategies. (a) Traditional training strategy, which suffers from poor generalization. (b) Replacing original fake images with online synthesized ones, which lacks the ability to detect manipulations in the source domain. (c) Training with both online synthesized and original fake images, which struggles to generalize effectively to the target domain.

performance under in-domain evaluation (where training and testing set share the same distribution), it often falters in cross-dataset settings, exhibiting markedly poorer generalization to unseen target domains [21]–[24].

To enhance poor cross-domain generalization performance, a prominent approach is to replace original fake samples with online synthesized fake data generated from real images [25]. These generated samples explicitly incorporate broadly generalizable artifacts rather than dataset-specific cues [26]. As illustrated in Fig. 1(b), these newly synthesized fake samples are generated online [27], and replace the original fake data in the training process. By disentangling the model from the artifacts of a fixed fake set, this strategy effectively improves the generalization ability of models in unseen domains.

Despite its benefits for cross-domain generalization, this training paradigm, *i.e.*, replacing original fake examples with online synthesized fakes alongside real images, induces an disconcerting side effect: a significant degradation in source domain detection performance. Specifically, detectors trained in this way often fail to identify the forgeries contained in the original dataset. An intuitive remedy is to reintroduce the displaced original fake data. Nevertheless, as illustrated in Fig. 1(c), doing so undermines the generalization ability in target domains. This issue, also known as the ‘ $1+1 < 2$ ’ problem, reflects a counterintuitive phenomenon: More and richer source domain data do not necessarily lead to better detection performance, even though the online synthesized samples are generally considered a beneficial augmentation. Several recent studies try to mitigate this challenge via superficial progressive training schemes [10]. By contrast, our method takes a fundamentally different tack: We first investigate why merging these data sources can paradoxically impair performance, *i.e.*, what underlying factors drive the ‘ $1 + 1 < 2$ ’ phenomenon?

*Corresponding author.

Ming-Hui Liu and Xin Luo are with the School of Software, Shandong University, Jinan 250101, China (e-mail: liuminghui@mail.sdu.edu.cn; luoxin.lxin@gmail.com). Xin-Shun Xu is with the School of Software, Shandong University, Jinan 250100, China, and also with the Quan Cheng Laboratory, Jinan 28666, China (e-mail: xuxinshun@sdu.edu.cn). Harry Cheng is with the School of Computer Science, National University of Singapore, Singapore (e-mail: xaCheng1996@gmail.com).

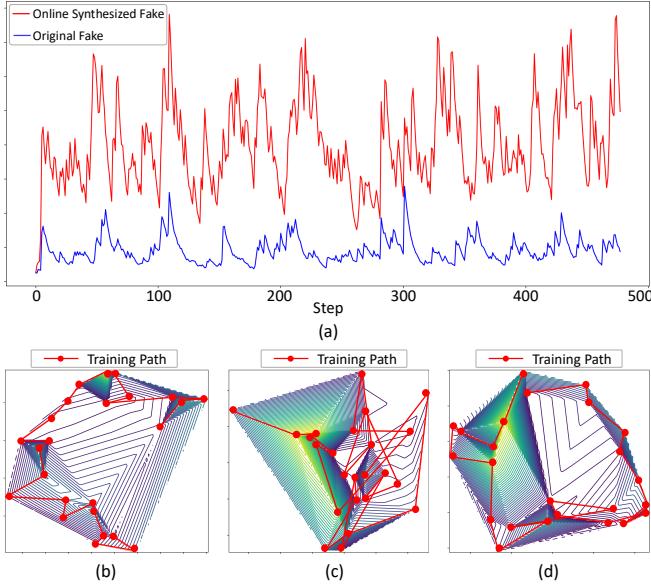


Fig. 2. Gradient conflicts when trained on different types of data. (a) When trained with two types of data, the model’s loss curve shows a fluctuating pattern with opposing peaks and valleys, indicating adversarial trends in optimization. (b) When trained on a single subset, the model converges smoothly. (c) However, when trained on both subsets simultaneously, the training becomes highly unstable, with significant oscillations. (d) Our CS-DFD method resolves the gradient conflict, presenting a smooth trend.

Our approach is motivated by intuitive comparisons of detector behavior on online synthesized versus original forgery data. To this end, we split the training set into two complementary subsets: **[real data, original fake data]** and **[real data, online synthesized fake data]**. During training, the detector concurrently processes both data subsets and computes the loss for each subset independently. We then track the optimization trajectories and Fig. 2(a) plots the classification losses for both training subsets over identical training iterations. From this figure, we identify a pronounced mutual suppression phenomenon: When one subset attains a lower loss value (located in the trough of the loss curve), the opposing subset exhibits a significantly higher loss (located in the peak of the loss curve). This observation implies that the model adopts contradictory learning logics to detect original forgery data and online synthesized forgery data, thereby deriving ***conflicting gradient directions***. To further validate this hypothesis, we visualize the loss landscape trajectories. As illustrated in Fig. 2(b), when trained on a single subset, the optimizer follows a smooth descent into a well-defined local minimum. However, as shown in Fig. 2(c), when joint training on both subsets, the model becomes trapped in a saddle region near a high curvature gradient basin, exhibiting oscillatory behavior and failing to converge to the global optimum. The above divergence in convergence behavior underscores how the gradient conflicts of different subsets impede effective joint training.

Building on our empirical observations, we pinpoint ***gradient conflicts between heterogeneous forgery data*** as a fundamental barrier for addressing the ‘ $1+1 < 2$ ’ issue. To achieve both high detection accuracy on the source domain and strong generalization to target domains, these conflicts must be ex-

plicitly resolved. We thus propose Conflict-Suppressed Deepfake Detection (CS-DFD), a unified framework that harnesses discriminative cues from both original and online synthesized forgeries while ensuring stable convergence toward the global optimum. CS-DFD comprises two complementary modules: i) *Update Vector Search (UVS)*. Instead of applying the total gradient which oscillates between competing objectives, UVS computes the individual gradients and loss descent rates for both original subset and online synthesized subset. Then, it searches within the neighborhood of the total gradient to find a conflict-free update vector that maximizes the descent on both losses. ii) *Conflict Gradient Reduction (CGR)*. Recognizing that adjusting update vectors alone cannot fully eliminate gradient conflict, CGR introduces a conflict descent loss that quantifies and penalizes conflicting gradients in the embedding space. Thus, it i) encourages the network to learn a diverse set of discriminative features in shallow layers and ii) gradually maps these features into a low-conflict embedding space at deeper layers. UVS and CGR allow CS-DFD to aggregate complementary features from all available forgery data without suffering the ‘ $1+1 < 2$ ’ degradation, thereby delivering both robust in-domain accuracy and superior cross-domain generalization. As shown in Fig. 2(d), the optimization trajectory becomes noticeably smoother after applying our method. Our contributions are summarized as three-fold:

- To the best of our knowledge, we are the first to identify the gradient conflicts inherent in heterogeneous forgery data and demonstrate that this issue induces a performance trade-off between the source and target domains.
- To address the gradient conflicts, we propose a novel Conflict-Suppressed Deepfake Detection (CS-DFD) framework which integrates two cooperative modules: The UVS module replaces the original gradient vectors using a conflict-free update vector, and the CGR module quantifies gradient conflict intensity and leverages this metric to learn a low-conflict embedding space.
- We conduct comprehensive experiments on multiple deepfake datasets. And the results demonstrate that our method significantly outperforms several state-of-the-art baselines in both source and target domains. Moreover, our approach can seamlessly be integrated into different backbone architectures, exhibiting excellent versatility.

II. RELATED WORK

A. Deepfake Generation and Detection

Generation. Benefiting from the continuous development of human portrait synthesis technologies, deepfake [28] has increasingly become a prominent concern for communities [29]. Early approaches in this field employ autoencoders [30] or generative adversarial networks (GANs) [31] to synthesize human likenesses [32]. For instance, StyleGAN [33] modifies high-level facial attributes with a progressive growing approach. IPGAN [34] disentangles the identity and attributes of the source and target faces, and then combines them to synthesize fake content. Moreover, identity-relevant features [35] have been introduced into deepfake generation [6].

In recent years, diffusion models [36]–[38] have garnered attention for their ability to sample high-quality images by first adding noise to the original image and then progressively denoising it [39], [40]. This adaptation of diffusion models into a foundation for deepfake synthesis has attracted significant interest. For instance, DiffSwap [7] frames face swapping as a conditional inpainting problem by masking the target face and guiding the diffusion process with identity and landmark conditions. DiffFace [41] introduces an ID-conditional diffusion model and uses off-the-shelf facial expert models and a target-preserving blending strategy to transfer identities. DCFace [4] presents a dual condition generator conditioned on both subject identity and external style factors to synthesize faces.

