

When Deepfake Detection Meets Graph Neural Network: a Unified and Lightweight Learning Framework

Haoyu Liu*
Nanyang Technological University
Singapore
haoyu.liu@ntu.edu.sg

Chaoyu Gong*
Nanyang Technological University
Singapore
chaoyu.gong@ntu.edu.sg

Mengke He
Nanyang Technological University
Singapore
mengke.he@ntu.edu.sg

Jiate Li
University of Southern California
Los Angeles, USA
jiateli@usc.edu

Kai Han
The University of Hong Kong
Hong Kong SAR
kaihanx@hku.hk

Siqiang Luo[†]
Nanyang Technological University
Singapore
siqiang.luo@ntu.edu.sg

Abstract

The proliferation of generative video models has made detecting AI-generated and manipulated videos an urgent challenge. Existing detection approaches often fail to generalize across diverse manipulation types due to their reliance on isolated spatial, temporal, or spectral information, and typically require large models to perform well. This paper introduces SSTGNN, a lightweight Spatial-Spectral-Temporal Graph Neural Network framework that represents videos as structured graphs, enabling joint reasoning over spatial inconsistencies, temporal artifacts, and spectral distortions. SSTGNN incorporates learnable spectral filters and temporal differential modeling into a graph-based architecture, capturing subtle manipulation traces more effectively. Extensive experiments on diverse benchmark datasets demonstrate that SSTGNN not only achieves superior performance in both in-domain and cross-domain settings, but also offers strong robustness against unseen manipulations. Remarkably, SSTGNN accomplishes these results with up to 42.4× fewer parameters than state-of-the-art models, making it highly lightweight and scalable for real-world deployment.

Keywords

Deepfake Detection; Graph Neural Network; Lightweight Model

1 Introduction

The rapid advancement of generative video models, such as Sora [40] and Runway Gen-2 [44], has significantly lowered the barrier for producing realistic synthetic videos. While these technologies enable creative applications, they also raise serious societal concerns, particularly when misused to manipulate visual evidence in sensitive domains like politics, finance, and public safety [37]. In response, detecting AI-generated videos has become a critical task for preserving media integrity and public trust. Despite recent progress, the detection landscape remains challenging. Manipulated videos can exhibit diverse and subtle forms of tampering, including fine-grained appearance alterations, imperceptible frequency artifacts, or temporally incoherent frame transitions [51].

Motivation. As the boundaries between real and synthetic videos continue to blur, detecting manipulated content is no longer

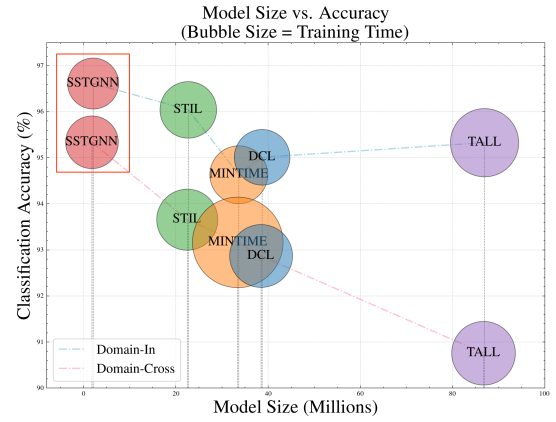


Figure 1: Comparison of accuracy, model size, and training cost for cross-domain and in-domain deepfake video classification. Our method, SSTGNN, achieves state-of-the-art detection performance while requiring up to 42× fewer parameters and reduced training time compared to baselines.

just a technical challenge but a fundamental requirement for safeguarding information integrity. Despite growing research attention, existing methods still struggle to generalize across diverse and unseen manipulation types, particularly those beyond facial forgeries. This calls for more versatile detection frameworks that can reason about manipulations at a deeper, more structural level. Therefore, our **first motivation** is to address this urgent and socially significant problem with a more generalizable and holistic detection method.

In this context, a critical question arises: how should we model the video content to enable such holistic understanding? Over the years, a wide range of deepfake detection methods have been proposed [51]. Among them, convolutional architectures such as ResNet [22] and Xception [9] have been widely adopted to capture spatial and temporal (i.e., visual-only) inconsistencies between real and manipulated content [2, 43]. In parallel, frequency-domain approaches [27, 34, 36] aim to detect high-frequency artifacts, up-sampling traces [32], and spectral anomalies characteristic of GAN-generated content [17]. However, these methods typically process spatial, temporal, and spectral information in isolation, or fuse them

*Both authors contributed equally to this research.

[†]Siqiang Luo is the corresponding author.

via simple late-stage aggregation, without capturing their complex interactions.

Moreover, these methods often rely on over-parameterized architectures to compensate for their limited representational capacity, resulting in large model sizes and reduced scalability. This limitation raises our **second motivation**: *the need for an unified and lightweight framework capturing interactions across spatial, temporal, and spectral domains*. In fact, graph-based representations offer a natural fit by treating a video as a spatiotemporal graph, where nodes represent local patch features and edges encode structured cross-domain relations. Graph Neural Networks (GNNs) built on such structures would learn expressive representations with significantly fewer parameters. Figure 1 illustrates quantitatively the trade-offs between accuracy, model size, and training cost across several state-of-the-art baselines. While some baselines achieve reasonable in-domain or cross-domain performance, they require larger models and more training time. In contrast, our GNN-based model achieves strong accuracy across domains with a significantly smaller model size (up to 42.4× fewer parameters) and lower training cost, as shown by the upper-left position and small marker size in both panels.

Technical Challenges. There are three major obstacles to advancing the use of GNNs for deepfake detection:

(1) How to construct expressive and efficient patch-level graphs? Unlike previous approaches that represent each frame or object as a node, our objective is to capture fine-grained visual inconsistencies at the patch level. This requires designing dense, high-resolution graphs where each node corresponds to a patch and edges encode both intra-frame and inter-frame relationships. The main challenge is to preserve local texture dependencies and temporal motion continuity, while maintaining tractable graph size and training efficiency.

(2) How to learn adaptive spectral filters on video graphs? Traditional frequency-based methods employ fixed transforms (e.g., DCT), which are limited in adapting to diverse manipulation artifacts. Our framework instead defines learnable spectral filters over the eigen-decomposition of the video graph Laplacian. The technical challenge is to ensure these filters are expressive enough to capture structural distortions, yet generalizable across videos with varying topologies, graph sizes, and manipulation types.

