Signal Processing: Image Communication 144 (2026) 117525

Contents lists available at ScienceDirect

Signal Processing: Image Communication

journal homepage: www.elsevier.com/locate/image

3D differential decomposition for video deepfake detection with identity suppression

Jie Gao [a,b], Marco Micheletto [b,*], Giulia Orrù [b], Xiaoyi Feng [a], Gian Luca Marcialis [b]

[a] Northwestern Polytechnical University, 1 Dongxiang Road, Xi'an, 710129, China
[b] University of Cagliari, Department of Electrical and Electronic Engineering, Via Marengo 3, Cagliari, 09123, Italy

ARTICLE INFO

Keywords:
3D differential modeling
Video deepfake detection
Identity suppression

ABSTRACT

Detecting deepfake videos remains a challenging task, especially in scenarios involving unknown manipulation methods or unseen data distributions. Most existing video deepfake detection methods rely on high-level semantic features, which often lead to overfitting of facial identity information and poor transferability. In this work, we explore a novel perspective by modeling videos through 3D differential operations along temporal and spatial dimensions. To exploit the spatial–temporal variation information of the video content, the proposed approach decomposes videos into single-axis 1D differential signals, which are then transformed into 2D representations for efficient learning. This procedure enables the use of lightweight 2D CNNs while retaining directional forgery cues. Our experiments, aimed at analyzing whether these differential signals capture discriminative patterns useful for distinguishing real from fake content, show that the proposed method achieves strong intra-dataset performance and reveals complementary information across dimensions. These findings suggest that differential signals could potentially support generalization when integrated into broader detection frameworks.

1. Introduction

The rapid advancement of deepfake generation technologies [1,2], driven by modern generative models such as diffusion-based architectures [3–5], text-to-image tools [6], and emerging text-to-video systems (e.g., Sora [7]), has significantly accelerated the synthesis of highly realistic yet entirely fabricated audiovisual content, including fake audio [8,9], video [10,11], and text [12]. While these technologies unlock new opportunities in entertainment, animation, and virtual communication, they simultaneously pose serious challenges for content authenticity and raise pressing concerns regarding public trust in digital media. In response to these risks, the research community has dedicated substantial effort to developing automated deepfake detection systems [13]. State-of-the-art approaches typically leverage the representational power of convolutional neural networks to extract high-level visual features and detect subtle artifacts indicative of tampering [14]. However, despite their high intra-dataset performance, many of these methods suffer from limited generalization to unseen manipulation techniques or distributions. One recurring issue is the over-reliance on facial identity-specific features, which tend to dominate the learned representations and hinder the model's ability to capture manipulation-related cues [15].

Several recent studies have explored strategies aimed at enhancing generalization, including multi-domain feature extraction, spanning spatial [16,17], temporal [18], and frequency domains [14], as well as data augmentation [19] and ensemble-based methods [20]. Although these approaches have shown promising performance, they often introduce high computational costs and complex data augmentation pipelines, limiting their real-world applicability. To overcome these limitations, we propose a 3D differential decomposition framework for video-based deepfake detection, designed to amplify manipulation traces while minimizing the influence of identity-specific information.

Our approach is grounded in prior findings showing that generative forgeries tend to introduce directional inconsistencies along the temporal and spatial axes, and that the presence of identity-specific cues can impair generalization across manipulation types [21,22]. We hypothesize that such inconsistencies are most prominent along three orthogonal dimensions, time, height, and width, and can be effectively captured through directional derivatives [23]. By decomposing each video into one-dimensional differential signals along these axes and mapping them into compact two-dimensional representations, the method retains forgery-related information while enabling efficient training using lightweight 2D CNNs, thereby achieving substantial

* Corresponding author.
E-mail addresses: jie_gao@mail.nwpu.edu.cn (J. Gao), marco.micheletto@unica.it (M. Micheletto), giulia.orru@unica.it (G. Orrù), fengxiao@nwpu.edu.cn (X. Feng), gianluca.marcialis@unica.it (G.L. Marcialis).

https://doi.org/10.1016/j.image.2026.117525
Received 8 July 2025; Received in revised form 4 November 2025; Accepted 20 February 2026
Available online 24 February 2026

computational savings without compromising the ability to capture discriminative manipulation patterns. To further reduce identity leakage and improve robustness, we introduce a dedicated preprocessing strategy that removes identity-revealing content and constructs pseudo-smooth interpolation samples to challenge the model's discriminative capability. We also treat the selection of the differential order as an optimization parameter and explore several fusion schemes to integrate complementary information across dimensions and differential levels.

In summary, the main contributions of this work are as follows:

- We introduce a novel 3D differential modeling paradigm for video deepfake detection, which reformulates the task as a problem of analyzing the smoothness of three one-dimensional signals derived along the temporal and spatial axes.
- We design a lightweight preprocessing and transformation pipeline that suppresses identity information and maps the differential signals into a compact 2D representation suitable for standard 2D CNNs.
- We define an optimization formulation for differential order selection and explore four fusion strategies to integrate multi-directional and multi-order signals. Our extensive experiments demonstrate the effectiveness and flexibility of the proposed framework.

The rest of the paper is organized as follows. Section 2 reviews related work on deepfake detection in different feature domains and modeling strategies. Section 3 introduces our proposed 3D differential decomposition framework, including its mathematical formulation, preprocessing techniques, and transformation strategy. Section 4 presents experimental results and ablation studies in multiple datasets, followed by discussion. Finally, Section 5 concludes the paper and outlines the directions for future research.

2. Related work

Deepfake detection has emerged as a critical task due to the increasing sophistication and accessibility of generative technologies capable of synthesizing hyper-realistic facial content [24]. A large body of work has focused on identifying artifacts induced by manipulation, which may occur either at the spatial level of individual frames or across the temporal dimension of videos. Accordingly, detection methods can be broadly categorized into frame-based and video-based approaches.

2.1. Frame-based DeepFake detection

Frame-level detection methods analyze individual video frames and primarily focus on spatial artifacts such as texture irregularities, blending boundaries, noise residuals, or inconsistencies in color and illumination. These methods predominantly leverage Convolutional Neural Networks (CNNs), which have proven to be powerful tools for spatial feature extraction across diverse fields. Beyond deepfake detection, CNNs are widely applied in areas such as medical science, where they aid in disease detection from medical imagery [25], and in remote sensing, where they analyze satellite data for environmental monitoring [26]. In the context of deepfakes, early approaches exploited low-level synthesis traces and hand-crafted features. For instance, Wang et al. [27] studied forensic noise patterns extracted from facial and background regions to differentiate real from synthetic content. Zhao et al. [28] proposed a multi-attention mechanism coupled with a textural enhancement module to highlight subtle artifacts. Gao et al. [16] introduced a dual-stream architecture designed to disentangle artifact-related and texture-related features, improving robustness across datasets. More recent works have focused on enhancing generalization through architectural or data-level diversity. Chen et al. [29] built a dynamic forgery configuration pool and employed adversarial training to generate diverse fake samples. Sun et al. [30] proposed a dual-contrastive learning strategy that uses both inter-instance and intra-instance constraints to guide representation learning. Yu et al. [31] incorporated disparity maps and meta-learning to enable zero-shot adaptation, while Li et al. [32] introduced Face X-ray, a supervised model trained on images with synthetic blending boundaries to highlight low-level compositional traces. Nirkin et al. [33] focused on inconsistencies between facial identity and context by training dual recognition networks on the face and non-face regions. Gao et al. [14] explored deepfake detection under heavy compression by fusing frequency-domain and RGB-based features through multi-branch networks, addressing degradation from codec artifacts and post-processing. Tan et al. [34] presented FreqNet, a frequency-space learning network that operates directly in the Fourier domain, focusing on high-frequency representations of both images and intermediate features to improve the robustness of the cross-domain with a lightweight architecture. More recently, Yermakov et al. [35] used the CLIP ViT-L/14 encoder with parameter-efficient fine-tuning (LN-tuning) and hyperspherical regularization, showing that minimal adaptation of large vision-language models can yield strong generalization across unseen forgery techniques. Although frame-based methods have demonstrated strong performance in detecting intra-frame artifacts, they inherently fail to leverage temporal information, which plays a critical role in detecting sequential manipulations. Moreover, their reliance on spatially learned semantics makes them particularly vulnerable to manipulations with high visual fidelity [36] and limits their ability to generalize under distribution shifts, compression, or subtle forgery pipelines [36,37].

2.2. Video-based DeepFake detection

Temporal-based approaches aim to capture temporal inconsistencies caused by frame-wise manipulations lacking inter-frame coherence. Zheng et al. [38] proposed a temporal-only convolutional model with 1×1 spatial kernels, later extended with transformer-based modules for long-range dependencies. Gu et al. [39] introduced a framework that models horizontal and vertical temporal discrepancies through differential analysis and further expanded this idea into a hierarchical contrastive framework [40] that combines global and local views. The same authors also designed intra-snippet and inter-snippet modules to isolate inconsistencies between frame segments [41]. Similarly, Zhao et al. [37] proposed a transformer with decomposed attention and self-subtraction to highlight artifacts, while Lu et al. [42] combined long-range attention with a spatial–temporal encoder to model global variations. Chen et al. [43] proposed a method based on 3D spatiotemporal trajectories. By constructing robust motion features from 2D and 3D frames and analyzing motion trajectories in phase space, their approach also maintains detection performance on compressed videos. Another critical axis of research concerns the role of facial identity. Some approaches explicitly leverage identity as a signal. For instance, Petmezas et al. [44] introduced a hybrid CNN-LSTM-Transformer model that uses 3D Morphable Models for facial feature extraction and operates on an identity verification principle, comparing a test video against reference videos of the genuine individual. While this can yield high accuracy and fast inference, it fundamentally ties the detector's validity to the availability of pristine reference data, making it a form of identity-based verification rather than a generic artifact detector. This reliance on identity underscores a fundamental risk for generic detection models: even when not explicitly designed for it, they may learn to rely on the victim's identity rather than the manipulation artifacts, which may hinder generalization in more unconstrained scenarios. Dong et al. [15] formalized this problem as *Implicit Identity Leakage* (IIL), showing that even when not explicitly supervised, standard binary classifiers tend to internalize facial identity information, which becomes a confounding factor during generalization. They introduced an ID-unaware detection model using artifact supervision to counteract this effect, improving generalization across datasets. However, their approach depends on the accurate localization

of manipulated regions and artifact-level annotations, which are not always available in practice. Our framework is positioned at the intersection of these considerations. We aim to capture subtle directional inconsistencies while minimizing identity leakage through differential decomposition and compact signal representation.

3. The proposed method

3.1. Motivation

A long-standing principle in video analysis is that explicit differences between adjacent observations provide simple yet effective proxies for temporal change and local inconsistencies. Rather than relying exclusively on raw frames, difference-based representations emphasize variations over static content and can expose departures from natural spatio-temporal behavior. This idea has proved broadly useful across video tasks: inter-frame differences have been used to capture motion cues efficiently in super-resolution [45,46], to provide competitive motion representations in action recognition at substantially lower cost than optical flow (e.g., RGB-difference inputs and temporal-difference modules in TSN/TDN) [47,48], and to reveal manipulation traces through frame-to-frame rate-of-change features in video forensics [49]. Deepfake detection, in particular, hinges on exposing intrinsic inconsistencies that differentiate synthetic from authentic sequences. Natural videos typically exhibit smooth signal evolution over time and across space, reflecting physical continuity; in contrast, synthetic content may introduce subtle artifacts and misalignments due to imperfections in generative models, leading to detectable discontinuities [39,50]. These discrepancies are rarely evident at the pixel level but emerge when observing how signals evolve along spatial and temporal dimensions. Building on the above evidence that difference-based representations amplify changes while suppressing static content, we cast deepfake detection as an assessment of signal smoothness: authentic content should preserve stable spatio-temporal behavior, while manipulated content should manifest detectable fluctuations.

To quantify and emphasize such discrepancies, we propose the use of first-order directional differences applied to short video clips. This differential operation acts as a signal enhancer, magnifying temporal or spatial inconsistencies that may otherwise be suppressed in raw input data. As highlighted in Section 2.2, conventional CNN-based approaches, which operate directly on frame images or volumetric video blocks, tend to entangle manipulation cues with identity-specific features, potentially undermining generalization, especially in cross-manipulation or cross-dataset settings. To mitigate this, our approach explicitly suppresses identity information by transforming raw video clips into a differential representation that highlights local signal variations while discarding static identity-related features. The proposed methodology proceeds in three conceptual stages.

- **Differential modeling**: raw video clips are decomposed using directional first-order differences along multiple spatio-temporal axes. This step is designed to amplify local signal irregularities.
- **Dimensional transformation**: the resulting multi-dimensional difference signals are then projected into a consistent 2D map representation, enabling the use of image-based backbones while preserving the enhanced artifacts.
- **Architectural specialization**: separate sub-networks are trained on different differential maps to specialize in various axes of manipulation, with fusion strategies designed to improve robustness.

Each of these stages is described in detail in the following subsections (see Fig. 1).