Detection. Deepfake detection [11], [12], [42]–[44] is generally cast as a binary classification task. Preliminary efforts often attempt to detect specific manipulation traces [45]–[47], which have shown certain improvements in the intra-data set setting. However, these methods often encounter inferior performance when applied to data with different distributions or manipulation methods. To address this generalization issue [14], researchers have conducted investigations from multiple perspectives, such as more effective model architectures [19], [48], [49], richer modal information [50]–[52], and more efficient data augmentation strategies [22]–[24], [53]. In recent years, online data synthesis methods have garnered significant attention [25], [54]. These methods highlight that fake samples in existing datasets often contain method-specific artifacts, which can lead to severe overfitting issues [13], [55], [56]. Therefore, these samples need to be replaced with more generalized fake images. For instance, Chen *et al.* [54] propose a facial region blending strategy that generates fake samples with specified forged regions, blending types, and blending ratios. SBI [25] introduces a self-blending strategy, which mixes different real facial images during training to capture boundary-fusion features. These methods typically replace the original source-domain fake data with the online-synthesized fake images. However, they also introduce the ‘1 + 1 < 2’ issue, where the detector is forced to trade off between detection performance in the source domain and generalization performance in the target domain.

B. Gradient Conflict

Historically, gradient conflict [57] issues have primarily been observed in multi-task learning and multi-objective optimization scenarios. It arises because models must simultaneously learn multiple unrelated tasks or optimize conflicting objectives. When different tasks or objectives require divergent—or even opposing—parameter update directions (*i.e.*, the gradient vectors), significant gradient conflicts emerge. During training, conflicting gradients destabilize parameter updates, causing oscillatory behavior that impedes convergence or traps models in suboptimal solutions. Consequently, multi-task learning often results in performance degradation compared to individually training each task. To address this, PCGrad [58] indicates gradient conflict as the primary obstacle in multi-task learning and resolves it by projecting conflicting gradients onto their normal planes, ensuring stable convergence.

GAC [59] identifies ascending directions for each gradient to prevent any single gradient from dominating optimization or conflicting with others. SAGM [60] minimizes empirical risk, perturbation loss, and their divergence simultaneously by maximizing gradient inner products. NASViT [61] observes that gradients from different sub-networks conflict with those of the supernet and combine a gradient projection algorithm, a switchable layer scaling design, and a regularization training recipe to alleviate the conflict issue. However, these studies are confined to addressing gradient conflicts in traditional multi-task learning scenarios, failing to consider that such conflicts may also arise between different data within the same task. Concurrently, we are the first to recognize that gradient conflicts between different types of fake data in deepfake detection tasks severely impede the optimization process, thereby preventing existing methods from improving generalization by enhancing the diversity of fake data.

III. PROBLEM FORMULATION

The deepfake detection task can be formulated as a binary classification problem, optimized with the cross-entropy loss:

$$\mathcal{L}_{\text{bce}} = -\frac{1}{N} \sum_{i=1}^N y_i \log p_i + (1 - y_i) \log (1 - p_i), \quad (1)$$

where x_i denote the i -th input sample in the training set $X = [x_i]_{i=1}^N$, N is the total number of samples. y_i and p_i are the ground-truth label and the predicted output for x_i , respectively. X serves as the source domain dataset, which consists of two subsets: real data x_r and fake data x_f . This training process achieves satisfactory detection performance on the source domain, while it has been widely recognized to induce overfitting, thereby posing generalization challenges on the target domain [21], *e.g.*, other deepfake datasets. To address this issue, a common approach is to replace original fake images x_f with online synthesized ones x_s [27]. Specifically, the training input $X = [x_r, x_f]$ is substituted with $X' = [x_r, x_s]$. The motivation behind this replacement is to exploit prior knowledge to generate challenging samples that embody more generalizable artifacts, such as blending boundaries, thereby improving the detection ability across diverse domains. And this operation can be directly integrated into the training process described in Equation (1).

However, due to the incompleteness of the training data (absence of original forgery data x_f), models trained with X' inevitably exhibit suboptimal performance on the source domain. To mitigate this limitation, a straightforward approach is to reintroduce x_f as part of the training data to construct a new source domain training set,

$$X'' = [x_r, x_s, x_f], \quad (2)$$

where both x_f and x_s are the forgery data. While this strategy can partially restore performance on the source domain, it tends to compromise the generalization ability [10]. Therefore, many studies implicitly assume an insurmountable inherent trade-off between source domain and target domain accuracy.

In this study, we aim to address the long-standing ‘1+1 < 2’ problem from a novel perspective. Specifically, we reformulate

the loss function in Equation (1) into a dual-stream formulation, where the model simultaneously receives two types of input: $X = [x_r, x_f]$, i.e., real data and original forgeries, and $X' = [x_r, x_s]$, i.e., real data and online synthesized forgeries. For each input stream, we compute its loss function separately, and the total classification loss can be redefined as:

$$\begin{aligned}\mathcal{L}_{\text{bce}} &= \mathcal{L}_1 + \mathcal{L}_2 \\ &= -\sum_{j=1}^2 \left(\frac{1}{N_j} \sum_{i=1}^{N_j} y_i \log p_i + (1 - y_i) \log (1 - p_i) \right),\end{aligned}\quad (3)$$

where \mathcal{L}_1 and \mathcal{L}_2 correspond to losses derived from different subsets. j is the set index. Specifically, \mathcal{L}_1 is computed based on set X and is considered effective for achieving strong performance on the source domain, while \mathcal{L}_2 is computed based on set X' and can significantly enhance the generalization ability on target domains. Equation (3) implies that, to achieve strong performance on both the source and target domains, \mathcal{L}_1 and \mathcal{L}_2 must be optimized cooperatively. In other words, at each training step t , the gradient vector g_0 is expected to drive the simultaneous minimization of both loss functions:

$$\begin{cases} \mathcal{L}_{1,\theta^{t+1}} < \mathcal{L}_{1,\theta^t}, & \text{s.t. } \theta^{t+1} = \theta^t - \alpha g_0, \\ \mathcal{L}_{2,\theta^{t+1}} < \mathcal{L}_{2,\theta^t}, \end{cases}\quad (4)$$

where α is a sufficiently small learning rate. θ^t denotes model parameters at time step t . However, as illustrated in Fig. 2, gradient conflicts arise between the original forgery data and the online synthesized forgery data. These conflicts hinder effective cooperative optimization of the model, thereby giving rise to the aforementioned trade-off problem. That is, minimizing one loss will inevitably amplify the other:

$$\mathcal{L}_{1,\theta^{t+1}} > \mathcal{L}_{1,\theta^t}, \text{ s.t. } \begin{cases} \theta^{t+1} = \theta^t - \alpha g_0, \\ \mathcal{L}_{2,\theta^{t+1}} > \mathcal{L}_{2,\theta^t}. \end{cases}\quad (5)$$

To alleviate this challenge, we propose Conflict-Suppressed Deepfake Detection (CS-DFD) framework, which resolves such gradient conflicts to simultaneously enhance performance on both the source and target domains.

IV. METHODOLOGIES

The primary objective of our CS-DFD approach is to alleviate gradient conflicts between heterogeneous forgery data, thereby synchronously enhancing the detection accuracy on both source and target domains. As illustrated in Fig. 3, our framework comprises two core modules: 1) the Update Vector Search (UVS) module, which computes a conflict-free update vector to replace the original gradient vector; 2) the Conflict Gradient Reduction (CGR) module, which quantifies conflict intensity using gradient vectors and leverages this measure to guide the model in learning a low-conflict embedding space. Both modules are designed to ensure the cooperative optimization of loss functions \mathcal{L}_1 and \mathcal{L}_2 , thereby effectively reducing gradient conflict while ensuring that the capability of learning diverse knowledge from heterogeneous forgery data.

A. Update Vector Search Module

To simultaneously utilize both original forgeries and online synthesized forgeries while minimizing their corresponding losses \mathcal{L}_1 and \mathcal{L}_2 , the UVS module searches for a conflict-free parameter update vector during the backpropagation stage.