(3) How to encode non-local temporal and spatial artifacts in graphs? Subtle manipulations such as jitter, frame inconsistencies, or unnatural motion patterns are often temporally sparse and spatially dispersed. To detect these, we propose encoding frame-wise differences as negatively temporal edges between corresponding patches. The challenge lies in amplifying temporal inconsistencies without compromising spatial-spectral coherence or introducing instability into the graph learning process.

To address the above challenges, we summarize our key contribution as follows:

- We propose SSTGNN—Spatial Spectral Temporal Graph Neural Network—a unified graph-based framework that transforms a video into a structured graph integrating spatial, spectral, and temporal information. Unlike prior works that apply GNNs in a shallow or decoupled fashion, our approach leverages expressive graph construction to capture fine-grained manipulation artifacts

and enables joint reasoning over visual, structural, and dynamic inconsistencies.

- We introduce adaptive spectral filtering on graphs via learnable spectral functions defined over the graph Laplacian. This allows the model to discover frequency-domain anomalies in a fully data-driven manner, overcoming the limitations of fixed transforms and enhancing generalization to unseen manipulation types and diverse video domains.
- We encode both spatial and temporal differentials as negatively weighted edges within the graph, enabling the model to effectively capture subtle spatial anomalies and temporal inconsistencies (such as jitter or abrupt transitions) that are typically difficult to detect with standard frame-wise aggregation. This formulation enhances the model’s sensitivity to temporal inconsistencies while effectively integrating spatial and spectral information.

Remarkably, SSTGNN achieves rich representational capacity while maintaining a highly compact architecture. **It delivers state-of-the-art detection performance while requiring up to 42× fewer parameters** than existing models. Furthermore, as demonstrated in Table 5, SSTGNN achieves faster training and inference times as well as substantially reduced memory consumption compared to the baselines. These results collectively highlight the scalability, efficiency, and real-world practicality of our unified graph modeling framework, making it especially well-suited for deployment in resource-constrained environments.

2 Related Work

Spatial/Temporal detection methods. A large body of prior work on deepfake detection focuses on spatial and temporal cues, typically leveraging convolutional networks. Early approaches based on architectures like XceptionNet [43] laid the groundwork by training on shared datasets in an end-to-end manner. To improve robustness, some methods target fine-grained regions, such as lips [19], eye blinking [24], or head pose anomalies [35], which are often poorly synthesized. To capture temporal inconsistencies, recent models incorporate temporal modules. FTCN [59] combines CNNs with transformers to detect short- and long-term dynamics, while attention-based architectures extract spatial features for fusion with temporal modules [58]. With the rise of Vision Transformers (ViTs), many recent models adopt hybrid CNN-ViT designs for better temporal representation learning [13, 25, 53]. Beyond architectural innovations, several studies address generalization to unseen forgeries [38, 52] and privacy risks such as identity leakage [12, 23].

Spectral-based detection methods. Deepfake generation often introduces artifacts in the spectral domain due to GAN upsampling and convolutional processing. Methods such as [16, 34] detect such artifacts by analyzing the spectral distribution, offering strong generalization with low computational overhead. SPSTL [32] further combines spatial and phase spectrum features to identify subtle distortions, while AVFF [39] utilizes log-mel spectrograms to uncover audio-visual inconsistencies in manipulated content. Durall *et al.* [14] show that GAN-generated images exhibit spectral characteristics that deviate from those of natural images. While effective, most

spectral-based approaches process frequency-domain signals independently, lacking mechanisms for joint modeling across spatial, temporal, and spectral modalities.

GNNs in Computer Vision. GNNs are powerful paradigm for modeling structured data [20, 30, 31, 33]. They have been successfully applied in tasks such as object recognition, segmentation, and video understanding [26]. In the context of deepfake detection, recent works explore GNNs to model relational structures across space or time. Tan *et al.* [48] use graph learning over action units to model facial region interactions, and Wang *et al.* [51] construct spatial-frequency graphs with fixed spectral filters. However, many of these approaches treat GNNs as black-box modules, relying on fixed adjacency without explicitly designing graph topologies or integrating multi-domain signals. Anurag *et al.* [3] represent frames as nodes and build temporal edges via similarity, enabling simple yet effective video-level reasoning. In contrast, we construct a fine-grained spatiotemporal graph where each frame is a subgraph, and cross-frame connections capture long-term dynamics. This design allows unified learning of spatial, spectral, and temporal dependencies and offers improved generalizability and interpretability.

3 SSTGNN

In this section, we introduce SSTGNN, a unified learning framework for detecting AI-generated videos. By modeling each video as a graph, SSTGNN enables flexible and principled aggregation across spatial, spectral, and temporal domains, achieving robust detection performance with dramatically fewer parameters, faster training times, and lower memory overhead. Our key insights are summarized as follows:

- **Unified Graph Modeling.** Detecting AI-generated videos requires capturing both *intra-frame* visual artifacts (e.g., unnatural textures or distorted facial features) and *inter-frame* temporal inconsistencies. We model image patches as nodes and each frame as a subgraph, constructing spatial edges to enable fine-grained spatial representation beyond coarse frame-level modeling. To capture temporal dynamics, we add edges between corresponding patches across frames based on appearance similarity or change. This yields a spatiotemporal graph supporting flexible graph-component designs in the spatial, spectral, and temporal domains.
- **Learnable Spectral Filtering.** Building on the constructed video graph, we employ learnable spectral filters—originating from spectral GNNs—to enhance frequency-aware feature extraction. Unlike traditional methods that rely on fixed or hand-crafted transforms, our approach adapts spectral filters during training, enabling more expressive and task-relevant representations.
- **Spatial-Temporal Differentials.** Inspired by recent advances such as the Neighboring Pixel Relationships (NPR) approach, we introduce both spatial and temporal differential modeling via negatively weighted edges. This enables the model to explicitly capture subtle spatial anomalies (e.g., local pixel artifacts) as well as temporal inconsistencies (e.g., frame-to-frame jitter or unnatural transitions), thereby improving sensitivity to the unique patterns of AI-generated content.

By integrating these complementary cues, SSTGNN yields a comprehensive representation for video authenticity analysis. We then provide detailed descriptions of each component in the following.