3.2. Multiaxial differential representation of video data

Building upon the rationale outlined in the previous Section, we seek to enhance the separability of forged content by characterizing local variations in visual signals. The core idea is to quantify irregularities via finite differences applied to structured signal representations. As a starting point, we consider the case of a one-dimensional discrete signal and define its successive backward finite differences as a means to capture changes in local smoothness. Let consider a one-dimensional discrete signal F of length N, defined as a sequence $\{x_1, x_2, \dots, x_a, \dots, x_N\}$. The backward finite differences of increasing order can be recursively computed as:

$$\begin{cases} \Delta^1 x_a = x_a - x_{a-1} \\ \Delta^2 x_a = \Delta^1(\Delta^1 x_a) \\ \quad\vdots \\ \Delta^{N-1} x_a = \Delta^1(\cdots(\Delta^1 x_a)) \end{cases} \tag{1}$$

where the sign Δ refers to finite difference operation. Each application of this operator isolates the local variation at a finer level. Thus, we can further obtain $N-1$ differential sequences $F^1, F^2, \dots, F^{N-1}$, which are described as formula (2).

$$\begin{cases} F^1 = \{\Delta^1 x_1, \dots, \Delta^1 x_a, \dots, \Delta^1 x_N\} \\ F^2 = \{\Delta^2 x_1, \dots, \Delta^2 x_a, \dots, \Delta^2 x_N\} \\ \quad\vdots \\ F^{N-1} = \{\Delta^{N-1} x_1, \dots, \Delta^{N-1} x_a, \dots, \Delta^{N-1} x_N\} \end{cases} \tag{2}$$

To apply this paradigm to videos, we model each input as a 4D tensor $\boldsymbol{V} \in \mathbb{R}^{C\times T\times H\times W}$ (channels C, frames T, spatial size $H \times W$). To compute directional finite differences along each axis $\alpha \in \{T, H, W\}$ with index set $\mathcal{I}_\alpha = \{1, \dots, L_\alpha\}$, we define the slicing map Π_α, which exposes the axis-ordered sequence of 2D slices orthogonal to α:

$$\begin{cases} S^T &= \{\, S_t^T := \boldsymbol{V}_{:,t,:,:} \in \mathbb{R}^{C\times H\times W} \,\}_{t=1}^{T}, \\ S^H &= \{\, S_h^H := \boldsymbol{V}_{:,:,h,:} \in \mathbb{R}^{C\times T\times W} \,\}_{h=1}^{H}, \\ S^W &= \{\, S_w^W := \boldsymbol{V}_{:,:,:,w} \in \mathbb{R}^{C\times T\times H} \,\}_{w=1}^{W}, \end{cases} \tag{3}$$

with index maps $F_T(t) = S_t^T$, $F_H(h) = S_h^H$, $F_W(w) = S_w^W$. Collecting the slice sequence and its index map, we write:

$$\boldsymbol{V} \xrightarrow{\Pi_T} (S^T, F_T), \qquad \boldsymbol{V} \xrightarrow{\Pi_H} (S^H, F_H), \qquad \boldsymbol{V} \xrightarrow{\Pi_W} (S^W, F_W). \tag{4}$$

To preserve sequence length, avoid wrap-around/reflection at the boundaries, and keep the differencing causal along the slice index, we work with the zero-extended version of each directional sequence, namely $S_k^\alpha = \mathbf{0}$ whenever $k \notin \mathcal{I}_\alpha$. Once the video tensor has been decomposed into directional sequences of 2D slices, differential operators are recursively applied along each axis to extract higher-order variations. The resulting transformation emphasizes localized discontinuities by capturing the evolution of signal changes across adjacent slices, yielding difference maps that highlight subtle temporal and spatial inconsistencies. Formally, the process is implemented as a recursive application of finite differences on each directional sequence of slices, as expressed below:

$$\begin{cases} \Delta^1 S_t^T = S_t^T - S_{t-1}^T \\ \Delta^2 S_t^T = \Delta^1(\Delta^1 S_t^T) \\ \quad\vdots \\ \Delta^{n_T} S_t^T = \Delta^1(\Delta^1(\cdots(\Delta^1 S_t^T))) \end{cases} \tag{5}$$

$$\begin{cases} \Delta^1 S_h^H = S_h^H - S_{h-1}^H \\ \Delta^2 S_h^H = \Delta^1(\Delta^1 S_h^H) \\ \quad\vdots \\ \Delta^{n_H} S_h^H = \Delta^1(\Delta^1(\cdots(\Delta^1 S_h^H))) \end{cases} \tag{6}$$

$$\begin{cases} \Delta^1 S_w^W = S_w^W - S_{w-1}^W \\ \Delta^2 S_w^W = \Delta^1(\Delta^1 S_w^W) \\ \quad\vdots \\ \Delta^{n_W} S_w^W = \Delta^1(\Delta^1(\cdots(\Delta^1 S_w^W))) \end{cases} \tag{7}$$

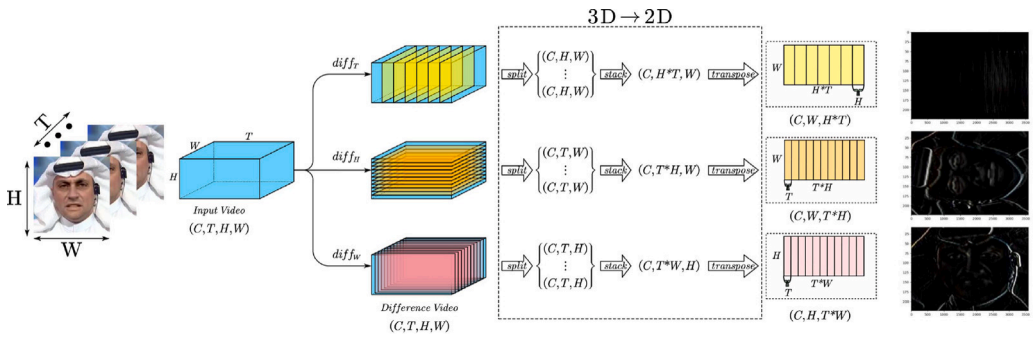

Fig. 1. Overview of the 3D-to-2D projection process. For each axis, directional differences are computed, trimmed, interpolated, and collapsed into 2D maps.

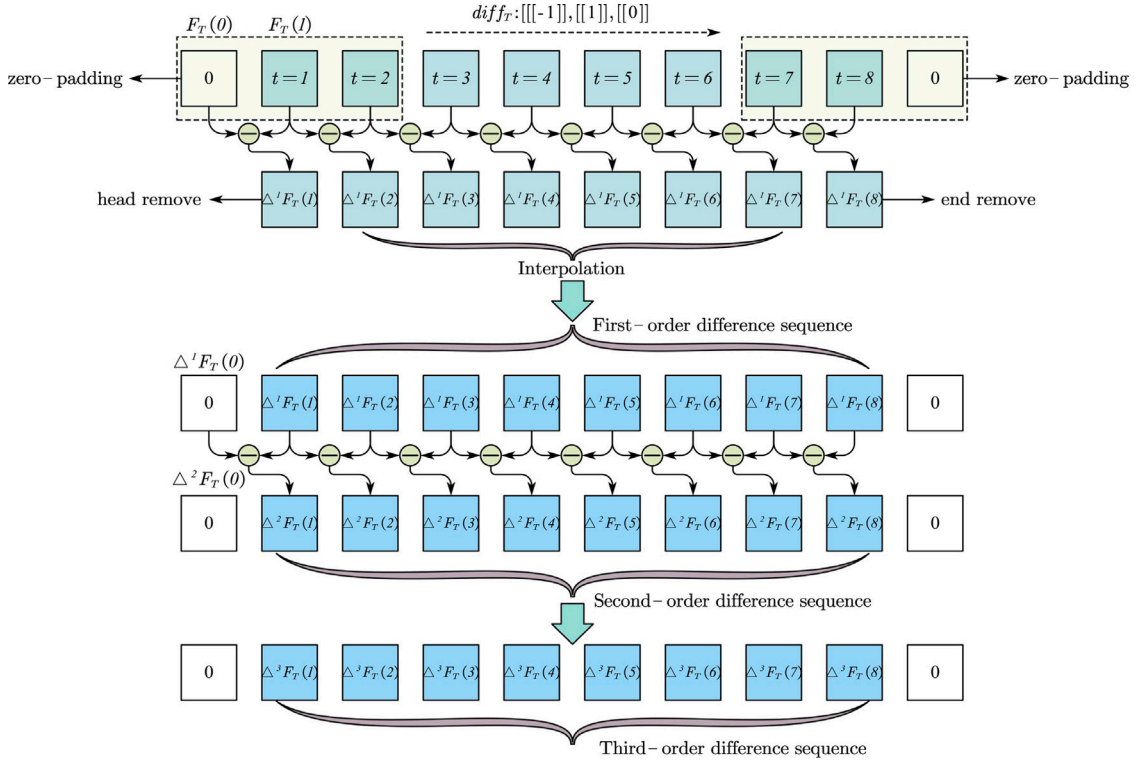

Fig. 2. Temporal differencing along the T-axis, with symmetric trimming and interpolation to mitigate padding artifacts.

The use of zero-padding at sequence boundaries, however, introduces artificial context at the initial and final slices of each directional stack. Such regions often retain static features, particularly identity-related content, due to the absence of genuine variation. To mitigate this, let $U^\alpha = \{\Delta^1 S^\alpha_k\}_{k=1}^{L_\alpha}$ denote the raw first-order sequence; we apply a symmetric trimming operator $\mathcal{T}_\alpha$ that discards both extremes, $\mathcal{T}_\alpha(U^\alpha) = \{U^\alpha_k\}_{k=2}^{L_\alpha-1}$, and an interpolation operator $\mathcal{J}_\alpha$ to restore the original length, yielding the boundary-corrected seed $\widetilde{U}^\alpha := \mathcal{J}_\alpha(\mathcal{T}_\alpha(U^\alpha))$. All higher orders are then evaluated on this corrected seed: $\Delta^1 S^\alpha := U^\alpha$ and, for $n_\alpha \geq 2$, $\Delta^{n_\alpha} S^\alpha := \Delta^{n_\alpha - 1}(\widetilde{U}^\alpha)$. The process is also illustrated in Fig. 2.

We can also express the backward differencing in a binomial, operator-level closed form that makes the recursion explicit. For any sequence $U = \{U_k\}_{k=1}^{L_\alpha}$, adopt the null extension $U_k = \mathbf{0}$ whenever $k \notin \mathcal{I}_\alpha$. Let I_α be the identity operator $(\mathrm{I}_\alpha U)_k = U_k$ and let B_α be the backward shift $(\mathsf{B}_\alpha U)_k := U_{k-1}$ (with the same null extension). Then the first-order backward difference can be written as

$$\Delta_\alpha := \mathrm{I}_\alpha - \mathsf{B}_\alpha. \tag{8}$$

For any integer $n \geq 0$,

$$\Delta_\alpha^n = (\mathrm{I}_\alpha - \mathsf{B}_\alpha)^n = \sum_{m=0}^{n} (-1)^m \binom{n}{m} \mathsf{B}_\alpha^m, \tag{9}$$

and, applied elementwise to U,

$$(\Delta_\alpha^n U)_k = \sum_{m=0}^{n} (-1)^m \binom{n}{m} U_{k-m} \qquad \text{with } U_{k-m} = \mathbf{0} \text{ if } k - m \notin \mathcal{I}_\alpha. \tag{10}$$

Taking $U = S^\alpha = \{S^\alpha_k\}$ gives the same first-order definition as above,

$$(\Delta^1 S^\alpha)_k = S^\alpha_k - S^\alpha_{k-1}. \tag{11}$$

Consistently with our boundary-corrected pipeline, if $U^\alpha := \Delta^1 S^\alpha$ and $\widetilde{U}^\alpha$ is its trimmed-and-interpolated version (same length L_α), then for any $n_\alpha \geq 2$

$$(\Delta^{n_\alpha} S^\alpha)_k = (\Delta_\alpha^{n_\alpha - 1} \widetilde{U}^\alpha)_k = \sum_{m=0}^{n_\alpha - 1} (-1)^m \binom{n_\alpha - 1}{m} \widetilde{U}^\alpha_{k-m}, \tag{12}$$

under the same null-extension convention.

Finally, each directional difference sequence $\Delta^{n_\alpha} S^\alpha = \{\Delta^{n_\alpha} S^\alpha_k\}_{k=1}^{L_\alpha}$ is collapsed into a single 2D map by an axis-preserving reshape that

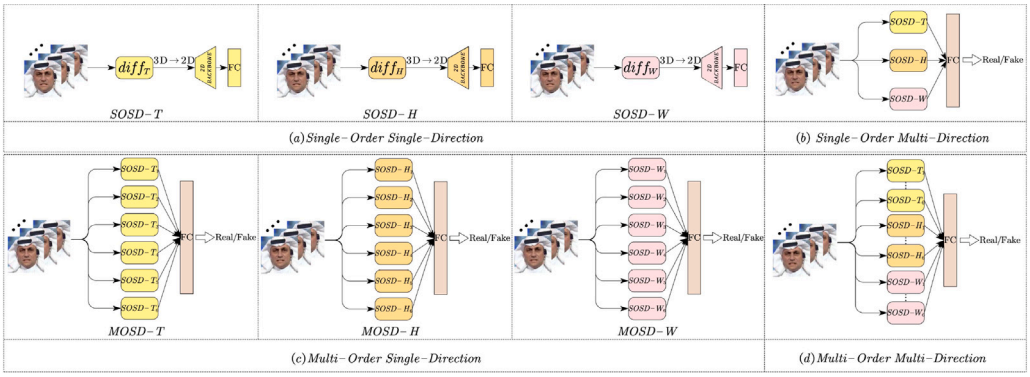

Fig. 3. Model configurations derived from the 3D differential decomposition framework: (a) single-order, single-direction; (b) single-order, multi-direction; (c) multi-order, single-direction; (d) multi-order, multi-direction. All models share the same feature extraction backbone structure and differ in how differential cues are combined.

keeps neighborhood along α: we permute the tensor so that α is the last index and stack the L_α slices contiguously along one spatial dimension, obtaining a map of size $C \times (D_1 \cdot L_\alpha) \times D_2$, where $(D_1, D_2) = (H, W)$ for $\alpha = T$, (T, W) for $\alpha = H$, and (T, H) for $\alpha = W$. The resulting maps serve as inputs to the 2D backbones, as detailed in the next Section.