Usually, during the optimization process of a vanilla deepfake detector, its parameters are updated based on the gradient vector g_0 of the loss function \mathcal{L}_{bce} . Thus, in each optimization iteration, the descent rate Δ_{bce} of \mathcal{L}_{bce} is given by:

$$\begin{aligned}\Delta_{\text{bce}} &= \frac{1}{\alpha} (\mathcal{L}_{\text{bce},\theta^t} - \mathcal{L}_{\text{bce},\theta^{t+1}}) \\ &= \frac{1}{\alpha} (\mathcal{L}_{\text{bce},\theta^t} - \mathcal{L}_{\text{bce},\theta^t - \alpha g_0}),\end{aligned}\quad (6)$$

where α is the learning rate; θ denotes the parameters.

However, as analyzed in previous Equation (5), loss components (\mathcal{L}_1 and \mathcal{L}_2) included in the total loss (\mathcal{L}_{bce}) tend to counteract each other due to gradient conflicts. As a result, the descent of the total loss cannot guarantee the reduction in each individual loss component. When the overall gradient g_0 satisfies the descent requirement of one component, it may lead to an increase in the other.

To address this optimization dilemma, we try to search for a low-conflict update vector v to replace the original gradient vector g_0 during the backpropagation process, thereby concurrently maximizing the reduction of both \mathcal{L}_1 and \mathcal{L}_2 :

$$\max_{j \in [1,2]} \Delta_j = \max_{v \in \mathbb{R}} \frac{1}{\alpha} (\mathcal{L}_{j,\theta} - \mathcal{L}_{j,\theta - \alpha v}).\quad (7)$$

Considering that directly solving for a substitutional update vector v is non-trivial, we apply a first-order Taylor expansion to Equation (7) and obtain the following simplified form:

$$\begin{aligned}\max_{j \in [1,2]} \Delta_j &\approx \max_{v \in \mathbb{R}} \frac{1}{\alpha} (\mathcal{L}_{j,\theta} - \mathcal{L}_{j,\theta} + \alpha g_j v) \\ &= \max g_j v,\end{aligned}\quad (8)$$

where g_j denotes the gradient vector of the loss sub-term \mathcal{L}_j .

In the context of resolving data conflicts, to ensure that both sub-losses decrease in a coordinated manner, it is sufficient to maximize the minimum descent rate of the two sub-losses:

$$\max_{v \in \mathbb{R}, j \in [1,2]} \min \Delta_j \approx \max_{v \in \mathbb{R}, j \in [1,2]} g_j v, \text{ s.t. } \|v - g_0\|^2 \leq c^2 \|g_0\|^2,\quad (9)$$

where the target update vector v lies within the neighborhood $c\|g_0\|$ of the original gradient vector g_0 to ensure stability. In this way, the complex **set-based** problem has been transformed into a simpler **extremum-based** problem.

To further construct a solvable constrained linear optimization problem, we introduce an auxiliary variable k to explicitly represent the lower bound of $\min \Delta_j$, and the original problem can be reformulated as solve for the maximum value of k :

$$\max_{v \in \mathbb{R}, j \in [1,2]} k, \text{ s.t. } \begin{cases} g_j v \geq k, \\ \|v - g_0\|^2 \leq c^2 \|g_0\|^2. \end{cases}\quad (10)$$

To solve Equation (10), we construct its Lagrangian function to integrate Equation (10) with its constraint conditions according to the construction principles:

$$\psi_{v,k,\lambda,\mu} = k - \lambda(\|v - g_0\|^2 - c^2 \|g_0\|^2) - \sum_{j=1}^2 \mu_j(k - g_j v).\quad (11)$$

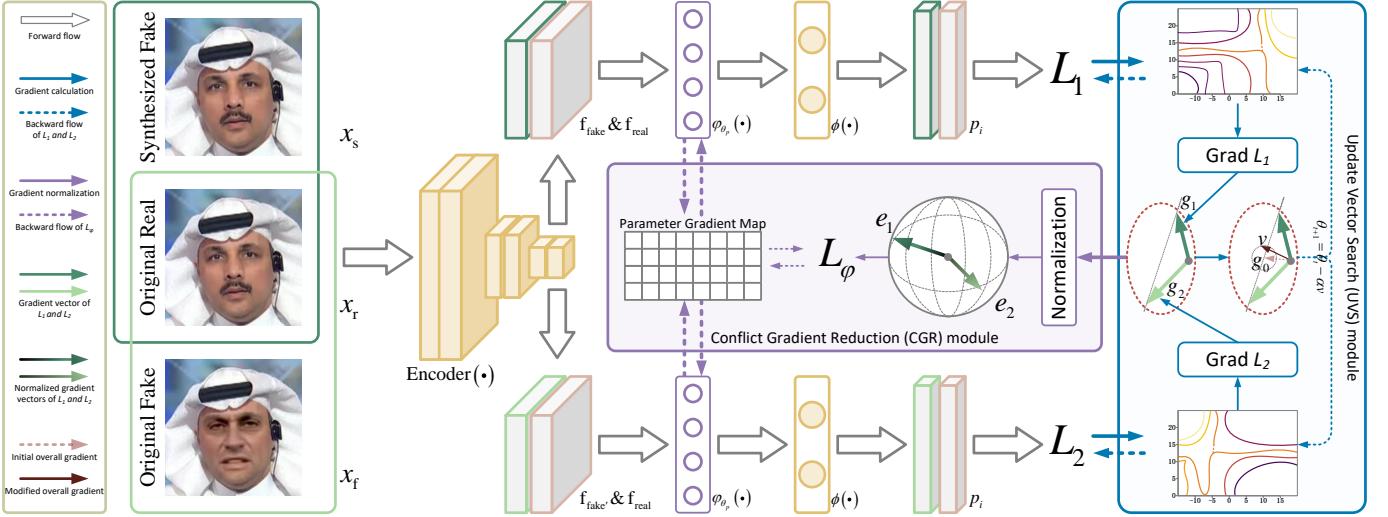


Fig. 3. Overall architecture of the CS-DFD framework, which consists of two modules: Update Vector Search (UVS) and Conflict Gradient Reduction (CGR). After computing the gradient vectors g_1 and g_2 corresponding to two sets of data, UVS adjusts the overall gradient vector g_0 to obtain an approximate gradient vector v that satisfies the descent requirements of loss functions \mathcal{L}_1 and \mathcal{L}_2 ; CGR measures the current level of conflict by g_1 and g_2 , and leverages conflict descent loss $\mathcal{L}_\varphi(\cdot)$ to project the features into a low-conflict embedding space.

To ensure the existence of the maximum value of k , we impose $\sum_{j=1}^2 \mu_j = 1$. As a result, the influence of the variable k can be eliminated, and the Lagrangian function simplifies to:

$$\psi_{v,\lambda,\mu} = \sum_{j=1}^2 \mu_j g_j v - \lambda(\|v - g_0\|^2 - c^2 \|g_0\|^2). \quad (12)$$

At this point, primal problem is transformed into an optimization problem of the Lagrangian function by maximizing the variable v and minimizing the variables λ and μ (*i.e.*, $\max_v \min_{\lambda,\mu} \psi$). Meanwhile, according to the strong duality principle, the optimal solution can be further approximated by minimizing the maximum of the Lagrangian function, *i.e.*:

$$\min_{\lambda,\mu} \max_v \psi = \min_{\lambda,\mu} \max_v g_w v - \lambda(\|v - g_0\|^2 - c^2 \|g_0\|^2), \quad (13)$$

where $g_w = \sum_{j=1}^2 \mu_j g_j$. Subsequently, we analytically derive and solve for the stationary points with respect to the variables v and λ , yielding the relevant extremum values v^* and λ^* :

$$\begin{cases} v^* = g_0 + g_w / 2\lambda, \\ \lambda^* = \|g_w\| / (2c\|g_0\|). \end{cases} \quad (14)$$

By substituting the obtained v^* and λ^* back, we derive the final dual objective, which involves only the variable μ :

$$\min_\mu \psi = \min_\mu g_w^T g_0 + c\|g_0\| \|g_w\|. \quad (15)$$

Finally, we use the projected gradient method to minimize the final dual objective ψ_μ and obtain the optimal μ^* . The optimization process is as follows:

$$\mu^{t+1} \leftarrow \mu^t - \beta \cdot \nabla_{\mu^t} \psi. \quad (16)$$

After the above derivations, we uniquely determine the desired update vector v^* and maximize the descent rate Δ_j of the subdominant sub-term. By seamlessly replacing the

original gradient vector g_0 with a low-conflict update vector v^* during backpropagation, the model can effectively maintain its detection performance in both the source and target domains.