3.1 Unified Graph Modeling

While GNNs have significant potential for AI-generated video detection, prior works often suffer from overly coarse graph designs, treating each frame as a single node and using GNNs as black-box temporal models [51]. Inspired by recent advances in patch-level image graph modeling [20, 21], we represent each frame as a subgraph of patch-level nodes. To capture temporal dynamics, we introduce temporal edges between corresponding patches across frames, forming a unified spatiotemporal graph for the video.

Formally, given a video clip $\mathcal{F} = \{F_1, F_2, \dots, F_T\}$ with T frames, we model each frame $F_t \in \mathbb{R}^{H \times W \times C}$ at time step t as a subgraph. Each frame is divided into N image patches of size $\ell \times \ell$, and their corresponding embeddings are extracted via a patch-wise encoder g_θ , which can be any pre-trained visual backbone. This yields a node set $\mathbf{V}^{(t)} = \{v_1^{(t)}, v_2^{(t)}, \dots, v_N^{(t)}\}$ and an embedding matrix $\mathbf{X}_t = \{x_1^{(t)}, x_2^{(t)}, \dots, x_N^{(t)}\} \in \mathbb{R}^{N \times d}$, where d is the embedding dimension, and $H \times W, C$ are the spatial and channel dimensions of the original frame. Without loss of generality, we also denote the nodes in $\mathbf{V}^{(t)}$ as $\{v_{i,j}^{(t)} | \forall 1 \leq i \leq N_H, \forall 1 \leq j \leq N_W\}$ to represent its coordinates, where $N_H = \lceil H/\ell \rceil$ and $N_W = \lceil W/\ell \rceil$.

To model the intra-frame structure, we construct edges between patch nodes based on visual similarity. For each feature matrix \mathbf{X}_t , each embedding is normalized as

$$\mathbf{X}_t^{\text{norm}} = \frac{\mathbf{X}_t}{\|\mathbf{X}_t\|_2 + \epsilon}, \quad \text{with } \epsilon = 10^{-4},$$

and the intra-frame edge weight matrix is defined as

$$\mathbf{A}_t = \text{SIM}(\mathbf{X}_t^{\text{norm}}, \mathbf{X}_t^{\text{norm}}),$$

where $(\mathbf{A}_t)_{i,j}$ denotes the similarity between nodes i and j in frame t , and thus the edge weight between $v_i^{(t)}$ and $v_j^{(t)}$. To balance computational efficiency and significance, a threshold parameter τ_s is used to prune edges with weights below τ_s . We denote the frame-level subgraph as $\mathbf{G}_t = (\mathbf{V}_t, \mathbf{A}_t)$.

To incorporate basic temporal patterns, we construct temporal edges between every two consecutive frames. At time step t , we bridge the subgraphs \mathbf{G}_t and \mathbf{G}_{t+1} by adding edges for similar node pairs at the same coordinates: $\mathbf{S}_t(v_i^{(t)}, v_i^{(t+1)})$

$$= \text{SIM}(\mathbf{A}_t(v_i^{(t)}), \mathbf{A}_{t+1}(v_i^{(t+1)})) + \text{SIM}(\mathbf{X}_t(v_i^{(t)}), \mathbf{X}_{t+1}(v_i^{(t+1)})),$$

where the first term measures structural similarity using adjacency matrices and the second considers embedding vector resemblance. If the similarity surpasses threshold τ_t , a temporal edge is defined as

$$\mathbf{S}_t(v_i^{(t)}, v_i^{(t+1)}) \geq \tau_t \implies \bar{\mathbf{A}}(v_i^{(t)}, v_i^{(t+1)}) = \mathbf{S}_t(v_i^{(t)}, v_i^{(t+1)}).$$

By constructing $\bar{\mathbf{A}}$ across all time steps, we form the complete video graph $\mathbf{G} = (\mathbf{V}, \mathbf{A}, \bar{\mathbf{A}}, \mathbf{X})$, where the node set $\mathbf{V} = \bigcup_{t=1}^T \mathbf{V}_t$ and the adjacency matrix \mathbf{A} combines spatial and temporal edges: $\mathbf{A} = \bigcup_{t=1}^T \mathbf{A}_t$. This unified graph captures both intra-frame (spatial) and inter-frame (temporal) correlations, supporting flexible and efficient spatial, spectral, and temporal reasoning.

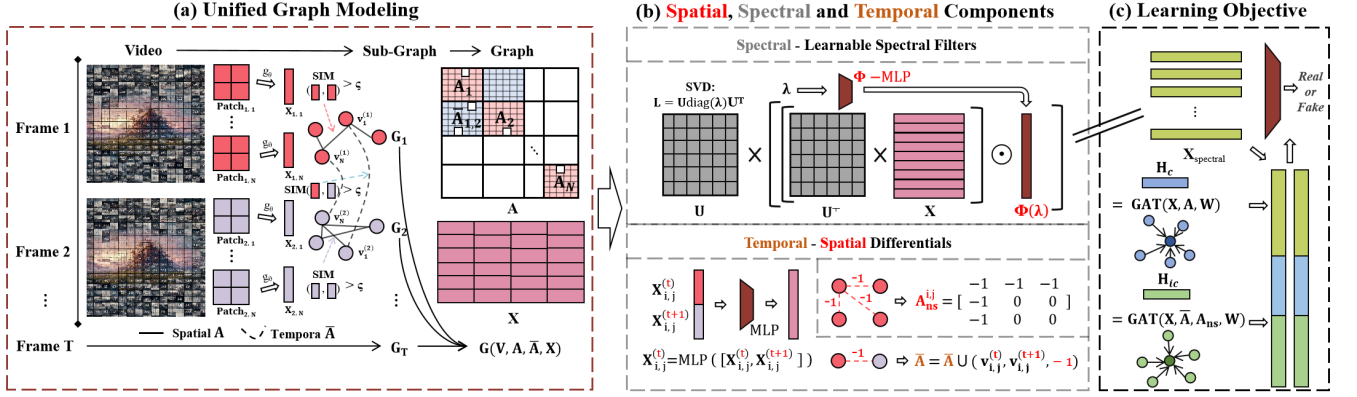


Figure 2: Flowchart of the proposed SSTGNN framework. (a) Each video frame is divided into patches, which are encoded as node embeddings to form intra-frame graphs $G_t = (V_t, A_t)$ based on patch similarity. Temporal edges connect corresponding patches across frames using both feature and structural similarity, resulting in a unified spatiotemporal graph. (b) A learnable spectral filter is applied over the Laplacian eigenbasis to extract frequency-domain representations. Temporal component includes concatenating embeddings and constructing inter-frame adjacency matrix A . Spatial component includes constructing sub-adjacency matrix $A_{ns}^{i,j}$ and learning features through two GATs operation. (c) The final features, $Z = [Z_{\text{spectral}}; Z_{\text{spatial}}]$, are concatenated and fed into a classifier for real/fake video prediction.