3.3. Model construction and fusion strategy

This Section introduces the architectural design of the models built upon the differential decomposition framework. The approach starts by considering differences computed along a single axis of the video volume, either temporal or spatial, and incrementally incorporates additional axes and difference orders.

In particular, we define each model by two key aspects: (1) the *direction* of the differential operator, which can act along the temporal axis (T), vertical spatial axis (H), or horizontal spatial axis (W); (2) the *order* of the operation, which corresponds to how many times the discrete difference is applied recursively.

Starting from basic models that apply a first-order difference along a single direction, we progressively construct more complex configurations by combining multiple directions and multiple orders. For clarity, we adopt the following naming convention:

- *Single-Order Single-Direction* (SOSD) models apply a single-order difference along one axis;
- *Single-Order Multi-Direction* (SOMD) models combine single-order differences from all three axes;
- *Multi-Order Single-Direction* (MOSD) models aggregate multiple orders of difference along a single axis;
- *Multi-Order Multi-Direction* (MOMD) models integrate all orders and directions into a unified framework.

All models rely on a shared backbone structure to process the differential representations derived from the input videos. An overview of the resulting architectures is shown in Fig. 3.

3.3.1. Single-Order Single-Direction (SOSD)

The Single-Order Single-Direction (SOSD) configuration constitutes the fundamental unit of the proposed modeling paradigm. Each SOSD model computes the nth order difference of the input video volume along a specific axis, temporal (T), height (H), or width (W), and processes the resulting volume through a 3D-to-2D transformation before applying a 2D convolutional backbone, as shown in Fig. 3(a). The complete procedure is formalized in Algorithm 1.

As previously discussed, the core idea behind this decomposition is that real and synthetic content often differ in their local fluctuation patterns, which can manifest differently along temporal and spatial dimensions. By applying differential operators of increasing order along a specific axis, the model may capture latent traces of manipulation, particularly those invisible at raw signal level. From a theoretical standpoint, the use of higher-order differences is grounded in the Taylor series expansion, which approximates a smooth function as a weighted sum of its successive derivatives. On a discrete grid, the continuous derivatives are replaced by forward finite differences and the factorial weights by binomial coefficients (Newton–Gregory expansion). Let $F(i)$ denote a one-dimensional discrete signal sampled along a generic axis,

Algorithm 1: Single-Order Single-Direction (SOSD)

Input: Video tensor $\boldsymbol{V} \in \mathbb{R}^{1\times C\times T\times H\times W}$;
selected axis $\alpha \in \{T, H, W\}$;
difference order $n \geq 1$;
Output: Predicted label $\hat{y} \in \{0, 1\}$
Function Backbone $M\,(\mathbb{R}^{C\times(D_1\cdot L_\alpha)\times D_2} \to \mathbb{R}^{1\times 2})$:
  Result: logits for the input map after the axis-preserving collapse

begin
  1: $\boldsymbol{V}_{\text{curr}} \leftarrow \boldsymbol{V}$; // residual volume (initially the raw video)
  2: $S^\alpha \leftarrow \Pi_\alpha(\boldsymbol{V}_{\text{curr}})$; // axis-ordered slice sequence along α; use two-sided null extension for out-of-range indices
  3: for $i = 1$ **to** n **do**
    if $i = 1$ **then**
      $U^\alpha \leftarrow \{\Delta^1 S^\alpha_k\}_{k=1}^{L_\alpha}$; // backward finite difference: $(\Delta^1 S^\alpha)_k = S^\alpha_k - S^\alpha_{k-1}$
      if $n = 1$ **then**
        $\text{seq}_{\text{curr}} \leftarrow U^\alpha$
      else
        $\widetilde{U}^\alpha \leftarrow \mathcal{J}_\alpha\big(\mathcal{T}_\alpha(U^\alpha)\big)$; // symmetric trim of ends, then interpolation to restore length L_α
        $\text{seq}_{\text{curr}} \leftarrow \widetilde{U}^\alpha$
    else
      $\text{seq}_{\text{curr}} \leftarrow \Delta_\alpha(\text{seq}_{\text{curr}})$; // apply the next-order backward difference along α
  5: *3D-to-2D collapse*: build $X \in \mathbb{R}^{C\times L_\alpha\times D_1\times D_2}$ by setting $X_{:,k,:,:} := \text{seq}_{\text{curr},k}$ for $k = 1, \ldots, L_\alpha$; then $Y \leftarrow \pi_{(C, D_1, L_\alpha, D_2)}(X)$ and $\boldsymbol{I} \leftarrow \text{reshape}\big(Y;\ C \times (D_1 L_\alpha) \times D_2\big)$;
  ; // if $\alpha = T$: $D_1 = H$, $D_2 = W$; if $\alpha = H$: $D_1 = T$, $D_2 = W$; if $\alpha = W$: $D_1 = T$, $D_2 = H$
  7: $\boldsymbol{z} \leftarrow \text{Backbone } M(\boldsymbol{I})$, $\hat{y} \leftarrow \arg\max \boldsymbol{z}$

assumed to be smooth, and let a be a reference index. The signal can be locally approximated as:

$$F(i) = \sum_{k=0}^{N-1} \binom{i-a}{k} \Delta^k F(a) \tag{13}$$

However, in practical scenarios involving deepfake generation, the observed signal $\tilde{F}(i)$ can be modeled as the superposition of a structured smooth component $F(i)$ and a perturbation term $G(i)$ arising from the generative process:

$$\tilde{F}(i) = F(i) + G(i) \tag{14}$$

While $F(i)$ is assumed to be smooth and differentiable, the additive noise component $G(i)$ introduced by the generative model is typically non-smooth and may encode local perturbations or generative artifacts. Despite this, discrete signals allow for finite-difference operations to be applied regardless of smoothness.

Since finite differences (as defined in Section 3.2) are linear, applying the nth order operator to (14) yields the additive decomposition:

$$\Delta^n \tilde{F}(i) = \Delta^n F(i) + \Delta^n G(i), \tag{15}$$

whose first–order instance is simply:

$$\Delta\tilde{F}(i) = \tilde{F}(i) - \tilde{F}(i-1) = \Delta F(i) + \Delta G(i). \tag{16}$$

In the case of a genuinely smooth signal $F(i)$, the contribution of $\Delta^n F(i)$ tends to decay with increasing n, due to the inherently low local variation of structured content. Conversely, the perturbation component $\Delta^n G(i)$, typically dominated by high-frequency distortions, remains substantial, thereby exposing fine-grained artifacts introduced by manipulation processes. However, this effect comes with inherent trade-offs: higher-order differences may also accentuate noise or attenuate meaningful structural patterns. Therefore, selecting an appropriate differential order becomes a non-trivial design choice, requiring a careful balance between discriminative sensitivity and representational robustness. To formalize this, we treat the SOSD task as a binary classification problem. Let $y \in \{0, 1\}$ denote the ground-truth label of the video sample, and let $\hat{y}_T$, $\hat{y}_H$, and $\hat{y}_W$ denote the predictions obtained when applying the differential operation along the respective axes. The classification loss for each axis-specific model, conditioned on its differential order n_T, n_H, and n_W, is defined as:

$$\begin{cases} L\left(y, \hat{y}_T | n_T\right) = -y \log\left(\hat{y}_T | n_T\right) - (1-y)\log\left(1 - \hat{y}_T | n_T\right) \\ L\left(y, \hat{y}_H | n_H\right) = -y \log\left(\hat{y}_H | n_H\right) - (1-y)\log\left(1 - \hat{y}_H | n_H\right) \\ L\left(y, \hat{y}_W | n_W\right) = -y \log\left(\hat{y}_W | n_W\right) - (1-y)\log\left(1 - \hat{y}_W | n_W\right) \end{cases} \tag{17}$$

In order to balance discriminative power and representational fidelity, we cast the order selection problem as a discrete optimization task. The optimal n_T^*, n_H^*, and n_W^* are defined as the minimizers of the respective axis-specific loss functions:

$$\begin{cases} n_T^* = \arg\min_{n_T} L\left(y, \hat{y}_T | n_T\right) \\ n_H^* = \arg\min_{n_H} L\left(y, \hat{y}_H | n_H\right) \\ n_W^* = \arg\min_{n_W} L\left(y, \hat{y}_W | n_W\right) \end{cases} \tag{18}$$

Hence, each optimal value reflects the most effective trade-off between sensitivity to manipulations and retention of discriminative content.

3.3.2. Axis–order fusion layer

While SOSD units are designed to extract manipulation traces along a specific axis and order, real-world artifacts often exhibit residual patterns across multiple orientations and scales. To leverage this complementary information, the proposed framework supports composite configurations built by aggregating multiple SOSD branches.

Let $\mathcal{B} = \{ f^{(b)} \in \mathbb{R}^2 \mid b = 1, \ldots, B\}$ denote a set of such branches, each producing a 2D logit vector. Their concatenation forms a joint representation which can be represented by formula (19).

$$f_{\mathcal{B}} = [\, f^{(1)} \parallel f^{(2)} \parallel \cdots \parallel f^{(B)}] \in \mathbb{R}^{2B} \tag{19}$$

This is passed through a lightweight fusion layer with parameters $W \in \mathbb{R}^{2\times 2B}$ and $b \in \mathbb{R}^2$ to produce the final logits:

$$\boldsymbol{z} = W f_{\mathcal{B}} + b, \qquad \hat{y} = \arg\max \boldsymbol{z}. \tag{20}$$

During training, all SOSD backbones in $\mathcal{B}$ are frozen, and only the fusion layer parameters (W, b) are updated. This mechanism enables the model to exploit directional and hierarchical cues without disrupting the specialization encoded in individual branches.

3.3.3. Single-Order Multi-Direction (SOMD)

The Single-Order Multi-Direction (SOMD) configuration is constructed to jointly consider the decision outputs derived from multiple directional analyses. Since each SOSD branch operates independently along a single axis, aggregating their logit-level predictions allows the system to incorporate diverse geometric perspectives while preserving architectural modularity.

To this end, SOMD instantiates three SOSD units, each applying a differential operator of fixed order n along one of the principal axes. The resulting logit vectors define the set $\mathcal{B}_{\text{SOMD}} = \{f_T, f_H, f_W\}$, with each $f_\alpha \in \mathbb{R}^2$ corresponding to the output of the branch along axis $\alpha \in \{T, H, W\}$.

The concatenated vector $f_{\mathcal{B}_{\text{SOMD}}} \in \mathbb{R}^6$ is computed as in Eq. (19) and passed through the fusion layer described in Eq. (20). The overall architecture is shown in Fig. 3(b).

3.3.4. Multi-Order Single-Direction (MOSD)

The Multi-Order Single-Direction (MOSD) configuration extends the SOSD paradigm by aggregating multiple logit-level outputs computed at increasing differential orders along a single axis. This design allows the architecture to encode directional traces at various levels of granularity, from low-order variations to higher-order residual dynamics.

Given a target axis $\alpha \in \{T, H, W\}$ and a maximum order N, the MOSD-α model instantiates N SOSD branches, each configured to apply the nth order differential operator $(1 \le n \le N)$ along the selected direction. We define the set of order-specific logit vectors as $\mathcal{B}_{\text{MOSD-}\alpha} = \{f_\alpha^{(1)}, \ldots, f_\alpha^{(N)}\}$, where each $f_\alpha^{(n)} \in \mathbb{R}^2$ is the output of the nth SOSD-α branch.

The resulting concatenated vector $f_{\mathcal{B}_{\text{MOSD-}\alpha}} \in \mathbb{R}^{2N}$ is computed as in Eq. (19), and passed through the fusion layer defined in Eq. (20). This setup enables the aggregation of directional information across multiple differential depths, while retaining the modular independence of each constituent branch. The MOSD architecture is illustrated in Fig. 3(c).

3.3.5. Multi-Order Multi-Direction (MOMD)

The Multi-Order Multi-Direction (MOMD) configuration generalizes the previous variants by combining multiple differential representations across both axes and orders. This structure is intended to capture a broad range of manipulation patterns, jointly encoding directional diversity and hierarchical depth.

The MOMD architecture comprises three MOSD modules, each targeting one of the principal axes $\alpha \in \{T, H, W\}$ with maximum order N. We define the combined set of logit vectors as

$$\mathcal{B}_{\text{MOMD}} = \{\, f_\alpha^{(n)} \in \mathbb{R}^2 \mid \alpha \in \{T, H, W\},\ n = 1, \ldots, N\}$$

with cardinality $|\mathcal{B}_{\text{MOMD}}| = 3N$. The concatenated representation $f_{\mathcal{B}_{\text{MOMD}}} \in \mathbb{R}^{6N}$ is computed following Eq. (19), and the final logits are obtained via Eq. (20).

By jointly leveraging differential signals from multiple orientations and scales, MOMD provides a flexible and modular way to model heterogeneous manipulation patterns. The complete architecture is shown in Fig. 3(d).

4. Experimental results

This Section presents a comprehensive experimental evaluation of the proposed framework. We first describe the datasets and training protocols, followed by a series of controlled studies designed to assess the effectiveness of the differential modeling strategies introduced in Section 3. The analysis addresses both intra- and cross-domain scenarios, providing quantitative and qualitative evidence for the benefits of axis–order modeling and modular fusion. Our goal is not only to validate the performance of individual configurations but also to gain insights into how residual traces emerge across different differential axes and orders.