B. Conflict Gradient Reduction Module

While the low-conflict update vector v^* can simultaneously satisfy the descent requirements of both \mathcal{L}_1 and \mathcal{L}_2 , it may increase the risk of losing specific discriminative knowledge associated with heterogeneous source forgery types. To preserve both common and specific knowledge from original and online synthesized forgeries, we insert a learnable feature projection layer φ_{θ_p} (with its parameters denoted as θ_p) before the classifier $\phi(\cdot)$. In this way, features can be mapped into a low-conflict embedding space without affecting the preceding network. The projection operation is defined as:

$$\hat{\mathbf{f}} = \varphi_{\theta_p}(\mathbf{f}), \quad (17)$$

where \mathbf{f} is the initial feature vector. $\hat{\mathbf{f}}$ is the output embedding that resides in a low-conflict space. Projection layer φ_{θ_p} is composed of several linear layers.

To obtain the desired projection layer φ_{θ_p} , we should measure the intensity of gradient conflicts. As for the gradient vectors g_1 and g_2 , which are computed from original fakes and online synthesized fakes, respectively, the intensity of their conflict can be quantified by the dot product of their unit vectors \mathbf{e}_1 and \mathbf{e}_2 . Inspired by this, we propose a Conflict Descent Loss $\mathcal{L}_\varphi(\cdot)$ to guide the reduction of the gradient conflict via \mathbf{e}_1 and \mathbf{e}_2 :

$$\mathcal{L}_\varphi = -\mathbf{e}_1^T \cdot \mathbf{e}_2 = -\frac{g_1^T}{\|g_1\|} \cdot \frac{g_2}{\|g_2\|}. \quad (18)$$

It is worth noting that \mathcal{L}_φ only affects the parameters θ_p in the projection layer φ_{θ_p} . The preceding layers will not be impeded by the projection and can preserve as much specific knowledge as possible from heterogeneous forgery data.

Jointly considering the two binary classification loss (*i.e.*, \mathcal{L}_1 and \mathcal{L}_2) and the conflict descent loss \mathcal{L}_φ , we formulate the total gradients of the feature projection layer φ_{θ_p} as follows:

$$\begin{aligned} \nabla \mathcal{L}_{\theta_p} &= \nabla \mathcal{L}_1 + \nabla \mathcal{L}_2 + \nabla \mathcal{L}_\varphi \\ &= g_{1,\theta_p} + g_{2,\theta_p} - \frac{H_{1,\theta_p} \cdot g_{2,\theta_p} + H_{2,\theta_p} \cdot g_{1,\theta_p}}{\|g_{1,\theta_p}\| \|g_{2,\theta_p}\|}, \end{aligned} \quad (19)$$

where H is the Hessian matrix, representing the second-order partial derivative of the corresponding loss function $\mathcal{L}_\varphi(\cdot)$. Given the parameters $\theta_p = [\theta_1, \theta_2, \dots, \theta_n]$ of projection layer φ_{θ_p} , H can be mathematically defined as:

$$H = \nabla_{\theta_p}^2 \mathcal{L}_\varphi = \begin{bmatrix} \frac{\partial^2 \mathcal{L}_\varphi}{\partial \theta_1^2} & \frac{\partial^2 \mathcal{L}_\varphi}{\partial \theta_1 \partial \theta_2} & \cdots & \frac{\partial^2 \mathcal{L}_\varphi}{\partial \theta_1 \partial \theta_n} \\ \cdots & \cdots & \ddots & \cdots \\ \frac{\partial^2 \mathcal{L}_\varphi}{\partial \theta_n \partial \theta_1} & \frac{\partial^2 \mathcal{L}_\varphi}{\partial \theta_n \partial \theta_2} & \cdots & \frac{\partial^2 \mathcal{L}_\varphi}{\partial \theta_n^2} \end{bmatrix}. \quad (20)$$

Since directly computing the full Hessian matrix H is computationally infeasible, we resort to finding an approximation. Given the properties of the Fisher information matrix (FIM), it is reasonable to assume that, under asymptotic conditions, the Hessian matrix H can be well approximated by the FIM, which is typically estimated via the gradient outer product matrix G during practical training process:

$$H \approx \text{FIM} = \mathbb{E}_{y|x} G. \quad (21)$$

For a given gradient vector $g \in \mathbb{R}^n$, its corresponding gradient outer product matrix $G \in \mathbb{R}^{n \times n}$ is formulated as:

$$G = g \cdot g^T = \begin{bmatrix} g_1^2 & g_1 g_2 & \cdots & g_1 g_n \\ \cdots & \cdots & \ddots & \cdots \\ g_n g_1 & g_n g_2 & \cdots & g_n^2 \end{bmatrix}. \quad (22)$$

To further reduce computational overhead, we diagonalize the outer product matrix G , and finally approximate H as:

$$H \approx \tau \cdot \text{Diag}(G) = \tau \cdot g \otimes g, \quad (23)$$

where the hyper-parameter τ is used to control the variance. The operator \otimes denotes element-wise (Hadamard) product.

Ultimately, according to Equation (23), the gradient of the feature projection layer φ_{θ_p} can be calculated as follows:

$$\begin{aligned} \nabla \mathcal{L}_{\theta_p} &= g_{1,\theta} + g_{2,\theta} \\ &\quad - \gamma(g_{1,\theta_p} \otimes g_{1,\theta_p} \otimes g_{2,\theta_p} + g_{2,\theta_p} \otimes g_{2,\theta_p} \otimes g_{1,\theta_p}), \end{aligned} \quad (24)$$

where denote $\gamma = \frac{\tau}{\|g_{1,\theta_p}\| \|g_{2,\theta_p}\|}$. Guided by the conflict descent loss \mathcal{L}_φ , the feature projection layer φ_{θ_p} gradually maps features into a low-conflict embedding space, while preserving the learned specific patterns of heterogeneous forgery types.

Overall, our proposed CS-DFD primarily focuses on the computation and utilization of gradient vectors. Specifically, the UVS module computes a low-conflict approximate gradient, while the CGR module leverages the gradients to facilitate the learning of low-conflict representations. Through the collaborative effect of these two modules, our approach effectively eliminates gradient conflicts among heterogeneous forgery data, while preserving detection performance in the source domains and enhancing generalization in the target domains. The workflow of our method is outlined in Algorithm 1.

Algorithm 1 An overview of our proposed CS-DFD.

TRAIN
For i **to** MaxEpoch **do**

Input: N labeled samples $[x_i, y_i]_{i=1}^N$ from two subsets $X = [x_r, x_f]$ and $X' = [x_r, x_s]$;

Step 1: Feature extraction and projection;
 $\mathbf{f} = \varphi_{\theta_p}(\mathbf{f})$;

Step 2: Gradient calculation;
 $\nabla \mathcal{L}_{\theta_p} = \nabla \mathcal{L}_1 + \nabla \mathcal{L}_2 + \nabla \mathcal{L}_\varphi$;

$\nabla \mathcal{L}_\varphi \approx -\gamma(g_{1,\theta_p} \otimes g_{1,\theta_p} \otimes g_{2,\theta_p} + g_{2,\theta_p} \otimes g_{2,\theta_p} \otimes g_{1,\theta_p})$;

Step 3: Gradient modification;

$g_0 \leftarrow v^* = g_0 + (\sum_{j=1}^2 \mu_j^* g_j) / 2\lambda$;

$\mu^* \leftarrow \mu^{t+1} = \mu^t - \beta \cdot \nabla_{\mu^t} \psi$;

Step 4: Gradient backpropagation;

$\theta^{t+1} = \theta^t - \alpha \cdot v^*$;

$\theta_{p+1}^{t+1} = \theta_p^t - \alpha \cdot \nabla \mathcal{L}_{\theta_p}$;

End

PREDICT

Input: N' samples from difference target domains;

Step 1: Feature extraction and projection;

Step 2: Classification;

V. EXPERIMENTS

A. Implementation

Training datasets. Following the common setting of deepfake detection studies [42]–[47], we trained our model on the FF++ dataset [18]. It consists of 1,000 real videos and 5,000 manipulated videos generated using five different forgery methods, *i.e.*, Deepfakes (DF), FaceSwap (FS), Face2Face (F2F), FaceShifter (Fsh), and NeuralTextures (NT), resulting in a total of 6,000 videos.

Testing datasets. The official test split of the FF++ dataset is used to evaluate the detection performance on the source domain. Furthermore, four widely used deepfake datasets represent the target domain to evaluate the generalizability:

1) Celeb-DF [62] is one of the most challenging datasets for deepfake detection, which contains 590 original YouTube videos and 5,639 corresponding high-quality deepfake videos.

2) DFDC [63] is one of the largest public deepfake datasets by far, with 23,654 real videos and 104,500 fake videos generated by eight facial manipulation methods. **3) DFDCp** [63] offers a smaller subset of 1,131 real videos and 4,113 forged videos, intended for rapid prototyping and preliminary evaluation ahead of the full DFDC. **4) UADFV** [64] consists of 98 videos (49 real and 49 fake) created using FakeApp.