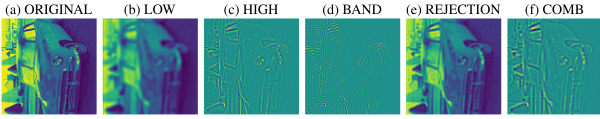


Figure 3: An illustration of an input image after our graph spectral filtering with specified function.

3.2 Spatial, Spectral and Temporal Components

Building upon our unified graph modeling, we are able to flexibly design specialized components tailored for artifact detection.

3.2.1 Learnable Spectral Filters. As motivated in Section 1, traditional frequency-based methods [51] apply fixed transforms (e.g., DCT) and hand-crafted filters to video frames. A filter is a function that specifies which frequency components to emphasize or suppress, e.g., high-pass filters highlight edges, while low-pass filters smooth textures. Such operations are common in image processing for enhancing or removing specific structures. In our case, we construct graphs from raw pixels and perform filtering in the graph spectral domain, where frequency reflects variation across the graph. This enables selective manipulation of signal components, akin to classical filters but grounded in the graph structure.

Moreover, traditional filters inject useful inductive bias, their static and hand-crafted nature may be suboptimal. We address this by learning *arbitrary filters* from graph signals, allowing the model to flexibly extract relevant frequency patterns for improved representation. Following standard GNN practice, we use the normalized graph Laplacian:

$$\mathbf{L} = \mathbf{I} - \mathbf{D}^{-1/2} \mathbf{A} \mathbf{D}^{-1/2},$$

where \mathbf{D} is the degree matrix and \mathbf{I} is the identity. To ensure symmetry, set $\mathbf{L} := (\mathbf{L} + \mathbf{L}^T)/2$ and perform eigendecomposition:

$$\mathbf{L} = \mathbf{U} \text{diag}(\boldsymbol{\lambda}) \mathbf{U}^T,$$

where $\boldsymbol{\lambda} \in \mathbb{R}^N$ and $\mathbf{U} \in \mathbb{R}^{N \times N}$ are the eigenvalues and eigenvectors. The learnable spectral filter is defined as

$$\phi(\boldsymbol{\lambda}) = f_{\text{MLP}}(\boldsymbol{\lambda}) \in \mathbb{R}^N,$$

enabling flexible, data-driven spectral manipulation.

Node embeddings \mathbf{X} are projected onto the spectral domain:

$$\tilde{\mathbf{X}} = \mathbf{U}^T \mathbf{X}.$$

The learned spectral filter modulates these coefficients, and the result is transformed back:

$$\mathbf{X}_{\text{spectral}} = \mathbf{U} \text{diag}(\phi(\boldsymbol{\lambda})) \tilde{\mathbf{X}}.$$

A global pooling operation yields aggregated spectral features:

$$\mathbf{Z}_{\text{spectral}} = \text{Pool}(\mathbf{X}_{\text{spectral}}) \in \mathbb{R}^d.$$

The intuition behind this approach is that representing raw pixels as a graph enables the model to learn arbitrary, task-specific spectral filters via parameterized functions of the eigenvalues $\boldsymbol{\lambda}$. This flexibility allows the model to selectively emphasize or suppress frequency components most relevant to the detection task. As shown in Figure 3, panels (b)–(e) illustrate the impact of applying low-pass, high-pass, band-pass, and band-rejection filters to $\boldsymbol{\lambda}$, thereby enhancing or attenuating various image features—such as smoothing textures, highlighting edges, or isolating intermediate-scale patterns. Panel (f) (“Comb”) demonstrates that aggregating multiple frequency bands yields richer and more detailed image reconstructions. Importantly, these spectral filters are not fixed; the filter function $\phi(\boldsymbol{\lambda})$, parameterized by MLP layers, supports fully adaptive and data-driven filtering tailored to the downstream task. This approach overcomes the limitations of hand-crafted filters and enables more expressive and effective feature extraction.

3.2.2 Spatial-Temporal Differentials as Negative Edges. Recall from Section 3.1 that we construct a unified graph using fundamental inter- and intra-frame edges by measuring similarity, providing initial *consistency* among patches. However, as shown in [18, 47], explicitly measuring spatial or temporal *inconsistency* is valuable for detecting generative bias. NPR [47] computes local pixel interdependence, and [18] exploits temporal differences across adjacent frames. Motivated by these findings, we incorporate negative edges for inconsistency message passing:

Spatial Differential. The original NPR technique focuses on local interdependence induced by up-sampling. Specifically, an ℓ_0 -NPR processes each pixel grid of size $\ell_0 \times \ell_0$ into a local differential version:

$$\begin{bmatrix} w_{1,1} & \cdots & w_{1,\ell_0} \\ \vdots & \ddots & \vdots \\ w_{\ell_0,1} & \cdots & w_{\ell_0,\ell_0} \end{bmatrix} \rightarrow \begin{bmatrix} \tilde{w}_{1,1} & \cdots & \tilde{w}_{1,\ell_0} \\ \vdots & \ddots & \vdots \\ \tilde{w}_{\ell_0,1} & \cdots & \tilde{w}_{\ell_0,\ell_0} \end{bmatrix},$$

where $\tilde{w}_{i,j} = w_{i,j} - w_{1,1}$, with $\ell_0 = 2$ shown to yield best performance [47]. This approach is effective in identifying fake regions.

We generalize this process to node embeddings by adding negative edges in our graph. Specifically for all $1 \leq i, j \leq \lceil N/\ell_0 \rceil$, $1 \leq t \leq T$, for node $v_{i_0,j_0}^{(t)}$ with $i_0 = i\ell_0$, $j_0 = j\ell_0$, we then construct a sub-adjacency matrix $\mathbf{A}_{\text{ns}}^{i,j} \in \mathbb{R}^{\ell_0 \times \ell_0}$ correspondingly:

$$\text{diag}(\mathbf{A}_{\text{ns}}^{i,j}) = \mathbf{1}; \quad \mathbf{A}_{\text{ns}}^{i,j}(v_{i_0,j_0}^{(t)}, :) = -\mathbf{1}; \quad \mathbf{A}_{\text{ns}}^{i,j}(:, v_{i_0}^{(t)}) = -\mathbf{1}.$$

This sub-adjacency matrix is used for message passing in graph learning (see Section 3.2.3). Intuitively, our approach captures local interdependence among nodes, applicable to pixels or higher-level embeddings. As proven in Theorem 3.1, with appropriate settings, ℓ_0 -NPR is a special case of our formulation after message passing.