4.1. Datasets and training protocol

We conduct experiments on four widely-used deepfake benchmarks: FaceForensics++ (FF++) [51], Celeb-DF [52], WildDeepfake (WildDF) [53], and the Deepfake Detection Challenge dataset (DFDC) [54]. Each dataset exhibits unique characteristics in terms of manipulation techniques, video realism, and distribution shifts. Below, we summarize their properties and describe the sampling strategies adopted in our study:

- **FaceForensics++ (FF++)**: This dataset includes five forgery types (Deepfakes (DF), Face2Face (F2F), FaceSwap (FS), FaceShifter (FSh), and NeuralTextures (NT)) each comprising 1000 manipulated videos. The forgeries differ in both synthesis mechanism and visual quality: DF and FSh perform face replacement using deep learning models; F2F and NT manipulate facial expressions via reenactment techniques; FS relies on geometric fitting and 3D projection. We use 750 real and 750 synthetic videos for training, and reserve 250 real and 250 synthetic videos for testing, separately for each manipulation type.
- **Celeb-DF** : Designed to address limitations in synthesis quality, Celeb-DF includes 590 real celebrity videos (Celeb-real), 300 real YouTube videos (YouTube-real), and 5639 high-quality synthetic videos (Celeb-synthesis). Artifacts commonly found in earlier datasets, such as lip-sync mismatches or blending artifacts, are substantially reduced. We select 700 real and 700 synthetic videos for training, and 190 real and 190 synthetic videos for testing.
- **WildDeepfake (WildDF)**: Built to reflect in-the-wild conditions, WildDF aggregates 707 manipulated videos and 7314 face sequences collected from the web. The synthetic content is highly diverse, featuring generative models based on GANs, diffusion models, and hybrid pipelines. We construct a training set of 1200 real and 1200 synthetic videos, and a test set of 300 real and 300 synthetic videos.
- **Deepfake Detection Challenge (DFDC)**: DFDC is the largest publicly available benchmark for facial forgery detection, with over 100,000 videos. It includes manipulated videos exhibiting variations in compression, resolution, noise, lighting, and camera angle, thereby simulating unconstrained real-world conditions. For our experiments, we randomly select 6000 real and 6000 synthetic videos for training, and 2000 real and 2000 synthetic videos for testing.

To preserve temporal consistency across frames, which is crucial for detecting manipulations, we avoid any form of cropping that may desynchronize facial regions. For each video, we extract 32 consecutive frames and divide them into two non-overlapping clips of 16 frames each, as a temporal sampling strategy. We emphasize that this segmentation into 16-frame clips is used solely for convenience in sampling and does not influence the granularity of our differential computation: all finite differences are computed operatorially on axis-ordered slice sequences as defined in Section 3.2 (Eqs. (5)–(7)), independently of this segmentation.

Moreover, to better understand domain-specific and domain-agnostic generalization, we adopt two training protocols:

- **Single-Domain Training.** The training set includes only one type of forgery from FF++ at a time. For each forgery type, we use 750 real and 750 synthetic videos for training, with the remaining 250 real and 250 synthetic videos for evaluation.
- **Multi-Domain Training.** We construct a composite training set from FF++ by randomly sampling 150 manipulated videos from each of the five forgery types, for a total of 750 synthetic videos. These are paired with the same 750 real videos used in the single-domain protocol. The test set remains unchanged to ensure comparability across settings.

It is worth noting that all experiments are framed as binary classification tasks (real vs. fake). Manipulation types (e.g., DeepFakes, Face2Face, FaceSwap, FaceShifter, NeuralTextures) are used only to define training and testing splits for cross-manipulation and cross-domain evaluation, not as separate output classes.

4.2. Implementation details

All SOSD variants are implemented using a 2D ResNet-18 backbone [55] pretrained on ImageNet. Each model processes single-order differences along a specific axis, using differential images as input. For composite architectures (SOMD, MOSD, MOMD), each SOSD branch operates as a frozen feature extractor, and a lightweight fully connected layer is added for fusion. Only this final layer is optimized, while all convolutional weights remain fixed. The models are trained for 15 epochs using cross-entropy loss, with the Stochastic Gradient Descent (SGD) optimizer, which has a momentum of 0.9 and a weight decay of 5e-2. Our experiments are conducted on a Linux system equipped with an NVIDIA GeForce RTX 3090 GPU. The initial learning rate is set to 0.0002, and a StepLR scheduler is employed to decrease the learning rate by a factor of 10 every 10 epochs.

4.3. Evaluation of differential architectures

In this section, we evaluate the impact of the differential modeling strategies on deepfake detection. We assess the performance of the architectures introduced in Sections 3.3.1–3.3.5, analyzing the role of axis direction, differential order, and fusion strategy. All results are reported in terms of Accuracy and Area Under the ROC Curve (AUC), computed on the test split of each dataset.

4.3.1. Visual analysis and verification

To gain preliminary insight into the behavior of differential operators across different axes and orders, we perform a qualitative and quantitative analysis on real and synthetic videos from FaceForensics++ (FF++). The goal is twofold: (1) to visually assess whether identity traces are progressively attenuated by higher-order differences, and (2) to investigate whether residual distortions introduced by forgery methods become more salient under differential transformation.

Qualitative observations. We first visualize the effect of differential operations of increasing order along the three axes on a real video sample. Fig. 4 displays the output of applying differential operators of order $n \in \{0, \dots, 6\}$, with the input consisting of 16 aligned frames. Along the spatial axes (H and W), the resulting images reveal a gradual suppression of coarse facial features as the order increases. Structural elements such as eyes, mouth, and jawlines progressively vanish, confirming the expected attenuation of low-frequency content. In contrast, the temporal axis (T) exhibits a different behavior: the first-order difference already removes most of the face identity, and further increasing the order yields minimal visual change. This suggests that identity information is more strongly concentrated along spatial dimensions, while temporal differences capture finer dynamics that do not always correlate with facial content.

To verify whether differential maps also enhance the separation between real and synthetic data, we compare outputs across the five

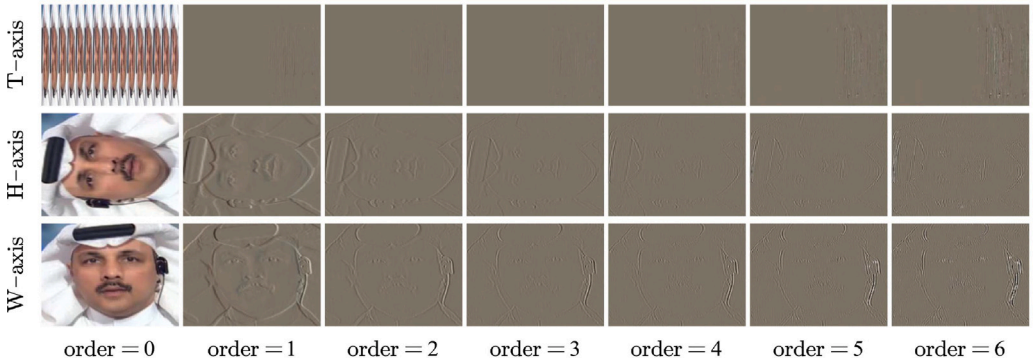

Fig. 4. Visualization of a real sample from FF++ after applying differential operators of order $n \in \{0, \ldots, 6\}$ along each principal axis (T, H, W). Each row corresponds to a fixed axis and each column to a specific order n.

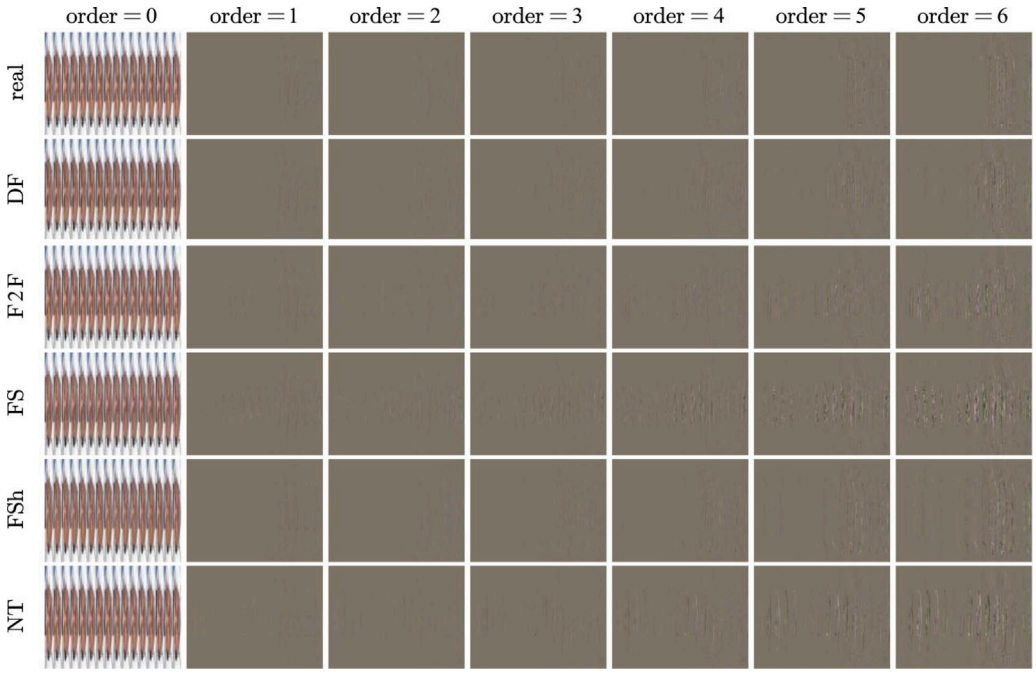

Fig. 5. Visualization of real and synthetic samples from FF++ after applying differential operators of order $n \in \{0, \ldots, 6\}$ along T axis. Each row corresponds to a specific forgery method, and each column to a differential order n.

manipulation types in FF++ (DF, F2F, FS, FSh, NT) against real samples. As shown in Figs. 5–7, the differential outputs reveal texture residuals and edge inconsistencies in synthetic samples, with mid-range orders (e.g., $n \in \{2, 3, 4\}$) often yielding the most perceptually distinct contrasts. Overall, certain orders and axes appear to emphasize tampering cues more effectively, depending on the structure and nature of the forgery.

Quantitative analysis. We complement the visual observations with statistical measurements of image texture using gray-level co-occurrence matrix (GLCM) descriptors. Specifically, we analyze two standard metrics: contrast (which measures intensity variability) and homogeneity (which measures local similarity). Fig. 8 reports the average GLCM contrast and homogeneity over 10 real and 50 synthetic FF++ videos, for orders $n \in \{0, \ldots, 6\}$ along each axis. Temporal and spatial patterns differ substantially. Along H and W, synthetic samples consistently exhibit higher contrast and lower homogeneity, indicating more abrupt residual variations and reduced smoothness compared to real content. Such trends remain stable across forgery methods. The T-axis shows a less predictable behavior: while contrast increases with order, the separation between real and synthetic data is less pronounced. This behavior may stem from the inherent variability of facial motion, which exhibits high-frequency changes that degrade the signal-to-noise ratio of temporal differences.

In summary, these results support the hypothesis that differential operators can highlight forgery artifacts by exposing inconsistencies along specific axes and orders. Spatial differentials are particularly informative, with residual patterns in forged data showing greater heterogeneity and contrast. Assessing how these cues translate into effective decision boundaries becomes the objective of the following section.

4.3.2. Effect of single-axis differencing

The first experimental question investigates whether differential operators applied along individual axes can enhance detection performance and generalization, particularly in challenging cross-forgery conditions. To this end, we evaluate Single-Order Single-Direction (SOSD) models, each trained on a specific differential axis — temporal (T), vertical (H), or horizontal (W) — and differential order $n \in \{0, \ldots, 6\}$. The case $n = 0$ corresponds to standard RGB frames. Results on the FF++ dataset are summarized in Table 1 for the T-axis, with evaluations on the H- and W-axes reported in Tables 2 and 3, respectively.

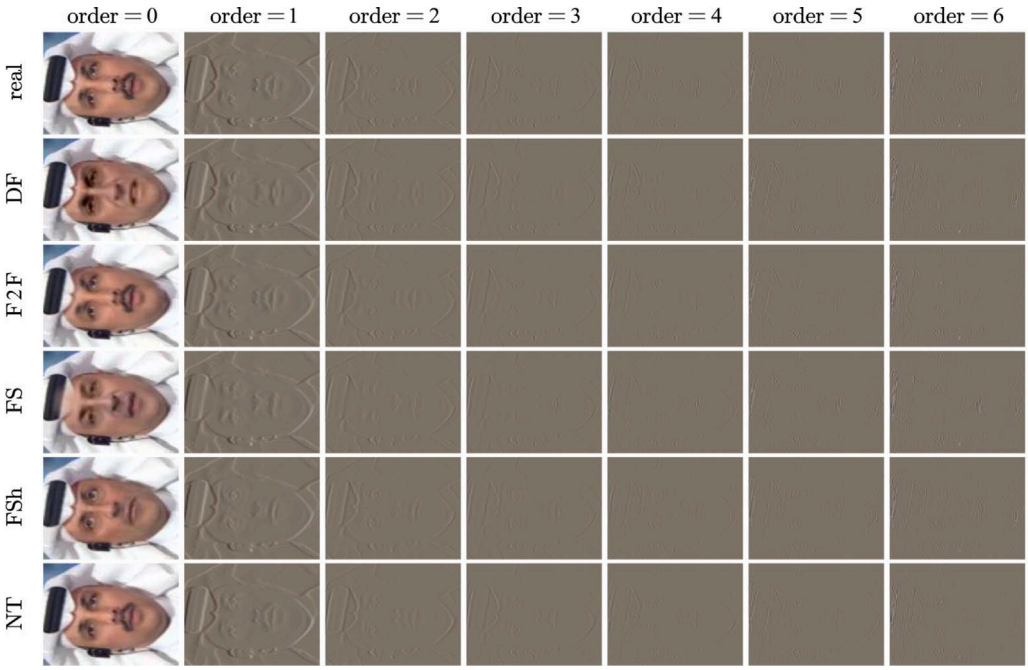

Fig. 6. Visualization of real and synthetic samples from FF++ after applying differential operators of order $n \in \{0, \ldots, 6\}$ along H axis. Each row corresponds to a specific forgery method, and each column to a differential order n.