We utilize Dlib¹ to extract faces and resize them to 256×256 pixels for both the training and testing sets. Our experiments are conducted on a single RTX 3090 GPU with a batch size of 16, and the EfficientNet [48] is employed as the backbone for CS-DFD. Also, to demonstrate the robustness of CS-DFD, we switch the backbone to other networks, *e.g.*, ViT [49] in Section V-B2. The hyperparameters c in the UVS module and

¹<http://dlib.net/>.

TABLE I

PERFORMANCE COMPARISON (%). THE ‘DATA USAGE’ COLUMN INDICATES THE TYPE OF FORGED DATA UTILIZED BY THE MODEL. SPECIFICALLY, ‘ORIGINAL’ DENOTES ORIGINALLY FAKE FORGERIES, WHEREAS ‘ONLINE’ REFERS TO ONLINE FORGERIES. \ddagger : WE RE-IMPLEMENTED THIS DETECTOR. -: THE AUTHORS DID NOT REPORT THE RESULTS ON THIS DATASET IN THEIR ORIGINAL PAPER.

Method	Venue	Data Usage		Testing Dataset					
		Original	Online	FF++	Celeb-DF	DFDC	DFDCp	UADF	Avg
\ddagger EfficientNet [48]	ICML’19	✓	✗	96.67	64.59	65.43	80.27	63.19	68.37
\ddagger Face X-ray [21]	CVPR’20	✓	✗	95.72	74.76	61.57	71.15	64.34	67.95
\ddagger CORE [65]	CVPRW’22	✓	✗	96.61	79.45	62.60	75.74	65.41	70.80
\ddagger RECCE [66]	CVPR’22	✓	✗	96.95	69.71	62.82	74.19	78.61	71.33
\ddagger UCF [55]	ICCV’23	✓	✗	97.16	81.90	66.21	80.94	93.15	80.55
FoCus [67]	TIFS’24	✓	✗	99.15	76.13	68.42	76.62	-	-
Qiao et al. [68]	TPAMI’24	✓	✗	99.00	70.00	-	-	78.00	-
GRU [69]	CVPR’24	✓	✗	98.40	89.00	-	-	-	-
\ddagger Effort [70]	ICML’25	✓	✗	98.85	94.38	74.49	84.22	95.07	87.04
\ddagger VLFDF [71]	CVPR’25	✓	✗	98.64	84.80	71.80	84.74	93.71	83.76
\ddagger I2G-PCL [27]	ICCV’21	✗	✓	92.21	71.12	65.55	73.58	94.08	76.08
\ddagger SBI [25]	CVPR’22	✗	✓	85.16	93.18	72.42	84.15	94.28	86.00
\ddagger FreqBlender [26]	NeurIPS’24	✗	✓	93.29	92.65	73.15	84.56	94.79	86.28
ProDet [10]	NeurIPS’24	✓	✓	95.91	84.48	72.40	81.16	-	-
CS-DFD	-	✓	✓	98.88	95.32	74.97	85.86	95.92	88.02

τ in the CGR module are 0.5 and 0.01, respectively. A detailed analysis of these choices can be found in Section V-D4.

B. Performance Comparison

1) *Comparison with SoTA detectors:* We present the comparison results between our proposed method CS-DFD and SoTA models in Table I. All of them are trained on FF++ and evaluated on five testing datasets, covering both the source and target domains. We use the Area Under the Curve (AUC) metric to evaluate performance on each individual dataset and also report the average AUC across the four target-domain test sets. From Table I, we have the following observations:

i) Training the model solely on original forgeries preserves high accuracy in the source domain, but fails to ensure generalization to the target domain. For example, although FoCus achieves the highest source-domain accuracy of 99.15% due to overfitting behavior, its target-domain AUC drops to 76.13% on Celeb-DF, highlighting poor generalization and limiting its effectiveness in practical applications.

ii) Training the model solely on online synthesized images yields the opposite results. For example, SBI achieves an average AUC of 86% across four cross-domain datasets, substantially outperforming other baselines, but it suffers from a pronounced drop in performance on the source domain, achieving an AUC of 85% on FF++.

iii) Training on both types of forgeries, our method achieves superior performance on the FF++ dataset as well as across all cross-dataset evaluations. For instance, our method achieves an AUC of about 99% on the FF++ dataset, surpassing all baselines that rely on online synthesized images, and performs on par with traditional detectors. Moreover, our approach substantially outperforms all baseline methods, achieving an average AUC of 88% across the target-domain testing datasets.

The above results demonstrate that our CS-DFD overcomes the gradient conflicts between heterogeneous forgery data,

TABLE II
PERFORMANCE OF APPLYING CS-DFD TO DIFFERENT BACKBONES.
ViT-L AND ViT-B REPRESENT DIFFERENT SCALES OF THE ViT.

Backbone	FF++	Celeb-DF	DFDC	DFDCp	UADFV
Xception (O)	96.37	56.75	64.19	72.17	62.05
Xception (S)	77.50	90.11	70.16	73.19	94.12
Xception (O+S)	94.86	87.77	68.96	72.36	92.87
Xception (CS-DFD)	97.48	88.44	74.01	79.89	95.16
ViT-L (O)	97.65	77.27	64.54	78.05	76.13
ViT-L (S)	70.32	88.26	71.98	84.60	93.71
ViT-L (O+S)	90.66	85.15	70.02	83.88	91.34
ViT-L (CS-DFD)	97.89	90.26	73.83	87.15	94.92
ViT-B (O)	98.38	89.63	68.00	80.04	80.17
ViT-B (S)	89.64	93.54	73.63	84.87	92.04
ViT-B (O+S)	98.17	91.28	71.11	82.68	90.21
ViT-B (CS-DFD)	98.44	93.66	76.92	85.30	94.85

thereby effectively leveraging more diverse training data to enhance model performance.

2) *Comparison with different backbones:* To validate the generalizability of our CS-DFD, we apply it to multiple backbone architectures, *i.e.*, Xception, ViT-L, and ViT-B. For each backbone, we conduct training under three baseline strategies: i) original real + original forgeries (denoted as (O)), ii) original real + online synthesized images (denoted as (S)), and iii) original real + original forgeries + online synthesized images (denoted as (O+S)). We compare the performance of these baselines with their counterparts enhanced by our CS-DFD method and summarized the results in Table II.

Across all three backbone architectures, we consistently observe that incorporating online synthesized images during training significantly degrades detection performance in the source domain. For instance, ViT-L trained with synthesized forgeries (ViT-L (S)) achieves only 70% AUC on FF++, a notable drop from the 97% AUC when trained exclusively on original forged images (ViT-L (O)). Moreover, a counterintuitive ‘1 + 1 < 2’ phenomenon arises: combining both original and online synthesized forgeries during training leads

TABLE III
AUC (%) COMPARISON OF DIFFERENT MODULES IN CS-DFD.

Backbone	UVS	CGR	Testing Set				
			FF++	Celeb-DF	DFDC	DFDCp	UADFV
✓			96.67	64.59	65.43	80.27	63.19
✓	✓		97.98	94.45	73.65	84.73	93.32
✓		✓	98.00	94.92	74.64	84.44	94.09
✓	✓	✓	98.88	95.32	74.97	85.86	95.92

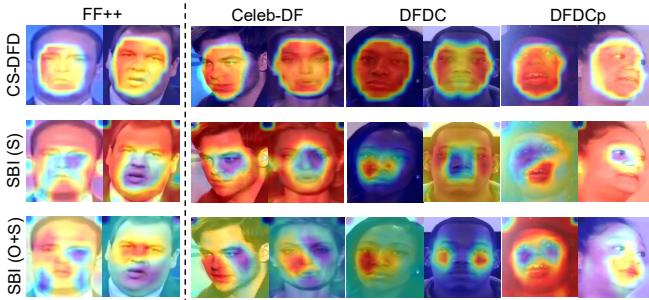


Fig. 4. The heatmaps of three different training strategies (*i.e.*, CS-DFD, SBI (S) and SBI (O+S)) on source domain dataset (*i.e.*, FF++) and target domain datasets (*i.e.*, Celeb-DF, DFDC, and DFDCp). The intensity of red color corresponds to the level of attention.

to performance degradation on both the source and target domains. For instance, Xception trained on the combined dataset (Xception (O+S)) achieves approximately 87% AUC on Celeb-DF, which is significantly lower than the 90% AUC achieved by Xception (S). In contrast, our proposed CS-DFD method effectively mitigates these limitations. For example, when applied to ViT-B, CS-DFD enables the model to surpass all baseline strategies across all datasets. This improved generalization performance on both source and target domains stems from CS-DFD resolving gradient conflicts by balancing the optimization trajectory for the two conflicting data types.