THEOREM 3.1. *An ℓ_0 -NPR is equivalent to our spatial differential module when a small patch size is used and traditional message passing is adopted; specifically, by setting patch size $\ell = 1$ and using SGC-aggregation [54].*

PROOF. We aim to prove that the ℓ_0 -NPR technique is equivalent to our spatial differential module under the assumption that the patch size $\ell_0 = 1$ and the Simplified Graph Convolution (SGC) aggregation method is used. The original ℓ_0 -NPR technique applies a local differential operation to pixel grids of size $\ell_0 \times \ell_0$. Specifically, for each pixel $w_{i,j}$ in the grid, the transformation is performed by subtracting the value of the top-left pixel, $w_{1,1}$, such that:

$$\tilde{w}_{i,j} = w_{i,j} - w_{1,1}, \quad \forall 1 \leq i, j \leq \ell_0.$$

This operation can be expressed in matrix form as:

$$\mathbf{W} \rightarrow \tilde{\mathbf{W}} = \mathbf{W} - \mathbf{W}_{1,1}\mathbf{1},$$

where \mathbf{W} is the matrix of pixel values and $\mathbf{W}_{1,1}$ denotes the top-left pixel in the grid, and $\mathbf{1}$ is a matrix of ones of the same dimension as \mathbf{W} . This transformation effectively captures local interdependence among pixels and is particularly useful in detecting manipulated regions within images. Now, we generalize this ℓ_0 -NPR process to graph embeddings. For a node embedding $v_{i_0,j_0}^{(t)}$ with coordinates

$i_0 = i\ell_0$ and $j_0 = j\ell_0$ for $1 \leq i, j \leq \lceil N/\ell_0 \rceil$, we construct a sub-adjacency matrix $\mathbf{A}_{\text{ns}}^{i,j}$ as follows:

$$\text{diag}(\mathbf{A}_{\text{ns}}^{i,j}) = \mathbf{1}, \quad \mathbf{A}_{\text{ns}}^{i,j}(v_{i_0,j_0}^{(t)}, :) = -\mathbf{1}, \quad \mathbf{A}_{\text{ns}}^{i,j}(:, v_{i_0}^{(t)}) = -\mathbf{1}.$$

This matrix is used for message passing during the graph learning process, where each node receives messages from its neighboring nodes based on the adjacency structure. The inclusion of negative edges in the adjacency matrix effectively mimics the differential operation in the ℓ_0 -NPR method, subtracting the feature values of neighboring nodes, just as ℓ_0 -NPR subtracts the top-left pixel value from each pixel in the grid. In graph learning, the message passing rule is typically expressed using aggregation methods such as Simplified Graph Convolution (SGC), which updates a node's feature vector by aggregating the features of its neighbors. The message passing rule for a node v at iteration t is given by:

$$h_v^{(t+1)} = \mathbf{A}h_v^{(t)},$$

where \mathbf{A} is the adjacency matrix, and $h_v^{(t)}$ is the feature vector of node v at time step t . This aggregation allows each node to incorporate information from its neighbors to update its feature representation. In our case, the adjacency matrix $\mathbf{A}_{\text{ns}}^{i,j}$ used in message passing includes negative edges, which reduce the influence of neighboring nodes, replicating the effect of the local differential operation used in the ℓ_0 -NPR method. Specifically, the negative edges subtract the feature values of neighboring nodes, akin to how each pixel in the ℓ_0 -NPR grid is adjusted by subtracting the top-left pixel. Finally, when $\ell_0 = 1$, the patch size is reduced to a single element (or pixel), simplifying the message passing operation to a difference between the node's feature and its neighbors' features, much like how the ℓ_0 -NPR technique subtracts the value of the top-left pixel from all other pixels in the grid. Thus, we conclude that, when $\ell_0 = 1$ and SGC-aggregation is used, our spatial differential module operates equivalently to the ℓ_0 -NPR technique. Both methods apply a local differential operation to capture interdependence within a small neighborhood, whether in the pixel grid or the graph embedding space, and therefore, we establish the equivalence:

ℓ_0 -NPR \equiv Spatial Differential (with SGC-aggregation, and $\ell_0 = 1$). \square

This theorem demonstrates that our spatial differential module not only subsumes the ℓ_0 -NPR as a special case but also extends its capabilities, highlighting the effectiveness and versatility of our framework.

Temporal Differential. To capture temporal inconsistency, we enhance each node's representation by integrating information from adjacent frames. Intuitively, AI-generated videos may show subtle inconsistencies or unnatural transitions that are missed by spatial features alone. By explicitly modeling temporal relationships, we improve artifact detection. For node $v_{i,j}^{(t)}$ at $1 \leq t < T$, with raw embedding $\mathbf{X}_{i,j}^{(t)}$, we concatenate features from the next frame and apply an MLP:

$$\mathbf{X}_{i,j}^{(t)} = \text{MLP}(\text{Concat}[\mathbf{X}_{i,j}^{(t)}, \mathbf{X}_{i,j}^{(t+1)}]).$$

This enables implicit integration of temporal dynamics, facilitating detection of subtle frame-to-frame changes.

We further introduce negative edges into the inter-frame adjacency matrix:

$$\bar{A}(v_{i,j}^{(t)}, v_{i,j}^{(t+1)}) = -1,$$

for all $1 \leq i \leq N_H$, $1 \leq j \leq N_C$, and $1 \leq t < T$. These negative edges penalize abrupt changes in node features across consecutive time steps, thereby guiding the model’s attention to regions exhibiting pronounced temporal discrepancies. As demonstrated in Section 4.4, incorporating these negative temporal edges substantially enhances the overall detection performance.

By combining implicit feature integration with explicit relational modeling, our temporal differential module proves highly effective in revealing frame-to-frame inconsistencies that are characteristic of synthesized or manipulated videos, thereby significantly improving detection accuracy.