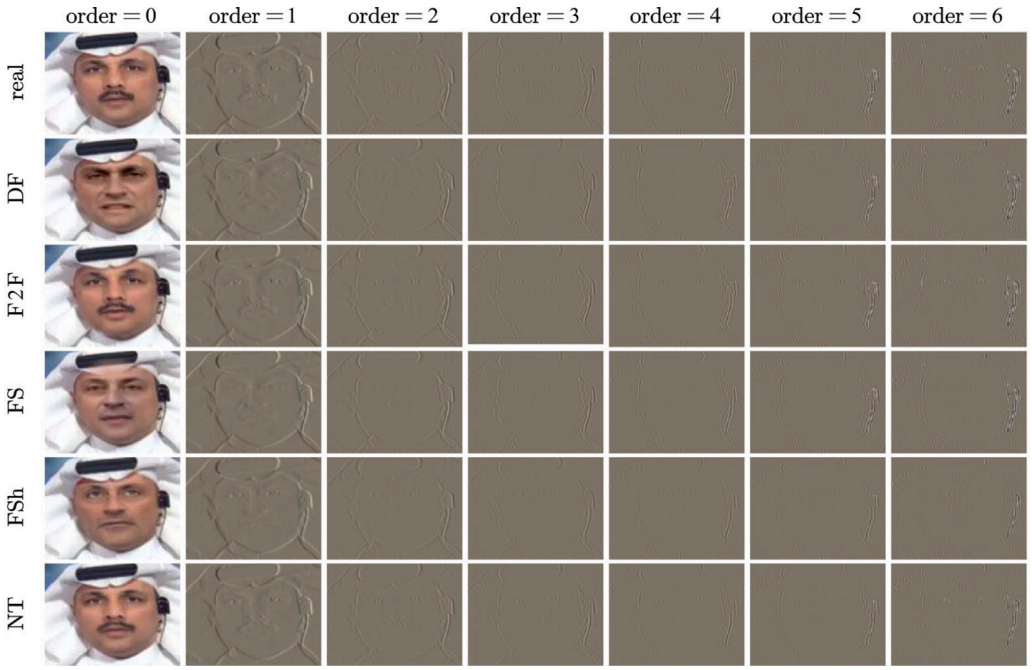

Fig. 7. Visualization of real and synthetic samples from FF++ after applying differential operators of order $n \in \{0, \ldots, 6\}$ along W axis. Each row corresponds to a specific forgery method, and each column to a differential order n.

For each model, we report both *Within-Domain* (WD), computed as the mean performance across all FF++ domains including the training domain, and *Cross-Domain* (CD), computed as the mean performance on test domains excluding the training one, thus reflecting generalization capability. Full per-domain results for each order and axis are available in the supplemental material.

In the single-domain setting, consistent trends emerge across axes and manipulation types.

First, low-to-mid order differencing (n = 1–3) systematically improves cross-forgery AUC over the RGB baseline. Higher orders ($n \geq 4$) tend to degrade performance, likely due to over-differencing and increased noise. For instance, when trained on FS using the T-axis, the $n = 3$ model reaches a cross-forgery AUC of 63.48%, improving over the RGB baseline ($n = 0$, 46.10%) by +17.4 points. Comparable gains are observed for the H- and W-axes (Tables 2–3), with peak improvements in the range of +1.1 to +17.4 points depending on axis and source.

Second, spatial differencing proves more stable across domains. Across the 20 evaluated cross-forgery transfers, H- or W-axis models with $n \leq 3$ surpass the RGB baseline in 18 cases (90%), while temporal differencing achieves the same in 14 cases (70%). These findings align

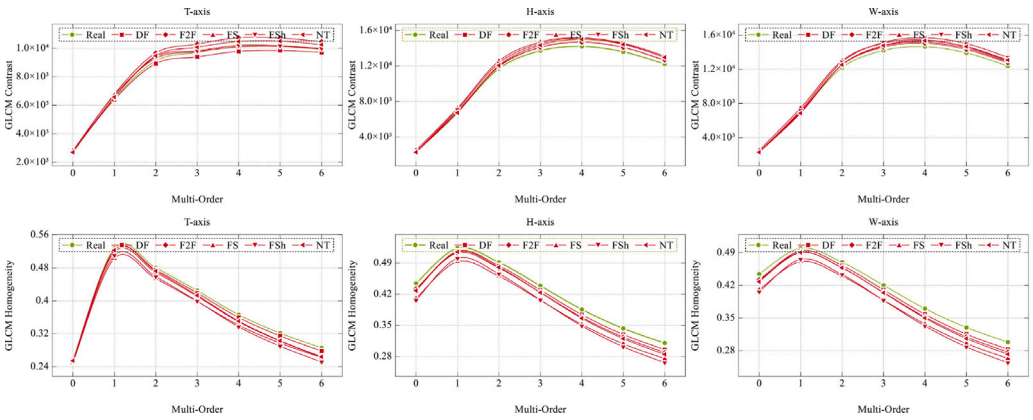

Fig. 8. Gray-level co-occurrence matrix (GLCM) contrast (top) and homogeneity (bottom) for differential images computed along each axis (T, H, W) and order $n \in \{0, \ldots, 6\}$. Curves report the average over 10 real and 50 synthetic samples from FF++.

Table 1
Evaluation of SOSD-T models across differential orders and FF++ training domains. Accuracy (%) and AUC (%) are reported for each differential order. *WD* (Within-Domain): average on all test domains (including training); *CD* (Cross-Domain): average on test domains excluding training. Bold highlights the best Accuracy and AUC per domain.

Order	DF		F2F		FS		FSh		NT		All FF++	
	WD	CD	WD	CD	WD	CD	WD	CD	WD	CD	WD	CD
0	62.78/69.79	54.08/62.36	61.22/**66.76**	52.23/**58.52**	58.84/56.83	49.15/46.10	60.32/61.46	50.70/51.95	66.78/73.37	60.50/67.43	84.48/91.56	–
1	61.80/70.94	52.35/63.67	60.58/65.18	51.18/56.54	59.74/60.53	50.20/50.79	60.48/61.44	51.03/51.88	**67.68/75.67**	**61.05/69.97**	89.98/95.99	–
2	63.32/72.08	54.25/65.11	60.22/65.57	50.68/57.04	59.64/66.53	49.93/58.25	63.20/66.32	54.30/57.93	64.12/70.73	56.70/63.88	90.42/96.80	–
3	63.52/71.68	54.48/64.61	61.04/64.98	51.65/56.28	60.00/**70.76**	50.38/**63.48**	64.04/66.12	55.33/57.70	63.38/72.13	55.63/65.59	89.90/96.23	–
4	61.64/71.04	52.18/63.83	60.76/64.31	51.43/55.50	59.96/69.93	50.33/62.46	64.98/**66.98**	56.65/**58.76**	64.00/71.22	56.68/64.57	90.28/96.44	–
5	**64.00/73.29**	**55.13/66.62**	60.88/64.34	51.53/55.52	59.76/62.23	50.10/52.83	65.20/66.58	56.90/58.28	64.70/73.41	57.30/67.20	**91.62/96.90**	–
6	62.08/71.12	52.68/63.91	**61.34**/65.30	**52.25**/56.71	**61.08**/68.20	**51.63**/60.28	**65.78**/66.46	**57.70**/58.16	63.28/73.71	55.53/67.51	90.88/96.41	–

Table 2
Evaluation of SOSD-H models across differential orders and FF++ training domains. Accuracy (%) and AUC (%) are reported for each differential order. *WD* (Within-Domain): average on all test domains (including training); *CD* (Cross-Domain): average on test domains excluding training. Bold highlights the best Accuracy and AUC per domain.

Order	DF		F2F		FS		FSh		NT		All FF++	
	WD	CD	WD	CD	WD	CD	WD	CD	WD	CD	WD	CD
0	61.40/71.59	53.03/64.70	62.84/72.10	55.58/65.74	59.30/61.58	50.40/52.24	60.24/65.13	51.65/56.70	65.62/70.11	61.68/65.03	67.96/72.43	–
1	65.84/74.56	58.43/68.34	66.28/77.78	58.93/72.43	59.32/64.21	50.08/55.45	60.10/66.18	51.63/58.09	69.56/76.95	64.65/72.05	78.68/87.01	–
2	64.42/**74.82**	56.60/**68.65**	**69.96**/80.26	**63.40**/75.58	60.52/64.14	51.33/55.29	58.94/62.10	50.33/52.99	71.84/76.92	67.40/72.03	81.82/91.04	–
3	66.28/73.80	58.75/67.41	68.22/78.06	61.75/73.05	**60.70**/66.77	**51.70**/58.61	60.52/64.84	52.90/56.85	71.96/76.21	67.83/71.53	**82.76/91.20**	–
4	65.44/74.03	57.80/67.69	69.04/**80.70**	62.73/**76.23**	60.12/65.19	51.03/56.71	60.96/65.55	53.83/57.87	**72.50/77.10**	**68.50/72.55**	82.70/90.62	–
5	**66.80**/71.87	**59.60**/65.08	65.32/78.37	58.13/73.43	60.12/**66.94**	50.95/**58.84**	**62.14**/65.70	**55.28**/57.98	69.58/74.17	65.23/69.14	80.54/88.76	–
6	64.60/73.22	57.05/66.81	68.02/79.48	61.40/74.77	59.48/64.95	50.20/56.38	61.82/**66.75**	54.98/**59.39**	67.92/73.36	63.93/68.70	79.60/88.98	–

Table 3
Evaluation of SOSD-W models across differential orders and FF++ training domains. Accuracy (%) and AUC (%) are reported for each differential order. *WD* (Within-Domain): average on all test domains (including training); *CD* (Cross-Domain): average on test domains excluding training. Bold highlights the best Accuracy and AUC per domain.

Order	DF		F2F		FS		FSh		NT		All FF++	
	WD	CD	WD	CD	WD	CD	WD	CD	WD	CD	WD	CD
0	62.54/71.96	53.88/65.05	61.00/66.48	54.00/59.04	**60.70**/68.20	**52.15**/60.57	63.12/**69.08**	55.13/**61.60**	61.52/66.26	56.30/59.82	69.62/76.45	–
1	64.48/**75.10**	56.45/**69.01**	**66.46**/75.91	**59.75**/70.21	59.96/67.46	50.68/59.43	61.86/68.46	53.53/60.81	68.46/**77.07**	63.38/**72.39**	**82.52/90.23**	–
2	**65.84**/72.72	**58.40**/66.15	65.58/**77.41**	58.13/**71.95**	59.76/**69.18**	50.65/**61.68**	61.28/64.12	53.40/55.75	**69.52**/74.16	64.65/68.97	81.74/89.88	–
3	64.38/71.81	56.30/64.97	64.74/76.64	57.20/71.10	60.20/66.89	51.30/58.89	61.36/65.95	54.00/58.33	68.70/74.71	64.63/69.82	81.16/88.49	–
4	65.44/71.56	58.00/64.71	64.26/76.94	56.75/71.50	59.94/63.21	51.13/54.34	**64.66**/66.72	**57.73**/59.02	69.24/75.14	**64.93**/70.41	77.98/82.59	–
5	63.10/70.21	55.48/63.21	65.14/76.14	58.03/70.50	60.28/65.07	51.53/56.58	62.96/66.55	55.95/59.08	68.56/74.10	63.98/69.29	81.44/87.88	–
6	64.00/71.44	56.65/64.73	63.20/74.68	55.33/68.64	59.76/62.91	51.05/53.99	64.04/66.92	57.53/59.43	67.48/72.87	62.95/68.06	81.48/85.28	–

with the residual texture statistics in Fig. 8, where spatial derivatives exhibit the clearest separation between real and fake samples.

Third, the optimal axis and order are forgery-dependent. Identity replacement forgeries (DF, FS) benefit more from temporal differencing, while expression-driven manipulations (F2F, NT) yield higher gains on spatial axes. Temporal operators often improve progressively with order and, in some cases they peak at higher values ($n \geq 5$). Spatial axes, by contrast, reach their maximum impact earlier, typically at $n = 1$ or $n = 3$. The same pattern is observed in the multi-domain configuration, where models are trained on the full set of FF++ manipulations. Compared to the RGB baseline, differencing yields average AUC gains between +5.3 and +18.8 points, and up to +14.8 in accuracy,

Table 4
Intra-dataset performance of SOSD models trained on Celeb-DF, WildDF, and DFDC under varying differential orders. Results are reported in terms of Accuracy (%) and AUC (%). Best results in bold.

Model		SOSD-T	SOSD-H	SOSD-W
Train	Order	Celeb-DF		
Celeb-DF	0	**96.71/99.54**	78.82/85.50	84.21/91.93
	1	93.68/98.39	**81.97/88.45**	82.24/89.85
	2	90.39/96.72	80.36/86.66	**84.47/92.39**
	3	92.37/97.24	76.45/83.64	80.53/89.16
	4	90.13/96.60	72.89/78.14	78.03/85.87
	5	89.87/96.15	72.37/78.69	76.18/83.77
	6	89.08/95.76	71.18/78.36	76.18/84.12
Train	Order	WildDF		
WildDF	0	**78.25/85.53**	69.92/77.01	74.92/81.26
	1	74.50/81.30	**75.17/82.12**	74.00/81.07
	2	73.75/80.13	72.50/79.21	**76.08/83.12**
	3	73.33/82.19	70.08/77.01	72.08/78.81
	4	73.75/80.41	67.92/73.89	70.75/76.36
	5	73.08/80.84	69.42/73.52	70.17/75.57
	6	73.25/79.66	68.75/73.46	71.00/77.24
Train	Order	DFDC		
DFDC	0	82.95/91.30	79.03/87.95	**81.20/89.45**
	1	83.51/91.13	**79.80/88.67**	80.04/88.72
	2	85.20/93.40	77.84/86.62	76.29/84.90
	3	**86.96/94.92**	75.50/84.10	74.00/81.95
	4	86.46/94.38	74.79/83.30	73.31/81.24
	5	86.79/94.47	73.45/81.37	71.39/79.19
	6	86.33/94.12	72.14/80.53	70.89/78.40

Table 5
Evaluation of SOMD models across differential orders and FF++ training domains. Accuracy (%) and AUC (%) are reported for each differential order. *WD* (Within-Domain): average on all test domains (including training); *CD* (Cross-Domain): average on test domains excluding training. Bold highlights the best Accuracy and AUC per domain.