C. Ablation Studies

We report the ablation study results of our proposed method in Table III. Specifically, we progressively integrate the two proposed modules, UVS and CGR, into the backbone model to examine their individual and combined contributions to performance. The results show that both modules independently contribute positively to model performance. For instance, the UVS module improves the AUC on Celeb-DF by 30%. Similarly, the CGR module yields a 9% AUC improvement on the DFDC dataset. It is worth noting that throughout the ablation experiments, our method does not suffer performance degradation in the source domain, even using a combination of online synthesized and original forgery images. This indicates that both modules are effective in alleviating gradient conflict. By combining UVS and CGR, the model achieves the best overall performance, confirming the complementary strengths of the two modules and their joint effectiveness in addressing training conflicts and improving cross-domain generalization.

D. Visualization

1) *Heat Maps:* Fig. 4 presents the heatmaps generated by three different methods, *i.e.*, CS-DFD, SBI (S), and SBI (O+S).

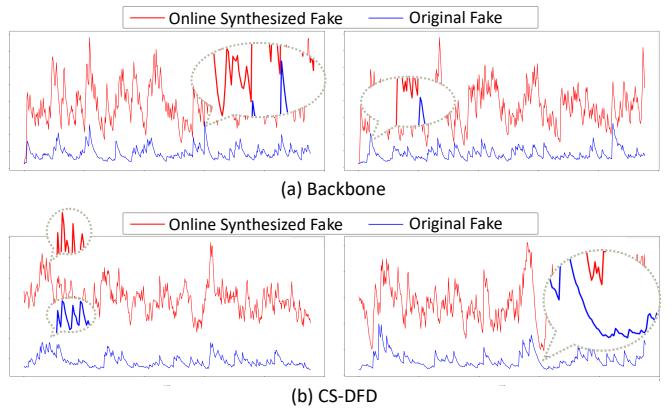


Fig. 5. Loss trend comparison of different methods when trained simultaneously on original forgeries and online synthesized forgeries. For each model, we present two non-cherry-picked training segments. In each subplot, the *x*-axis is the training steps, and the *y*-axis represents the loss value.

Among them, SBI (S) is trained with [real, online synthesized fake images], whereas SBI (O+S) and CS-DFD are trained with [real, online synthesized fake images, original fake images]. As observed in Fig. 4, SBI (S) exhibits a consistent detection tendency across all target domain fake data, primarily focusing on the contours of facial blending. This consistent behavior contributes to improved generalization performance. However, SBI's ability to capture more subtle forgery artifacts is limited. For instance, the heatmap in SBI (S) shows that the model tends to overlook the central facial region, thereby reducing its in-domain effectiveness. In contrast, SBI (O+S) exhibits confused detection patterns across the fake images. Specifically, its attention regions exhibit considerable uncertainty — for instance, on DFDCp, the model sometimes focuses heavily on the eyes, while at other times it attends to the lips. This confusion arises from gradient conflicts introduced by the inclusion of heterogeneous data during training, which ultimately disrupts the optimization direction and degrades performance. In contrast, our method, CS-DFD, clearly overcomes these issues. For instance, it consistently exhibits stable attention regions across different domains, with broader focus areas and richer forensic features. This enables the model to achieve consistently high accuracy across both in-domain and out-of-domain scenarios.

2) *Loss Comparisons:* In Fig. 5, we illustrate the gradient conflict in the backbone model by analyzing the loss trends before and after applying the CS-DFD method. Specifically, Fig. 5(a) shows the loss curves when the model is trained using the traditional approach with [real, online synthesized images, original fake images]. It can be observed that a strong antagonistic pattern exists between the two types of fake data: when the loss for one type peaks, the loss for the other reaches a minimum. This phenomenon indicates a fundamental conflict in optimization directions induced by the two types of data, which prevents the model from effectively learning meaningful forgery patterns. As a result, the model performs poorly on both in-domain and out-of-domain data. In contrast, our proposed CS-DFD method mitigates this conflict. As shown in the figure, the losses for both types of data

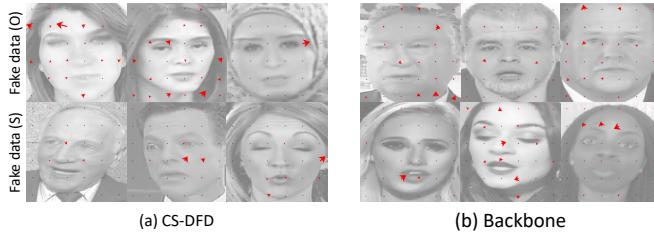


Fig. 6. Visualization of vector fields at different pixel locations for selected training samples. (a) and (b) present the results of our CS-DFD framework and the backbone model, respectively. It can be observed that, under the guidance of CS-DFD, samples within the same training batch (aligned in the same column) exhibit more consistent gradient directions.

follow a consistent and coordinated trend, suggesting that they contribute synergistically to model optimization. This enables the model to maintain strong in-domain detection performance while simultaneously enhancing its generalization capability.

3) *Gradient Vector Field Analysis*: Fig. 6 illustrates the gradient vector field generated using the ‘quiver’ function for a single sample during training. Fig. 6(a) and (b) show the results obtained using our proposed model CS-DFD, and the backbone model, respectively. The red arrows evenly distributed across the image pixels represent the local gradients at different positions. The direction indicated by each arrow represents the direction of the gradient, and the length of the arrow corresponds to the magnitude of the gradient. Meanwhile, to enhance the contrast of the visualization results, the original input images are converted to grayscale to display the gradient vectors better. By adjusting the scale factor in the ‘quiver’ function, only the dominant gradients are shown.

To demonstrate the effectiveness of CS-DFD in mitigating gradient conflicts, we arrange both the originally fake images and online synthesized fake images within the same training batch in the same column. As the gradient vector fields show, training with our CS-DFD, samples from different sources in the same batch exhibit more consistent dominant gradient directions, indicating that the inter-source gradient conflicts have been significantly alleviated.

4) *Hyperparameters Analysis*: In the proposed method, two hyperparameters play a critical role: c in Equation (9) and τ in Equation (23). The former controls the degree to which the update vector v deviates from the original gradient g_0 , ensuring that the new vector remains sufficiently close to preserve training stability while deviating enough to resolve conflicts caused by the original gradient. This enables the optimization to directly benefit from the conflict-free update vector obtained via the UVS module. The latter hyperparameter, τ , governs the sensitivity of the diagonal approximation of the Hessian matrix and directly influences how strongly the conflict loss steers the update direction.

We present a detailed hyperparameter analysis in Fig. 7. The impact of c on model performance exhibits a typical sweet spot curve: as c increases, the AUC on the target domain initially improves, reaches a peak, and then declines. When c is too small, the update vector closely resembles the original gradient, failing to effectively mitigate gradient conflicts. As c increases, these conflicts are alleviated, leading

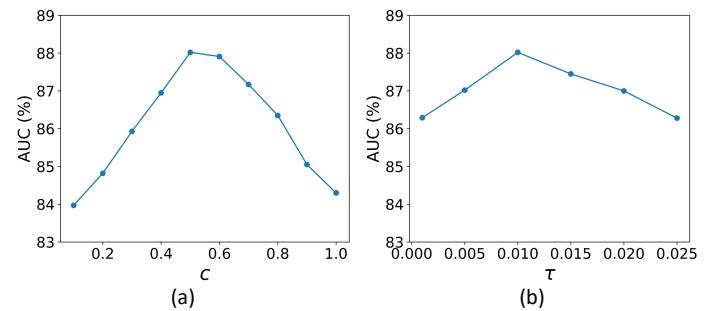


Fig. 7. Hyperparameter sensitivity analysis for c and τ . (a) Adjusting c changes the search range of the update vector, which in turn affects model performance. (b) Varying τ influences the degree of conflict suppression in the embedding space, impacting the generalization ability.

to better generalization. However, excessively large values of c may cause the update direction to deviate too far from the true optimization trajectory, resulting in less reliable result. A similar trend is observed for τ . An appropriately chosen τ effectively constrains conflicting update directions without overly suppressing feature diversity, thereby enhancing generalization. In contrast, a large τ leads to excessive smoothing of projected features, which suppresses expressive capacity and ultimately causes a drop in AUC. Based on these observations, we empirically set $c = 0.5$ and $\tau = 0.01$ as the default hyperparameter configuration for our experiments.