3.2.3 Conventional Learning Model. With all edges constructed, we apply the Graph Attention Network (GAT) [49] to our video graph due to its effectiveness in message passing. A single-layer GAT (X, A, W) applies a shared linear transformation $W \in \mathbb{R}^{d \times d}$ to each node feature $x_i \in \mathbb{R}^d$:

$$h_i = Wx_i.$$

For each edge (i, j) , it computes an unnormalized attention score:

$$e_{ij} = \text{LeakyReLU}(a^\top [h_i \| h_j]),$$

where a is a learnable vector and $\|$ denotes concatenation. Attention scores are masked by the adjacency matrix A and normalized:

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k \in N(i)} \exp(e_{ik})}.$$

Each node then aggregates neighbors’ features with a nonlinear activation σ :

$$h'_i = \sigma \left(\sum_{j \in N(i)} \alpha_{ij} h_j \right).$$

We employ two GAT layers to capture both consistency and inconsistency:

$$H_c = \text{GAT}(X, A, W),$$

$$H_{ic} = \text{GAT}(X, \bar{A} \cup A_{ns}, W),$$

where A encodes standard edges, \bar{A} complementary connections, and A_{ns} negative spatial edges. Final node representations are then concatenated and processed by an MLP layer:

$$Z_{\text{spatial}} = \text{MLP}(\text{Concat}[H_c, H_{ic}]).$$

This representation will then be combined with the spectral feature to support binary classification of video authenticity.

3.3 Learning Objective

Through the above steps, we have established a unified graph modeling and learning framework tailored for AI-generated video detection. In the final stage, all extracted representations are integrated for classification. Specifically, we concatenate learned embeddings from the spatial, spectral, and temporal domains:

$$Z = [Z_{\text{spatial}}; Z_{\text{spectral}}].$$

This overall representation is passed through an MLP to produce classification logits:

$$y = \text{MLP}(Z) \in \mathbb{R}^{N \times 2},$$

where label 0 denotes authentic videos and label 1 indicates AI-generated content. The model is trained end-to-end using cross-entropy loss to optimize classification. The full procedure of SSTGNN is outlined in Algorithm 1.

3.4 Procedure of SSTGNN

In Algorithm 1, the SSTGNN algorithm begins by dividing the video into N patches and extracting embeddings through the function g_θ . Next, intra-frame adjacency matrices A_t are computed using similarity measures between the normalized feature vectors of the frames. This adjacency matrix is then used to construct a frame-level graph G_t , which is subsequently bridged across consecutive frames using the temporal adjacency mechanism. After pruning edges based on predefined thresholds τ_s and τ_t , a unified graph G is formed. The spectral filter is learned using a multi-layer perceptron (MLP) applied to the eigenvalues of the graph Laplacian, yielding spectral features. The algorithm then constructs sub-adjacency matrices for spatial and temporal components, and negative edges are added to capture the differential effects. These processed features are then aggregated using Graph Attention Networks (GAT) to learn both spatial and spectral representations. Finally, the spatial and spectral features are concatenated to generate the final prediction, which is used for deepfake detection. The whole approach efficiently models the spatiotemporal information by utilizing a combination of graph-based learning and spectral filtering, significantly enhancing the detection performance across a variety of deepfake video manipulations.

Algorithm 1 SSTGNN model

Input: $\mathcal{F} \in \mathbb{R}^{T \times H \times W \times C}$, thresholds τ_s and τ_t

Output: Prediction results $y \in \mathbb{R}^{N \times 2}$

- 1: Divide frames into N patches and extract embeddings via g_θ
 - 2: Calculate intra-frame edge $A_t = \text{SIM}(X_t^{\text{norm}}, X_t^{\text{norm}})$, and construct frame-level subgraph $G_t = (V_t, A_t)$
 - 3: Bridge subgraphs G_t and G_{t+1} as $S_t(v_i^{(t)}, v_i^{(t+1)})$
 - 4: Form the graph $G = (V, A, \bar{A}, X)$ after applying τ_s and τ_t to prune edges
 - 5: Learn the spectral filter $\phi(\lambda) = f_{\text{MLP}}(\lambda) \in \mathbb{R}^N$, and yield spectral features Z_{spectral}
 - 6: Construct the sub-adjacency matrices A_{ns} and add negative edges
 - 7: Concatenate features from adjacent frames and add negative edge $\bar{A}(v_{i,j}^{(t)}, v_{i,j}^{(t+1)}) = -1$ into inter-frame adjacency matrix
 - 8: Apply GAT to video graph and learn the spatial features
 - 9: Concatenate learned features $Z = [Z_{\text{spatial}}; Z_{\text{spectral}}]$ and obtain the prediction results
-

4 Experiments

In this section, we explore the following research questions (RQs):

Table 1: Description and comparison of baselines. We highlight SSTGNN for its unified capture of spatial, spectral and temporal designs, and ultra light model sizes.

Methods	Publication	Features	Model Size
MINTIME [10]	IEEE TIFS 2024	Spatial, Temporal	33.64m
TNEM [45]	IEEE TIFS 2024	Spatial, Temporal	-
TALL [10]	ICCV 2023	Spatial, Temporal	86.91m
DCL [46]	AAAI 2022	Spatial	38.69m
SPLAFS [6]	IEEE TCSVT 2022	Spatial, Temporal	-
RECCE [5]	CVPR 2022	Spatial	-
STIL [18]	ACM MM 2021	Spatial, Temporal	22.69m
MultiAtt [57]	CVPR 2021	Spatial	-
FaceXray [28]	CVPR 2020	Spatial	-
XceptionNet [9]	CVPR 2017	Spatial	22.90m
SSTGNN	This paper	Spatial, Spectral, Temporal	2.05m

- **RQ1: In-domain Detection Performance.** How effective is SSTGNN at detecting fake videos when trained and tested on the same dataset?
- **RQ2: Cross-domain Generalization.** How well does SSTGNN generalize across different video generation domains, specifically (i) cross-method generalization, where SSTGNN is trained on one video generation method and tested on another, and (ii) cross-dataset generalization, evaluating SSTGNN’s performance across multiple datasets in one-to-many and many-to-many settings?
- **RQ3: Resource Allocation (Memory, Time, and Number of Parameters).** How lightweight is SSTGNN in terms of memory consumption, training time, and model size?
- **RQ4: Ablation Study.** What are the individual contributions of SSTGNN’s components, patch size, and pruning threshold to its overall performance?