Order		0	1	2	3	4	5	6
DF	WD	62.40/69.27	64.30/75.24	64.28/76.00	64.50/75.98	**64.92/76.57**	63.92/73.74	64.62/73.89
	CD	53.20/62.04	55.48/69.03	55.53/70.01	55.70/69.97	**56.23/70.72**	55.08/67.20	55.88/67.43
F2F	WD	63.18/70.93	63.36/72.04	62.54/76.04	62.90/74.95	62.78/**79.27**	63.52/74.19	**64.06**/74.68
	CD	54.43/64.03	54.58/65.51	53.43/70.12	53.95/68.79	53.93/**74.14**	54.60/67.81	**55.43**/68.50
FS	WD	60.02/59.17	59.84/64.08	60.24/63.83	60.78/67.34	60.48/66.86	60.54/63.28	**61.56/70.37**
	CD	50.35/49.17	50.10/55.12	50.55/54.81	51.15/59.20	50.70/58.60	50.90/54.25	**52.13/63.00**
FSh	WD	61.12/62.11	60.88/62.91	61.68/64.83	62.76/67.50	64.72/**68.56**	**66.16**/63.72	65.86/66.81
	CD	51.70/52.93	51.55/53.96	52.60/56.14	53.98/59.46	56.40/**60.83**	**58.18**/56.90	57.73/59.34
NT	WD	67.72/74.05	71.82/**78.80**	72.42/77.48	**73.00**/77.32	71.18/77.33	71.36/76.54	70.14/73.93
	CD	61.55/68.42	66.03/**73.78**	67.10/72.46	**67.75**/72.17	65.55/71.98	65.58/71.10	64.13/68.12
All FF++		85.38/89.98	90.68/94.67	**93.52**/95.01	91.62/94.07	92.54/**96.50**	91.98/95.32	92.14/93.93

depending on axis and order. Therefore, even under heterogeneous training conditions, differencing continues to enhance performance without compromising robustness.

Finally, we assess the generalization capacity of SOSD models beyond FF++ by replicating the same experimental setup on Celeb-DF, WildDF, and DFDC. The results, reported in Table 4 largely confirm the trends observed on FF++. Across all datasets, spatial differencing emerges as the most stable strategy: both H and W axes achieve stable peak performance at low to mid orders ($n = 1$–2), while higher orders consistently degrade performance. On the T-axis, effects are more dataset-specific: Celeb-DF and WildDF show no benefit from differencing, whereas DFDC responds positively to moderate temporal differencing, with a peak at $n = 3$. The axis- and dataset-specific nature of the observed performance trends confirms the absence of a universally optimal configuration, thereby motivating the fusion mechanisms introduced in the following sections to exploit complementary information across axes and differential orders.

4.3.3. Complementarity across directions

To assess whether combining directional differences improves detection and generalization, we now evaluate the performance of Single-Order Multi-Directional models across differential orders. Similarly to the SOSD analysis, the focus is on identifying performance trends over increasing difference orders within and across forgery domains. Results for FF++ domains are reported in Table 5. An overview comparing SOMD to SOSD baselines is also presented in Fig. 9, which displays the WD AUC across differential orders for each FF++ training domain. Overall, SOMD models consistently outperform the RGB baseline across most training domains and orders. On FF++, fusion amplifies the benefits observed in the best SOSD variants, particularly at mid-range orders ($n = 1$–4). For instance, order $n = 4$ achieves peak WD

Table 6
Intra-dataset performance of SOMD models trained on Celeb-DF, WildDF, and DFDC under varying differential orders. Results are reported in terms of Accuracy (%) and AUC (%). Best results in bold.

Order	Celeb-DF	WildDF	DFDC
0	**93.82/96.72**	**79.92**/82.12	87.29/93.97
1	90.26/96.24	79.00/**85.79**	87.36/94.58
2	89.61/92.76	76.42/77.05	87.63/94.95
3	88.68/92.38	75.83/78.97	**88.00/95.56**
4	85.26/89.11	74.17/79.25	87.04/95.29
5	84.34/87.63	75.92/76.93	87.29/95.02
6	83.03/88.68	74.50/77.48	87.03/94.54

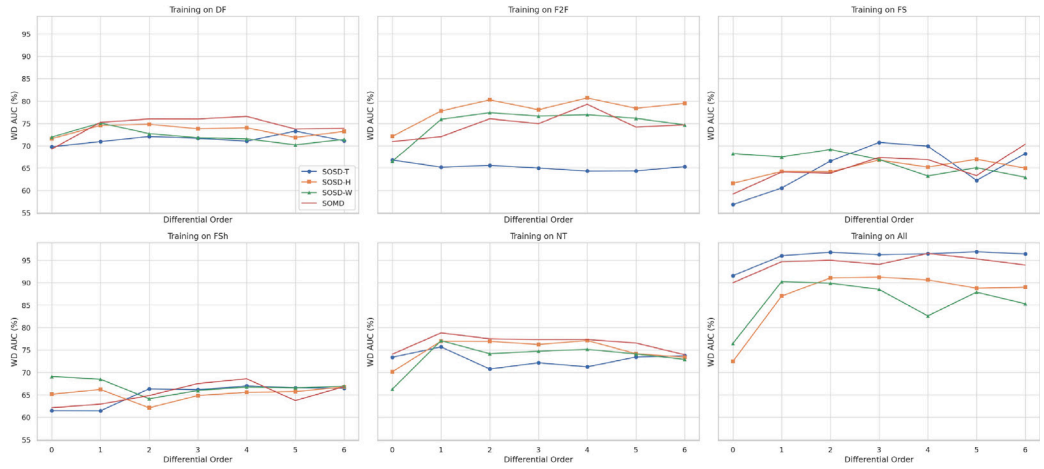

Fig. 9. Comparison of Within-Domain (WD) AUC across differential orders ($n = 0$–6) for four model configurations: SOSD-T, SOSD-H, SOSD-W, and SOMD. Each subplot corresponds to a different FF++ training domain (DF, F2F, FS, FSh, NT, and All FF++).

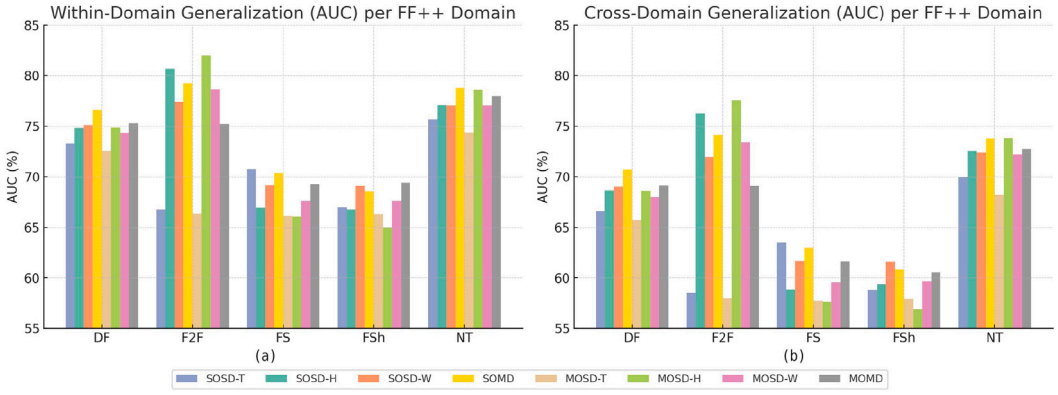

Fig. 10. Within- and cross-domain generalization performance (AUC %) of all our models on the FF++ dataset. (a) Within-domain results: average on all test domains (including training); (b) Cross-domain results: average on test domains excluding training.

Table 7
Evaluation of MOSD-T, -H, -W and MOMD models on FF++ training domains. Accuracy (%) and AUC (%) are reported for each differential order. *WD* (Within-Domain): average on all test domains (including training); *CD* (Cross-Domain): average on test domains excluding training.

Model		MOSD-T	MOSD-H	MOSD-W	MOMD
DF	WD	63.82/72.57	65.16/74.86	64.50/74.34	63.26/75.29
	CD	54.80/65.72	56.80/68.60	56.43/68.00	54.13/69.12
F2F	WD	60.92/66.34	68.72/81.96	64.82/78.64	61.94/75.22
	CD	51.40/57.98	61.53/77.56	56.88/73.41	52.70/69.08
FS	WD	59.64/66.14	62.36/66.06	60.06/67.57	60.30/69.26
	CD	49.90/57.70	53.55/57.63	50.80/59.55	50.50/61.63
FSh	WD	64.08/66.29	60.62/64.97	62.52/67.59	63.64/69.39
	CD	55.33/57.90	52.18/56.90	54.58/59.69	54.80/60.54
NT	WD	65.06/74.35	72.14/78.57	71.08/77.08	69.74/77.96
	CD	57.60/68.23	67.15/73.83	66.20/72.21	63.13/72.74
All FF++		93.46/98.08	85.56/93.13	85.74/93.21	94.36/98.22

AUC in four out of five training domains, with notable gains over RGB, all with improvements exceeding +6.45. Even the NT, which shows a slight improvement, has a gain of +3.28. Unlike SOSD models, where higher-order differencing often degraded performance, SOMD remains stable up to $n = 4$, with moderate drops at $n \geq 5$. This suggests that fusing derivatives along T, H, and W compensates for the sensitivity of individual axes to over-differencing. Fig. 9 further supports this interpretation. Across all training domains, SOMD exhibits smoother trends and more stable performance across orders compared to individual SOSD variants. While it does not consistently outperform the best SOSD model in every case — particularly for F2F and FS — its overall behavior remains competitive. This may be partly attributed to the simplicity of the fusion scheme, which was deliberately designed to be lightweight in order to isolate the effect of combining directions. More sophisticated strategies might better leverage directional complementarity and improve overall effectiveness. In the multi-domain training configuration, SOMD reaches its highest WD AUC of 96.50% at order $n = 4$, markedly outperforming the RGB baseline (89.98%). While the absolute peak is comparable to that of the strongest SOSD-T setup (96.90%), SOMD demonstrates greater stability across orders, maintaining high AUCs even when differencing increases. Finally, Table 6 reports intra-dataset results on Celeb-DF, WildDF, and DFDC. While the overall trends remain aligned with the single-axis models, SOMD exhibits distinct behaviors across datasets. On DFDC, fusion proves

Table 8
Intra-dataset performance of MOSD-T, -H, -W and MOMD models trained on Celeb-DF, WildDF, and DFDC. Results are reported in terms of Accuracy (%) and AUC (%).

Training dataset	MOSD-T	MOSD-H	MOSD-W	MOMD
Celeb-DF	93.06/98.21	80.26/87.62	83.55/92.53	89.61/95.50
WildDF	74.92/82.17	73.42/80.28	77.17/83.84	76.67/83.31
DFDC	89.39/96.14	81.16/89.99	79.13/88.11	89.98/96.74

Table 9
Cross-compression and brightness generalization for SOSD and SOMD models trained on FF++ (c23) and tested on c40, low-brightness (LB), and high-brightness (HB). Results are reported as Accuracy (%) and AUC (%). Best results in bold.

Model	SOSD-T	SOSD-H	SOSD-W	SOMD
Order		c23 → c40		
0	63.40/72.72	60.90/61.48	62.40/64.70	66.60/70.57
1	62.40/73.11	52.20/68.54	57.60/65.82	64.00/74.41
2	61.70/69.91	58.00/64.03	56.30/59.97	62.30/68.78
3	**67.00/74.79**	55.00/61.05	54.10/55.60	59.50/65.78
4	65.40/73.04	53.10/57.22	53.10/57.91	61.90/67.54
5	67.00/74.78	55.70/58.78	56.60/62.12	66.10/66.89
6	66.20/71.27	56.00/60.57	55.90/63.57	63.20/59.00
Order		c23 → c23-LB		
0	68.50/82.39	51.20/51.04	55.70/65.55	66.20/76.97
1	80.20/92.03	69.40/79.98	68.20/79.90	74.30/90.53
2	85.10/92.98	66.30/83.23	64.70/79.40	78.10/95.29
3	80.90/93.20	71.20/84.84	73.90/81.79	84.40/92.43
4	86.20/95.39	67.00/84.23	71.40/79.92	**84.80/95.73**
5	86.30/94.58	68.90/80.96	73.70/87.10	83.50/92.50
6	85.70/94.43	62.70/82.24	73.90/83.04	81.40/93.32
Order		c23 → c23-HB		
0	79.10/89.14	66.80/74.46	65.50/71.63	81.80/86.91
1	86.10/94.55	70.20/77.87	78.70/87.30	88.40/90.53
2	86.80/94.60	75.10/86.72	73.70/83.40	86.70/91.79
3	86.00/93.30	72.60/82.30	77.00/86.99	86.60/86.32
4	87.20/93.87	74.20/80.90	71.90/74.01	87.80/92.31
5	**88.30/95.41**	73.30/83.40	76.80/82.45	89.20/91.34
6	86.50/94.32	75.20/82.63	73.80/74.54	87.00/87.74

Table 10
Cross-compression and brightness generalization for MOSD and MOMD models trained on FF++ (c23) and tested on c40, low-brightness (LB), and high-brightness (HB). Results are reported as Accuracy (%) and AUC (%). Best results in bold.