E. Efficiency Analysis

In our experiments, we observe that the computational efficiency of the CS-DFD method is nearly identical to that of the original backbone models. Over the course of the entire training process, CS-DFD incurs only a modest increase in computational time—approximately 2% more than the backbone alone (20.3 hours vs. 20 hours), which is acceptable given the significant gains in generalization performance. A critical reason for this is that both modules in CS-DFD are designed to be lightweight. Specifically, CS-DFD comprises two modules: UVS and CGR. For UVS, during parameter update, gradients are computed separately for original fakes and online synthesized fakes. An update vector is then constructed to minimize the conflict between these two sources. This process involves computing the norms, inner product, and a linear combination of g_1 and g_2 , which has a time complexity of $\mathcal{O}(P)$, where P denotes the total number of model parameters. This cost is negligible when compared to a full backward pass. For CGR, it introduces a small trainable projection layer ϕ_{θ_p} and a conflict descent loss \mathcal{L}_ϕ , which jointly learn embeddings that mitigate gradient conflict. The output dimension d of ϕ_{θ_p} is much smaller than P , making its forward and backward passes computationally inexpensive. The conflict loss employs a Hessian-diagonal approximation (as described in Equation (23)), which involves only element-wise multiplications over the projected feature vectors and has a time complexity of $\mathcal{O}(d)$, and $d \ll P$. In summary, CS-DFD introduces virtually no additional computational overhead, while effectively addressing gradient conflict through its lightweight and efficient design.

VI. CONCLUSION

This work provides an in-depth analysis of the ‘ $1 + 1 < 2$ ’ performance degradation phenomenon in deepfake detection, identifying conflicting gradients from heterogeneous forgery data as the primary cause of poor cross-domain generalization. To address this challenge, the Conflict-Suppressed Deepfake Detection (CS-DFD) framework is proposed. This framework simultaneously aligns the descent directions of losses from diverse data sources and reduces representation-level gradient discrepancies. Extensive experiments on multiple deepfake benchmarks validate the effectiveness of CS-DFD and confirm that suppressing gradient conflicts is essential for achieving robust, generalizable deepfake detection.

REFERENCES

- [1] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, “High-resolution image synthesis with latent diffusion models,” in *CVPR*, 2022, pp. 10 674–10 685.
- [2] F. Bao, C. Li, J. Zhu, and B. Zhang, “Analytic-dpm: an analytic estimate of the optimal reverse variance in diffusion probabilistic models,” in *ICLR*, 2022, pp. 1–12.
- [3] J. Sohl-Dickstein, E. A. Weiss, N. Maheswaranathan, and S. Ganguli, “Deep unsupervised learning using nonequilibrium thermodynamics,” in *ICML*, 2015, pp. 2256–2265.
- [4] M. Kim, F. Liu, A. K. Jain, and X. Liu, “Dcfface: Synthetic face generation with dual condition diffusion model,” in *CVPR*, 2023, pp. 12 715–12 725.
- [5] H. Cheng, Y. Guo, T. Wang, L. Nie, and M. Kankanhalli, “Diffusion facial forgery detection,” in *ACM MM*, 2024, pp. 5939–5948.
- [6] C. Xu, J. Zhang, M. Hua, Q. He, Z. Yi, and Y. Liu, “Region-aware face swapping,” in *CVPR*, 2022, pp. 7622–7631.
- [7] W. Zhao, Y. Rao, W. Shi, Z. Liu, J. Zhou, and J. Lu, “Diffswap: High-fidelity and controllable face swapping via 3d-aware masked diffusion,” in *CVPR*, 2023, pp. 8568–8577.
- [8] K. Sun, S. Chen, T. Yao, Z. Zhou, J. Ji, X. Sun, C.-W. Lin, and R. Ji, “Towards general visual-linguistic face forgery detection,” *arXiv preprint arXiv:2502.20698*, pp. 1–10, 2025.
- [9] Z. Chen, K. Sun, Z. Zhou, X. Lin, X. Sun, L. Cao, and R. Ji, “Diffusionface: Towards a comprehensive dataset for diffusion-based face forgery analysis,” *CoRR*, pp. 1–10, 2024.
- [10] J. Cheng, Z. Yan, Y. Zhang, Y. Luo, Z. Wang, and C. Li, “Can we leave deepfake data behind in training deepfake detector?” in *NeurIPS*, 2024, pp. 1–12.
- [11] C. Hong, Y. Hsu, and T. Liu, “Contrastive learning for deepfake classification and localization via multi-label ranking,” in *CVPR*, 2024, pp. 17 627–17 637.
- [12] R. Xia, D. Liu, J. Li, L. Yuan, N. Wang, and X. Gao, “Mmnet: multi-collaboration and multi-supervision network for sequential deepfake detection,” *IEEE TIFS*, pp. 3409–3422, 2024.
- [13] Z. Yan, Y. Zhao, S. Chen, M. Guo, X. Fu, T. Yao, S. Ding, and L. Yuan, “Generalizing deepfake video detection with plug-and-play: Video-level blending and spatiotemporal adapter tuning,” in *CVPR*, 2025, pp. 12 615–12 625.
- [14] C. Tan, H. Liu, Y. Zhao, S. Wei, G. Gu, P. Liu, and Y. Wei, “Rethinking the up-sampling operations in cnn-based generative network for generalizable deepfake detection,” in *CVPR*, 2024, pp. 28 130–28 139.
- [15] Z. Sun, S. Chen, T. Yao, B. Yin, R. Yi, S. Ding, and L. Ma, “Contrastive pseudo learning for open-world deepfake attribution,” in *ICCV*, 2023, pp. 20 882–20 892.
- [16] G. Chen and C. Hsu, “Jointly defending deepfake manipulation and adversarial attack using decoy mechanism,” *IEEE TPAMI*, pp. 9922–9931, 2023.
- [17] Z. Chen, J. Duan, L. Kang, and G. Qiu, “Supervised anomaly detection via conditional generative adversarial network and ensemble active learning,” *IEEE TPAMI*, pp. 7781–7798, 2023.
- [18] A. Rössler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner, “Faceforensics++: Learning to detect manipulated facial images,” in *ICCV*, 2019, pp. 1–11.
- [19] Y. Qian, G. Yin, L. Sheng, Z. Chen, and J. Shao, “Thinking in frequency: Face forgery detection by mining frequency-aware clues,” in *ECCV*, 2020, pp. 86–103.
- [20] H. Zhao, W. Zhou, D. Chen, T. Wei, W. Zhang, and N. Yu, “Multi-attentional deepfake detection,” in *CVPR*, 2021, pp. 2185–2194.
- [21] L. Li, J. Bao, T. Zhang, H. Yang, D. Chen, F. Wen, and B. Guo, “Face x-ray for more general face forgery detection,” in *CVPR*, 2020, pp. 5000–5009.
- [22] Y. Luo, Y. Zhang, J. Yan, and W. Liu, “Generalizing face forgery detection with high-frequency features,” in *CVPR*, 2021, pp. 16 317–16 326.
- [23] C. Wang and W. Deng, “Representative forgery mining for fake face detection,” in *CVPR*, 2021, pp. 14 923–14 932.
- [24] S. Dong, J. Wang, R. Ji, J. Liang, H. Fan, and Z. Ge, “Implicit identity leakage: The stumbling block to improving deepfake detection generalization,” in *CVPR*, 2023, pp. 3994–4004.
- [25] K. Shiohara and T. Yamasaki, “Detecting deepfakes with self-blended images,” in *CVPR*, 2022, pp. 18 699–18 708.
- [26] J. Zhou, Y. Li, B. Wu, B. Li, J. Dong *et al.*, “Freqblender: Enhancing deepfake detection by blending frequency knowledge,” in *NeurIPS*, 2024, pp. 44 965–44 988.
- [27] T. Zhao, X. Xu, M. Xu, H. Ding, Y. Xiong, and W. Xia, “Learning self-consistency for deepfake detection,” in *ICCV*, 2021, pp. 15 023–15 033.
- [28] X. Xu, Y. Chen, X. Tao, and J. Jia, “Text-guided human image manipulation via image-text shared space,” *IEEE TPAMI*, pp. 6486–6500, 2022.
- [29] Y. Nirkin, L. Wolf, Y. Keller, and T. Hassner, “Deepfake detection based on discrepancies between faces and their context,” *IEEE TPAMI*, pp. 6111–6121, 2022.
- [30] D. P. Kingma and M. Welling, “Auto-encoding variational bayes,” in *ICLR*, 2014, pp. 1–14.
- [31] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. C. Courville, and Y. Bengio, “Generative adversarial nets,” in *NeurIPS*, 2014, pp. 2672–2680.
- [32] J. Thies, M. Zollhöfer, M. Stamminger, C. Theobalt, and M. Nießner, “Face2face: Real-time face capture and reenactment of RGB videos,” in *CVPR*, 2016, pp. 2387–2395.
- [33] T. Karras, S. Laine, and T. Aila, “A style-based generator architecture for generative adversarial networks,” in *CVPR*, 2019, pp. 4401–4410.
- [34] J. Bao, D. Chen, F. Wen, H. Li, and G. Hua, “Towards open-set identity preserving face synthesis,” in *CVPR*, 2018, pp. 6713–6722.
- [35] V. Blanz, K. Scherbaum, T. Vetter, and H. Seidel, “Exchanging faces in images,” *Computer Graphics Forum*, pp. 669–676, 2004.
- [36] Q. Zhang, M. Tao, and Y. Chen, “Gddim: Generalized denoising diffusion implicit models,” *arXiv preprint arXiv:2206.05564*, 2022.
- [37] A. Q. Nichol and P. Dhariwal, “Improved denoising diffusion probabilistic models,” in *ICML*, 2021, pp. 8162–8171.
- [38] M. W. Y. Lam, J. Wang, D. Su, and D. Yu, “BDDM: bilateral denoising diffusion models for fast and high-quality speech synthesis,” in *ICLR*, 2022, pp. 1–12.
- [39] J. Song, C. Meng, and S. Ermon, “Denoising diffusion implicit models,” in *ICLR*, 2021, pp. 1–12.
- [40] J. Ho, A. Jain, and P. Abbeel, “Denoising diffusion probabilistic models,” in *NeurIPS*, 2020, pp. 1–12.
- [41] K. Kim, Y. Kim, S. Cho, J. Seo, J. Nam, K. Lee, S. Kim, and K. Lee, “Diffface: Diffusion-based face swapping with facial guidance,” *CoRR*, pp. 1–11, 2022.
- [42] Z. Yan, Y. Luo, S. Lyu, Q. Liu, and B. Wu, “Transcending forgery specificity with latent space augmentation for generalizable deepfake detection,” in *CVPR*, 2024, pp. 8984–8994.
- [43] W. Guan, W. Wang, J. Dong, and B. Peng, “Improving generalization of deepfake detectors by imposing gradient regularization,” *IEEE TIFS*, pp. 5345–5356, 2024.
- [44] M.-H. Liu, X.-Q. Liu, X. Luo, and X.-S. Xu, “Data: Multi-disentanglement based contrastive learning for open-world semi-supervised deepfake attribution,” *arXiv preprint arXiv:2505.04384*, 2025.
- [45] S. Jia, C. Ma, T. Yao, B. Yin, S. Ding, and X. Yang, “Exploring frequency adversarial attacks for face forgery detection,” in *CVPR*, 2022, pp. 4093–4102.
- [46] X. Wu, Z. Xie, Y. Gao, and Y. Xiao, “Sstnet: Detecting manipulated faces through spatial, steganalysis and temporal features,” in *ICASSP*, 2020, pp. 2952–2956.
- [47] H. Zhang, M. Liu, Z. Liu, X. Song, Y. Wang, and L. Nie, “Multi-factor adaptive vision selection for egocentric video question answering,” in *ICML*, 2024, pp. 59 310–59 328.
- [48] M. Tan and Q. V. Le, “Efficientnet: Rethinking model scaling for convolutional neural networks,” in *ICML*, 2019, pp. 6105–6114.