Baselines. We compare SSTGNN¹ against 10 state-of-the-art baseline methods, which were selected based on their relevance and proven effectiveness in deepfake detection tasks. These baselines include various approaches, ranging from spatiotemporal analysis to graph-based and attention-based methods, as shown in Table 1. For open-sourced baselines, we run their official implementations using the hyperparameters specified in their respective papers. For baselines without publicly available code, we directly report the results from the original publications, ensuring consistent settings.

Datasets. We evaluate SSTGNN and the 10 baselines on a comprehensive range of datasets, as detailed in Table 7 in Appendix A.1. These datasets are widely used in the deepfake detection community. The FF++ dataset consists of four subsets, each generated with a distinct face manipulation technique: DeepFakes (DF), Face2Face (F2F), FaceSwap (FS), and NeuralTextures (NT). These subsets facilitate a systematic evaluation across different types of facial forgeries.

Experimental Setup. To address **RQ1**, we perform training and testing on the same dataset using six datasets: DF, F2F, FS, NT, CD (v1 and v2), and Wild-DF. For **RQ2**, we employ a three-pronged approach: (i) fine-grained cross-subset testing within the four FF++ subsets, training each model on one subset and testing

Table 2: In-domain performance comparison (AUC). Red marks best performance and blue marks indicate the runner-up. Rank denotes the average ranking of all methods.

Methods	NT	FS	F2F	DF	CD-V1	CD-V2	Wild-DF	Avg	Rank
MINTIME	99.3	98.6	98.79	92.15	97.52	95.48	81.46	94.64	9.00
TALL	99.05	97.89	97.69	94.45	96.79	94.97	86.11	95.50	8.00
MultiAtt	93.44	98.87	97.96	98.84	99.42	97.86	90.74	96.16	7.00
RECCE	93.63	98.82	98.06	99.65	99.94	96.81	92.02	96.13	6.00
SPLAFS	97.71	99.27	99.05	99.55	98.72	97.32	91.89	97.50	6.29
FaceXray	98.93	99.20	99.06	99.17	98.35	97.89	92.11	97.10	5.29
DCL	97.69	99.90	99.28	99.76	99.96	99.77	88.96	97.76	4.29
TNEM	98.06	99.90	99.33	99.98	99.59	98.51	92.63	98.14	4.14
STIL	98.72	99.91	99.86	99.99	99.90	99.86	92.36	98.80	2.86
SSTGNN	98.76	99.97	99.67	99.99	99.96	99.86	94.36	99.51	1.14

it on the other three; (ii) training a model on the full FF++ dataset and evaluating it on three external datasets: Wild-DF, CD-v2, and DFDC; and (iii) testing SSTGNN in a 3-to-3 setting, where training data consists of SEINE, SVD, and Pika, and test data includes Crafter, Gen2, and Lavie. For **RQ3**, all models are trained on the following datasets: DF, F2F, FS, NT, CD-v1, Wild-DF, SEINE, and SVD. We record both the peak and average GPU memory usage, as well as training time, to assess resource efficiency. Ablation studies for **RQ4** are conducted exclusively on the NT dataset, enabling us to evaluate the impact of individual components on overall performance.

Model Configuration. Our SSTGNN uses a modified ResNet-50 as the backbone, where we retain the network up to layer-2, followed by a global average pooling layer to generate compact feature representations. The model is trained using the Adam optimizer with a learning rate of 0.0001, a batch size of 16, and a total of 100 epochs. Performance is evaluated using two metrics: Area Under the ROC Curve (AUC) and Accuracy (Acc). All experiments are conducted on a machine with three NVIDIA RTX A5000 GPU (24GB), two Intel Xeon Gold 6238R CPUs (2.20GHz), and 200GB DDR4 RAM.

4.1 RQ1: In-domain Detection Performance

Table 2 compares our method with seven state-of-the-art deepfake detection approaches across seven datasets. Our method consistently ranks among the top performers, achieving the highest average AUC (99.51%) and the best overall rank (1.14) across all datasets, highlighting its superior in-domain performance and strong generalization. Specifically, our model achieves the best AUC on FS (99.97%), DF (99.99%), CD-V1 (99.96%), CD-V2 (99.86%), and Wild-DF (94.36%), while also securing second-best AUC on NT (98.76%) and F2F (99.67%). This demonstrates our model’s robustness across both canonical manipulations and more diverse, challenging settings (e.g., CD-V1/V2 and Wild-DF). Compared to recent methods like STIL and DCL, our approach not only matches or exceeds their performance on most datasets but also maintains full evaluation coverage, unlike some baselines with missing results. Moreover, earlier methods such as FaceXray and MultiAtt show significantly lower performance, especially on unconstrained datasets. Overall, the results demonstrate that SSTGNN offers a compelling balance of accuracy and reliability across various deepfake detection scenarios.

¹Code available via email to haoyu.liu@ntu.edu.sg.

Table 3: Cross-domain: 1-to-3 setting within four FF+ subsets. "-" denotes that the method is trained on that subset.

Methods	DF	F2F	FS	NT	Avg
MultiAtt	-	66.41	67.33	66.01	66.58
STIL	-	65.70	40.74	75.10	60.51
DCL	-	77.13	61.01	77.01	71.00
RECCE	-	70.66	74.29	67.34	72.49
SSTGNN	-	79.34	58.71	83.42	73.82
MultiAtt	73.04	-	65.10	71.88	67.00
STIL	76.63	-	51.24	73.30	67.39
DCL	91.91	-	59.58	66.72	67.25
RECCE	75.99	-	62.71	72.32	69.27
SSTGNN	82.38	-	62.90	76.67	72.16
MultiAtt	82.33	66.11	-	63.31	74.40
STIL	38.20	49.60	-	49.45	56.83
DCL	74.80	69.75	-	52.60	65.19
RECCE	82.39	64.44	-	56.70	75.84
SSTGNN	69.99	66.48	-	55.79	72.56
MultiAtt	74.56	80.61	66.07	-	72.88
STIL	93.28	89.17	51.24	-	73.48
DCL	91.23	52.13	79.31	-	73.61
RECCE	78.83	80.89	63.70	-	74.27
SSTGNN	93.58	95.19	64.07	-	84.28

4.2 RQ2: Cross-domain Generalization

We assess cross-domain generalization under comprehensive experimental settings and provide explainable analyses to showcase the superior generalizability of SSTGNN.