Model	c40	c23-LB	c23-HB
MOSD-T	**67.70/76.65**	85.40/96.39	89.60/96.66
MOSD-H	56.70/64.58	72.00/88.07	77.30/87.11
MOSD-W	57.00/63.05	73.00/86.76	77.60/89.60
MOMD	63.70/69.38	**86.50/97.16**	**92.20/97.19**

particularly effective: SOMD not only reaches the top AUC values but also shows a stable performance across all orders. Conversely, on Celeb-DF and WildDF, fusion offers no substantial gain. The WD AUC curves of SOMD largely mirror those of SOSD-T and SOSD-W, with slight advantages only at specific orders. Overall, the benefits of directional fusion appear more pronounced under high domain variability. In contrast, for more homogeneous or axis-sensitive datasets, combining all directions may reduce the impact of the most informative axis, leading to diminished returns.

4.3.4. Complementarity across orders

Previous Sections showed that the optimal differential order (n) varies across axes and training domains, thus requiring a prior assumption that may not generalize. To address this limitation, we evaluate two architectures that aggregate information across multiple orders: Multi-Order Single-Direction, which fuses orders along a fixed axis, and Multi-Order Multi-Direction, which combines all orders and directions. The goal is to assess whether ensemble fusion of differential scales can replace manual order selection and offer greater robustness under distribution shifts. Results across FF++ (Table 7) and external datasets (Table 8), supported by Fig. 10, reveal the following insights:

- Multi-order aggregation provides moderate gains, especially on degraded or low-performing domains like FS and FSh. Improvements are more visible for spatial axes, particularly MOSD-H, which achieves 77.56% AUC on F2F (CD), outperforming all other configurations in that setting.
- Temporal fusion (MOSD-T) is less reliable. In cross-domain settings, it often degrades performance. This is likely due to inconsistency across temporal orders: single-order models can exploit the best differential level, while MOSD-T must also process noisy or detrimental signals (e.g., $n \geq 5$), which may dilute useful information.
- MOMD exhibits a distinct trade-off. While it aggregates all differential orders and axes into a unified representation, this broad coverage does not consistently translate into superior performance. On domains such as F2F and NT, it underperforms compared to more targeted models such as MOSD-H, suggesting that indiscriminate fusion may lead to interference when some residual components are less informative. Nevertheless, MOMD maintains competitive performance across all domains and avoids major failures, making it a pragmatic fallback without axis- or order-level optimization.

Instead, within-domain performance is generally more homogeneous across configurations (Fig. 10a), suggesting that the advantage of multi-scale or multi-axis fusion emerges primarily when facing distributional shifts. External benchmarks offer a final confirmation (Table 8).

Table 11
Comparison of state-of-the-art models and our proposed methods in terms of backbone architecture, pre-trained dataset, number of parameters, FLOPs per video clip, and model size.

Model	Backbone	Pre-trained Dataset	Params	FLOPs	Size
I3D [56]	3D Inception	Kinetics-400	12.29 M	27.88 G	98.6 MB
SlowFast [57]	slowfast	Kinetics-400	33.65 M	25.43 G	269.8 MB
RCN [58]	2D VGG11	ImageNet	15.70 M	120.35 G	583.3 MB
X3D [59]	x3d	Kinetics-400	2.98 M	5.08 G	24.3 MB
VTN [60]	VIT	Kinetics-400	112.82 M	270.70 G	905.7 MB
TAM [61]	2D ResNet50	Kinetics-400	24.78 M	66.13 G	198.5 MB
Timesformer [62]	VIT	Kinetics-600	121.10 M	380.05 G	970.3 MB
STIL [39]	SCNet50	ImageNet	22.69 M	76.09 G	182.6 MB
FreqNet [34]	–	–	1.7M	31.57 G	22.4 MB
HiFE [14]	–	–	47.55 M	180.59 G	363 MB
CLIP-ViT-L/14 [35]	CLIP-ViT-L/14 (Encoder)	WebImageText (WIT)	303 M	1250 G	1.2 GB
SOSD-best (Ours)	2D ResNet18	ImageNet	11.18 M	29.18 G	89.5 MB
SOMD-best (Ours)			33.53 M	87.53 G	134.4 MB
MOSD-best (Ours)			67.07 M	175.06 G	268.8 MB
MOMD (Ours)			201.20 M	525.17 G	806.4 MB

Table 12
AUC comparison of our model with the state-of-the-art models on the FF++ dataset. We select the best results for our models among the SOSD model, SOMD model, and MOSD model based on the average AUC. The training strategies include single-domain training (DF, F2F, FS, FSh, NT) and multi-domain training (All FF++).

Model	DF		F2F		FS		FSh		NT		All FF++
	WD	CD	WD	CD	WD	CD	WD	CD	WD	CD	WD
I3D	70.98	63.87	68.74	61.73	64.59	56.14	70.36	63.21	72.16	67.00	85.46
SlowFast	68.88	62.18	53.27	50.71	73.07	66.48	60.97	59.30	49.48	49.23	83.44
RCN	64.38	56.87	62.23	55.32	55.80	46.32	64.65	56.72	65.65	60.77	76.44
X3D	71.39	64.25	68.12	60.44	73.67	67.13	72.84	66.17	68.99	62.36	89.51
VTN	73.24	66.59	67.23	59.34	60.95	51.36	66.06	57.66	70.66	64.93	89.62
TAM	65.75	57.76	66.79	59.16	52.38	41.61	60.99	56.56	71.50	66.43	79.27
Timesformer	72.36	65.47	67.39	59.58	62.05	52.77	68.55	60.76	69.80	64.11	88.21
STIL	71.25	64.18	67.40	59.40	59.98	50.04	62.71	53.62	75.27	69.56	96.96
FreqNet	66.46	59.16	60.65	52.57	45.49	34.58	62.32	54.75	64.40	60.89	71.32
HiFE	69.42	61.79	71.29	64.21	61.62	52.03	61.29	51.63	75.44	69.74	96.97
CLIP-ViT-L/14	86.00	82.51	89.20	86.62	83.07	78.84	83.65	79.60	85.14	81.61	**98.27**
baseline-best	71.96	65.05	72.10	65.74	68.20	60.57	69.08	61.60	73.37	67.43	91.56
SOSD-best (Ours)	75.10	69.01	80.70	76.23	70.76	63.48	68.46	60.81	77.10	72.55	96.90
SOMD-best (Ours)	76.57	70.72	79.27	74.14	70.37	63.00	68.56	60.83	78.80	73.78	96.50
MOSD-best (Ours)	74.86	68.60	81.96	77.56	67.57	59.55	67.59	59.69	78.57	73.83	98.08
MOMD (Ours)	75.29	69.12	75.22	69.08	69.26	61.63	69.39	60.54	77.96	72.74	98.22

Spatial MOSD variants (MOSD-H, MOSD-W) generally retain competitive performance, especially on Celeb-DF and DFDC, where AUC scores exceed 92%. However, gains over the corresponding SOSD models remain modest. In some cases, performance slightly degrades, particularly on WildDF, suggesting that spatial multi-order aggregation is not uniformly effective across datasets. Temporal fusion (MOSD-T), while unstable across domains, remains competitive within each dataset. Finally, MOMD maintains strong and balanced performance, confirming the trend observed on FF++ and reinforcing the model's suitability as a general-purpose fallback when directional cues and residual structures vary or cannot be reliably estimated in advance. In summary, multi-order fusion is beneficial, but its effect depends on the axis and the dataset. The most consistent gains come from directional fusion: SOMD regularly ranks among the top performers across FF++ domains and external benchmarks, supporting the hypothesis that forgery cues are not isotropically distributed and that directional complementarity plays a key role in robust detection. However, when such order-level optimization is not feasible, MOMD remains a stable alternative for its accuracy and broader applicability.

4.4. Evaluation under real-world distortions

A critical aspect for the practical deployment of deepfake detectors is their robustness against real-world distortions such as compression and illumination changes. To evaluate these aspects, we conducted two dedicated experiments. Firstly, we trained our models on the lightly compressed FF++ (c23) dataset and tested them on the heavily compressed counterpart (c40). As shown in Tables 9–10, all model variants experience a significant performance degradation, with decreases reaching up to 30 percentage points in AUC. This magnitude of degradation is in line with prior work [63], confirming that cross-compression generalization is a systemic limitation in deepfake detection rather than a weakness specific to differential modeling. Then, we further assessed the robustness of the models under illumination variations, by testing on low- and high-brightness versions of FF++ (c23). In this case, performance consistently remained high across all architectures, with no significant degradation observed. Overall, compression artifacts emerge as the principal bottleneck for generalization, while illumination shifts exert negligible influence. Incorporating compression perturbations during training represents a straightforward yet essential step, which should be integrated with broader domain adaptation strategies to mitigate the impact of codec-induced distortions and move toward truly robust deployment.

4.5. Comparative experiment

We conclude with a comparative assessment against eleven deepfake detectors selected to represent a diverse set of temporal and spatial modeling strategies, including 3D convolutional networks, 2D+temporal hybrids, frequency-domain architectures, semantic-level multimodal models, and transformer-based architectures (Table 11). Although originally developed for generic video understanding, these models serve as strong task-agnostic baselines. All comparative methods were trained from scratch using the same data as our approach, processing the raw

Table 13
Cross-Data evaluation from FF++ to other datasets. Acc(%) and AUC(%) are reported. The training set is DF-FF++.

Model	Intra-Data	Cross-Data			Avg.	Cross-DataAvg.
	DF-FF++	Celeb-DF	WildDF	DFDC		
I3D	97.00/99.42	62.24/73.25	52.58/54.04	56.23/59.19	67.01/71.48	56.92/62.16
SlowFast	88.70/95.69	61.84/65.27	55.75/59.51	51.61/51.02	64.48/67.87	56.40/58.60
RCN	90.10/94.46	60.53/64.58	53.83/55.03	54.34/55.65	64.70/67.43	56.23/58.42
X3D	99.00/99.95	49.87/59.92	54.50/54.79	54.76/62.14	64.53/69.20	53.04/58.95
VTN	98.50/99.85	70.26/83.22	62.75/71.84	57.03/63.66	72.14/79.64	63.35/72.91
TAM	90.80/97.71	56.32/64.36	48.58/52.27	55.44/56.74	62.79/67.77	53.45/57.79
Timesformer	98.90/99.92	68.68/81.74	63.17/66.16	57.95/61.25	72.18/77.27	63.27/69.72
STIL	97.80/99.53	55.00/61.44	56.00/65.22	54.14/55.32	65.74/70.38	55.05/60.66
FreqNet	89.20/95.68	53.68/56.17	50.17/48.74	56.05/60.43	62.27/65.25	53.30/55.11
HiFE	99.40/99.93	60.79/78.51	61.83/70.36	55.75/62.33	69.44/77.78	59.46/70.40
CLIP-ViT-L/14	99.40/99.97	77.37/86.73	76.67/85.73	65.15/76.21	79.65/87.16	73.06/82.89
SOSD-best (Ours)	95.60/99.04	47.63/44.76	54.58/55.79	53.78/53.64	62.90/63.31	52.00/51.40
SOMD-best (Ours)	99.70/99.98	49.21/53.05	58.25/58.99	56.18/55.38	65.84/66.85	54.55/55.81
MOSD-best (Ours)	98.60/99.90	53.03/58.95	57.83/59.10	53.85/54.98	65.83/68.23	54.90/57.68
MOMD (Ours)	99.80/99.99	54.74/59.61	60.33/68.46	55.61/54.82	67.62/70.72	56.89/60.96

3D video volumes (16 consecutive frames) without any differential modeling or 3D-to-2D transformation. As shown in Table 12, our 3D Differential Decomposition models consistently rank among the top performers in both the within- and cross-domain setting. SOMD and spatial MOSD configurations exhibit particularly stable behavior across manipulation types, maintaining accuracy close to that of transformer-based methods while requiring substantially fewer resources. In particular, as evident from the joint analysis of computational complexities and number of parameters shown in Table 11 and from the AUCs obtained on the FF++ dataset shown in Table 12, the transformer- and multimodal-based architectures (e.g., VTN, CLIP-ViR-L/14) achieve high generalization scores but at a high cost, ranging from 180M to 300M parameters and over 1000 GFLOPs per 16-frame input. In contrast, frequency-domain models such as FreqNet and HiFE are more compact but show limited consistency across datasets, highlighting that frequency cues alone may not guarantee robustness. Our 3D Differential Decomposition stands in between these extremes, requiring only 12M parameters and 13.6 GFLOPs, yet delivering stable performance across both intra- and cross-dataset evaluations. Finally, Table 13 extends the evaluation to cross-dataset generalization. While all models experience a drop in performance when tested on Celeb-DF, WildDF, and DFDC after training on DF-FF++, MOMD achieves a solid compromise between robustness and performance, reaching an average AUC of 60.96%. Although this remains below the top-performing transformer-based architectures (e.g., VTN at 72.91% and CLIP-ViR-L/14 at 82.891%), MOMD consistently ranks among the most stable non-attention-based approaches, maintaining competitive results across all targets and outperforming several 3D CNN baselines. Taken together, these findings suggest three practical design guidelines for differential-based detectors: (i) low-order spatial derivatives ($n = 1$–3) are the most transferable; (ii) directional fusion is more effective than multi-order aggregation when generalization is critical; and (iii) MOMD offers a robust fallback when no axis- or manipulation-specific information is available.