- [49] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, “An image is worth 16x16 words: Transformers for image recognition at scale,” in *ICLR*, 2021, pp. 1–12.
- [50] H. Cheng, Y. Guo, T. Wang, Q. Li, X. Chang, and L. Nie, “Voice-face homogeneity tells deepfake,” *ACM ToMM*, pp. 1–22, 2023.
- [51] A. Haliassos, R. Mira, S. Petridis, and M. Pantic, “Leveraging real talking faces via self-supervision for robust forgery detection,” in *CVPR*, 2022, pp. 14 930–14 942.
- [52] T. Mittal, U. Bhattacharya, R. Chandra, A. Bera, and D. Manocha, “Emotions don’t lie: An audio-visual deepfake detection method using affective cues,” in *ACM MM*, 2020, pp. 2823–2832.
- [53] K. Sun, T. Yao, S. Chen, S. Ding, J. Li, and R. Ji, “Dual contrastive learning for general face forgery detection,” in *AAAI*, 2022, pp. 2316–2324.
- [54] L. Chen, Y. Zhang, Y. Song, L. Liu, and J. Wang, “Self-supervised learning of adversarial example: Towards good generalizations for deepfake detection,” in *CVPR*, 2022, pp. 18 689–18 698.
- [55] Z. Yan, Y. Zhang, Y. Fan, and B. Wu, “UCF: uncovering common features for generalizable deepfake detection,” in *ICCV*, 2023, pp. 22 355–22 366.
- [56] Y.-H. Han, T.-M. Huang, S.-T. Lo, P.-H. Huang, K.-L. Hua, and J.-C. Chen, “Towards more general video-based deepfake detection through facial feature guided adaptation for foundation model,” in *CVPR*, 2025, pp. 22 995–23 005.
- [57] B. Liu, X. Liu, X. Jin, P. Stone, and Q. Liu, “Conflict-averse gradient descent for multi-task learning,” *NeurIPS*, pp. 18 878–18 890, 2021.
- [58] T. Yu, S. Kumar, A. Gupta, S. Levine, K. Hausman, and C. Finn, “Gradient surgery for multi-task learning,” in *NeurIPS*, vol. 33, 2020, pp. 5824–5836.
- [59] B. M. Le and S. S. Woo, “Gradient alignment for cross-domain face anti-spoofing,” in *CVPR*, 2024, pp. 188–199.
- [60] P. Wang, Z. Zhang, Z. Lei, and L. Zhang, “Sharpness-aware gradient matching for domain generalization,” in *CVPR*, 2023, pp. 3769–3778.
- [61] C. Gong and D. Wang, “Nasvit: Neural architecture search for efficient vision transformers with gradient conflict-aware supernet training,” in *ICLR*, 2022.
- [62] Y. Li, X. Yang, P. Sun, H. Qi, and S. Lyu, “Celeb-df: A large-scale challenging dataset for deepfake forensics,” in *CVPR*, 2020, pp. 3204–3213.
- [63] B. Dolhansky, J. Bitton, B. Pflaum, J. Lu, R. Howes, M. Wang, and C. Canton-Ferrer, “The deepfake detection challenge dataset,” *CoRR*, pp. 1–13, 2020.
- [64] Y. Li, M.-C. Chang, and S. Lyu, “In ictu oculi: Exposing ai generated fake face videos by detecting eye blinking,” in *WIFS*, 2018, pp. 1–7.
- [65] Y. Ni, D. Meng, C. Yu, C. Quan, D. Ren, and Y. Zhao, “CORE: consistent representation learning for face forgery detection,” in *CVPRW*, 2022, pp. 12–21.
- [66] J. Cao, C. Ma, T. Yao, S. Chen, S. Ding, and X. Yang, “End-to-end reconstruction-classification learning for face forgery detection,” in *CVPR*, 2022, pp. 4103–4112.
- [67] J. Tian, P. Chen, C. Yu, X. Fu, X. Wang, J. Dai, and J. Han, “Learning to discover forgery cues for face forgery detection,” *IEEE TIFS*, pp. 3814–3828, 2024.
- [68] T. Qiao, S. Xie, Y. Chen, F. Retraint, and X. Luo, “Fully unsupervised deepfake video detection via enhanced contrastive learning,” *IEEE TPAMI*, pp. 4654–4668, 2024.
- [69] J. Choi, T. Kim, Y. Jeong, S. Baek, and J. Choi, “Exploiting style latent flows for generalizing deepfake video detection,” in *CVPR*, 2024, pp. 1133–1143.
- [70] Z. Yan, J. Wang, P. Jin, K.-Y. Zhang, C. Liu, S. Chen, T. Yao, S. Ding, B. Wu, and L. Yuan, “Orthogonal subspace decomposition for generalizable ai-generated image detection,” in *ICML*, 2025.
- [71] K. Sun, S. Chen, T. Yao, Z. Zhou, J. Ji, X. Sun, C.-W. Lin, and R. Ji, “Towards general visual-linguistic face forgery detection,” in *CVPR*, 2025, pp. 19 576–19 586.