5. Conclusion

In this paper, we introduced a structured differential modeling paradigm that operates directly on the 3D spatio-temporal volume of deepfake videos. By applying directional finite differences along temporal and spatial axes, and projecting the resulting residuals into a compact 2D representation, our method enables a factorized and interpretable analysis of forgery traces. Beyond this transformation, we investigated two complementary fusion strategies: combining information from multiple derivative orders (multi-order fusion), and aggregating responses across distinct axes (multi-directional fusion). Our experiments showed that directional fusion (e.g., SOMD) achieves a strong trade-off between accuracy and efficiency, while joint fusion across axes and orders (MOMD) offers stable performance under severe domain shifts. Taken together, our findings deliver not only a competitive detection framework, but also actionable design principles for architectures that exploit the non-isotropic structure of deepfake artefacts.

CRediT authorship contribution statement

Jie Gao: Writing – original draft, Software, Methodology, Conceptualization. **Marco Micheletto:** Writing – original draft, Validation, Methodology. **Giulia Orrù:** Writing – original draft, Validation, Investigation. **Xiaoyi Feng:** Writing – review & editing, Supervision. **Gian Luca Marcialis:** Writing – review & editing, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work is partially supported by the China Scholarship Council (No. 202206290093), by SERICS (PE00000014) under the Italian Ministry of University and Research (MUR) National Recovery and Resilience Plan funded by the European Union - NextGenerationEU, and within the PRIN 2022 PNRR - BullyBuster 2 – the ongoing fight against bullying and cyberbullying with the help of artificial intelligence for the human wellbeing (CUP: P2022K39K8). The BullyBuster project has been included in the Global Top 100 list of AI projects addressing the 17 United Nations Strategic Development Goals by the International Research Center for Artificial Intelligence under the auspices of UNESCO.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.image.2026.117525.

Data availability

Data will be made available on request.

References

[1] M. Ivanovska, V. Struc, On the vulnerability of deepfake detectors to attacks generated by denoising diffusion models, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2024, pp. 1051–1060.

[2] J. Choi, T. Kim, Y. Jeong, S. Baek, J. Choi, Exploiting style latent flows for generalizing deepfake video detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024, pp. 1133–1143.

[3] Y. Xiao, Q. Yuan, K. Jiang, J. He, X. Jin, L. Zhang, Ediffsr: An efficient diffusion probabilistic model for remote sensing image super-resolution, IEEE Trans. Geosci. Remote Sens. 62 (2023) 1–14.

[4] A. AV, S. Das, A. Das, et al., Latent flow diffusion for deepfake video generation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024, pp. 3781–3790.

[5] A. Aghasanli, D. Kangin, P. Angelov, Interpretable-through-prototypes deepfake detection for diffusion models, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023, pp. 467–474.

[6] Z. Sha, Z. Li, N. Yu, Y. Zhang, De-fake: Detection and attribution of fake images generated by text-to-image generation models, in: Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security, 2023, pp. 3418–3432.

[7] D.S. Vahdati, T.D. Nguyen, A. Azizpour, M.C. Stamm, Beyond deepfake images: Detecting ai-generated videos, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024, pp. 4397–4408.

[8] J. Yi, C. Wang, J. Tao, C.Y. Zhang, C. Fan, Z. Tian, H. Ma, R. Fu, Scenefake: An initial dataset and benchmarks for scene fake audio detection, Pattern Recognit. 152 (2024) 110468.

[9] W. Zhou, Z. Yang, C. Chu, S. Li, R. Dabre, Y. Zhao, K. Tatsuya, Mos-fad: Improving fake audio detection via automatic mean opinion score prediction, in: ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP, IEEE, 2024, pp. 876–880.

[10] C. Peng, Z. Miao, D. Liu, N. Wang, R. Hu, X. Gao, Where deepfakes gaze at? spatial–temporal gaze inconsistency analysis for video face forgery detection, IEEE Trans. Inf. Forensics Secur. (2024).

[11] D.A. Coccomini, G.K. Zilos, G. Amato, R. Caldelli, F. Falchi, S. Papadopoulos, C. Gennaro, Mintime: Multi-identity size-invariant video deepfake detection, IEEE Trans. Inf. Forensics Secur. (2024).

[12] L. Dugan, D. Ippolito, A. Kirubarajan, S. Shi, C. Callison-Burch, Real or fake text?: Investigating human ability to detect boundaries between human-written and machine-generated text, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 37, 2023, pp. 12763–12771.

[13] T. Zhang, Deepfake generation and detection, a survey, Multimedia Tools Appl. 81 (5) (2022) 6259–6276.

[14] J. Gao, Z. Xia, G.L. Marcialis, C. Dang, J. Dai, X. Feng, Deepfake detection based on high-frequency enhancement network for highly compressed content, Expert Syst. Appl. 249 (2024) 123732.

[15] S. Dong, J. Wang, R. Ji, J. Liang, H. Fan, Z. Ge, Implicit identity leakage: The stumbling block to improving deepfake detection generalization, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 3994–4004.

[16] J. Gao, M. Micheletto, S. Concas G. Orrù, X. Feng, G.L. Marcialis, F. Roli, Texture and artifact decomposition for improving generalization in deep-learning-based deepfake detection, Eng. Appl. Artif. Intell. 133 (2024) 108450.

[17] C. Tan, Y. Zhao, S. Wei, G. Gu, P. Liu, Y. Wei, Rethinking the up-sampling operations in cnn-based generative network for generalizable deepfake detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024, pp. 28130–28139.

[18] D. Cozzolino, A. Rössler, J. Thies, M. Nießner, L. Verdoliva, Id-reveal: Identity-aware deepfake video detection, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 15108–15117.

[19] L. Bondi, P. Bestagini E. Daniele Cannas, S. Tubaro, Training strategies and data augmentations in cnn-based deepfake video detection, in: 2020 IEEE International Workshop on Information Forensics and Security, WIFS, 2020, pp. 1–6, http://dx.doi.org/10.1109/WIFS49906.2020.9360901.

[20] S. Concas, S.M. La Cava, G. Orrù, C. Cuccu, J. Gao, X. Feng, G.L. Marcialis, F. Roli, Analysis of score-level fusion rules for deepfake detection, Appl. Sci. 12 (15) (2022) 7365.

[21] Y. Yu, X. Zhao, R. Ni, S. Yang, Y. Zhao, A.C. Kot, Augmented multi-scale spatiotemporal inconsistency magnifier for generalized deepfake detection, IEEE Trans. Multimed. 25 (2023) 8487–8498.

[22] L. Lin, X. He, Y. Ju, X. Wang, F. Ding, S. Hu, Preserving fairness generalization in deepfake detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024, pp. 16815–16825.

[23] B. Liu, B. Liu, M. Ding, T. Zhu, X. Yu, Ti2net: temporal identity inconsistency network for deepfake detection, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2023, pp. 4691–4700.

[24] Z. Ba, Q. Liu, Z. Liu, S. Wu, F. Lin, L. Lu, K. Ren, Exposing the deception: Uncovering more forgery clues for deepfake detection, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 38, 2024, pp. 719–728.

[25] R.S. FADAAM, N.A. ATASOY, Classification of Alzheimer's disease using 2dcnn technology using magnetic resonance imaging, Imaging (MCI) 10 (7) (2023).

[26] E.K. Ghasrodashti, P. Adibi, H. Karshenas, H.B. Kashani, J. Chanussot, Multi-modal image classification based on convolutional network and attention-based hidden markov random field, IEEE Trans. Geosci. Remote Sens. (2025).

[27] T. Wang, K.P. Chow, Noise based deepfake detection via multi-head relative-interaction, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 37, 2023, pp. 14548–14556.

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

[29] L. Chen, Y. Zhang, Y. Song, L. Liu, J. 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, 2022, pp. 18710–18719.

[30] K. Sun, T. Yao, S. Chen, S. Ding, J. Li, R. Ji, Dual contrastive learning for general face forgery detection, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 36, 2022, pp. 2316–2324.

[31] B. Yu, X. Li, W. Li, J. Zhou, J. Lu, Discrepancy-aware meta-learning for zero-shot face manipulation detection, IEEE Trans. Image Process. (2023).

[32] L. Li, J. Bao, T. Zhang, H. Yang, D. Chen, F. Wen, B. Guo, Face x-ray for more general face forgery detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 5001–5010.

[33] Y. Nirkin, L. Wolf, Y. Keller, T. Hassner, Deepfake detection based on discrepancies between faces and their context, IEEE Trans. Pattern Anal. Mach. Intell. 44 (10) (2021) 6111–6121.

[34] C. Tan, Y. Zhao, S. Wei, G. Gu, P. Liu, Y. Wei, Frequency-aware deepfake detection: Improving generalizability through frequency space domain learning, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 38, 2024, pp. 5052–5060.

[35] A. Yermakov, J. Cech, J. Matas, Unlocking the hidden potential of clip in generalizable deepfake detection, 2025, arXiv preprint arXiv:2503.19683.

[36] J. Thies, M. Zollhöfer, M. Nießner, Deferred neural rendering: Image synthesis using neural textures, Acm Trans. Graph. (TOG) 38 (4) (2019) 1–12.

[37] C. Zhao, C. Wang, G. Hu, H. Chen, C. Liu, J. Tang, Istvt: interpretable spatial–temporal video transformer for deepfake detection, IEEE Trans. Inf. Forensics Secur. 18 (2023) 1335–1348.

[38] Y. Zheng, J. Bao, D. Chen, M. Zeng, F. Wen, Exploring temporal coherence for more general video face forgery detection, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 15044–15054.

[39] Z. Gu, Y. Chen, T. Yao, S. Ding, J. Li, F. Huang, L. Ma, Spatiotemporal inconsistency learning for deepfake video detection, in: Proceedings of the 29th ACM International Conference on Multimedia, 2021, pp. 3473–3481.

[40] Z. Gu, T. Yao, Y. Chen, S. Ding, L. Ma, Hierarchical contrastive inconsistency learning for deepfake video detection, European Conference on Computer Vision, Springer, 2022, pp. 596–613.

[41] Z. Gu, Y. Chen, T. Yao, S. Ding, J. Li, L. Ma, Delving into the local: Dynamic inconsistency learning for deepfake video detection, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 36, 2022, pp. 744–752.

[42] W. Lu, L. Liu, B. Zhang, J. Luo, X. Zhao, Y. Zhou, J. Huang, Detection of deepfake videos using long-distance attention, IEEE Trans. Neural Netw. Learn. Syst. (2023).

[43] Z. Chen, X. Liao, X. Wu, Y. Chen, Compressed deepfake video detection based on 3d spatiotemporal trajectories, in: 2024 Asia Pacific Signal and Information Processing Association Annual Summit and Conference, IEEE, 2024, pp. 1–8.

[44] G. Petmezas, V. Vanian, K. Konstantoudakis, E.E. Almaloglou, D. Zarpalas, Video deepfake detection using a hybrid cnn-lstm-transformer model for identity verification, Multimedia Tools Appl. (2025) 1–20.

[45] Y. Xiao, Q. Yuan, K. Jiang, X. Jin, J. He, L. Zhang, C. w. Lin, Local–global temporal difference learning for satellite video super-resolution, IEEE Trans. Circuits Syst. Video Technol. 34 (4) (2023) 2789–2802.

[46] T. Isobe, X. Jia, X. Tao, C. Li, R. Li, Y. Shi, J. Mu, H. Lu, Y.-W. Tai, Look back and forth: Video super-resolution with explicit temporal difference modeling, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 17411–17420.

[47] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, L. Van Gool, Temporal segment networks: Towards good practices for deep action recognition, European Conference on Computer Vision, Springer, 2016, pp. 20–36.

[48] L. Wang, Z. Tong, B. Ji, G. Wu, Tdn: Temporal difference networks for efficient action recognition, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 1895–1904.

[49] G. Lee, M. Kim, Deepfake detection using the rate of change between frames based on computer vision, Sensors 21 (21) (2021) 7367.

[50] Y. Xu, J. Liang, L. Sheng, X.-Y. Zhang, Learning spatiotemporal inconsistency via thumbnail layout for face deepfake detection, Int. J. Comput. Vis. 132 (12) (2024) 5663–5680.

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

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

[53] B. Zi, M. Chang, J. Chen, X. Ma, Y.-G. Jiang, Wilddeepfake: A challenging real-world dataset for deepfake detection, in: Proceedings of the 28th ACM International Conference on Multimedia, 2020, pp. 2382–2390.

[54] B. Dolhansky, R. Howes, B. Pflaum, N. Baram, C.C. Ferrer, The deepfake detection challenge (dfdc) preview dataset, 2019, arXiv preprint arXiv:1910.08854.

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

[56] J. Carreira, A. Zisserman, Quo vadis, action recognition? a new model and the kinetics dataset, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 6299–6308.

[57] C. Feichtenhofer, H. Fan, J. Malik, K. He, Slowfast networks for video recognition, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 6202–6211.

[58] O. De Lima, S. Franklin, S. Basu, B. Karwoski, A. George, Deepfake detection using spatiotemporal convolutional networks, 2020, arXiv preprint arXiv:2006.14749.

[59] C. Feichtenhofer, X3d: Expanding architectures for efficient video recognition, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 203–213.

[60] D. Neimark, O. Bar, M. Zohar, D. Asselmann, Video transformer network, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 3163–3172.

[61] Z. Liu, L. Wang, W. Wu, C. Qian, T. Lu, Tam: Temporal adaptive module for video recognition, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 13708–13718.

[62] G. Bertasius, H. Wang, L. Torresani, Is space–time attention all you need for video understanding?, in: ICML, Vol. 2, 2021, p. 4.

[63] C. Zhao, C. Wang, Z. Song, G. Hu, L. Wang, D. Miao, Multi-definition deepfake detection via semantics reduction and cross-domain training, Pattern Recognit. 163 (2025) 111469.