1-to-3 setting. Table 3 presents the cross-method generalization results for various deepfake detection approaches, where each model is trained on a specific forgery type and tested on other types within the FF++ dataset. This evaluation highlights each method’s ability to generalize across different manipulation techniques. Our method consistently outperforms competing approaches across all training settings. When trained on DF, our method achieves the highest average accuracy (73.82%), and obtains the best results on both NT (83.42%) and F2F (79.34%). Similarly, with F2F as the training type, our model again ranks first (72.16%), significantly outperforming others on NT (76.67%). Under FS training, RECCE achieves the highest average (75.84%), closely followed by our method (72.56%), with strong results on F2F (66.48%) and DF (69.99%). In the NT-trained setting, our method attains a remarkable 84.28% average accuracy—the best overall—demonstrating superior generalization to unseen videos such as F2F (95.19%).

3-to-3 setting. To further evaluate robustness in realistic scenarios, we adopt a 3-to-3 setting, where the model is trained on SEINE, SVD, and Pika, and tested on Crafter, Gen2, and Lavie (Table 4). This setup simulates diverse generative patterns in both training and testing. Our method consistently achieves strong and balanced performance across all test datasets. On Crafter, it attains the highest accuracy (95.83%) and AUC (99.62%), outperforming STIL and DCL. On Gen2, it achieves the top AUC (98.96%) and competitive accuracy (93.06%). While baselines like STIL and DCL may perform well on individual

Table 4: Cross-domain: 3-to-3 setting from training on SVD, Seine and Pika, and testing on Crafter, Gen2 and Lavie.

Methods	Crafter		Gen2		Lavie	
	Acc	AUC	Acc	AUC	Acc	AUC
STIL	95.74	97.12	97.39	98.23	87.40	90.80
DCL	93.07	98.78	93.63	98.88	90.97	93.90
MINTIME	92.77	96.57	92.37	97.56	85.00	87.84
TALL	91.29	97.64	92.19	98.67	84.81	87.45
SSTGNN	95.83	99.62	93.06	98.96	89.72	94.48

domains, they tend to experience notable drops in performance when evaluated across other domains.

Remark. Under comprehensive cross-domain elevations, we highlight SSTGNN for its out-breaking generation ability, benefiting from our unified graph representation design.

Explainable Analyses. To assess the generalization capabilities of our method, we conduct t-SNE visualization and Principal Component Analysis (PCA) on both SSTGNN and the runner-up method, STIL, using the FF++ dataset.

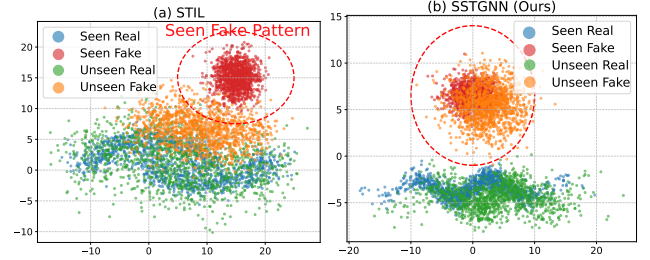


Figure 4: t-SNE visualization of the learned features from (a) STIL and (b) SSTGNN on FF++. We highlight that SSTGNN achieves a more natural separation between real and fake samples, demonstrating improved generalization to unseens.

Figure 4 showcases the t-SNE embeddings of feature representations. In panel (a), the features from STIL reveal a significant clustering of seen fake samples, forming a dense cluster dominated by fake samples, as highlighted by the red ellipse. However, other samples—particularly unseen fakes—are more diffusely distributed and not distinctly separated from real samples. This suggests that STIL tends to overfit to the known fake patterns, restricting its ability to generalize effectively to unseen manipulations. In contrast, panel (b) illustrates that SSTGNN produces a more balanced feature space, where seen fake, unseen fake, and real samples are better separated, albeit with some natural overlap. This indicates that SSTGNN excels at capturing subtle forgery characteristics, thus improving its generalization across diverse fake video types.

To quantitatively analyze the information content within the learned feature representations, we apply Principal Component Analysis (PCA), examining the explained variance across principal components, as shown in Figure 5. For STIL, over 90% of the variance is concentrated within the top three components, suggesting a

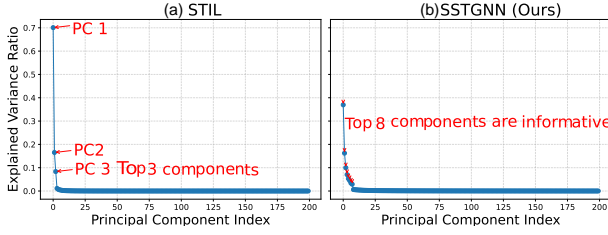


Figure 5: PCA-based analysis of feature space expressiveness on FF++. For STIL, over 90% of the variance is explained by only *three* principal components, reflecting a low-rank and limited representation. In comparison, SSTGNN spreads the variance across the first *eight* components, indicating a more expressive and information-rich feature space.

low-rank structure that limits its expressiveness and generalization potential. In contrast, SSTGNN distributes the variance more evenly across the first seven to eight components, indicating a much richer and more expressive feature space. This broader distribution of variance enables SSTGNN to capture diverse and previously unseen forgery patterns, reducing the risk of overfitting to shallow, dataset-specific features. Such a structure enhances the model’s ability to generalize robustly on various deepfake detection tasks.

4.3 RQ3: Resource Allocation

To evaluate the practical efficiency of our framework, we benchmark SSTGNN against strong baselines on FF++ in terms of training time, inference speed, and GPU memory usage. As observed in Table 5, SSTGNN consistently achieves the highest efficiency across all measured metrics. It delivers the lowest training times (4698–4746 seconds), offering a relative reduction of 27.4% compared to the second-best method. During inference, SSTGNN processes samples in 10–11 milliseconds—enabling real-time or low-latency deployment with 20.0% and 9.1% speed improvements. Furthermore, it maintains the lowest GPU memory consumption (8540–8546 MB), achieving over 119% relative savings. These results highlight the computational efficiency and scalability of our approach: SSTGNN achieves state-of-the-art detection performance with minimal resource demands, making it well suited for both academic research and real-world, resource-constrained environments.

4.4 RQ4: Ablation Study

To better understand the contribution of each component and the impact of hyper-parameters, we conduct an ablation study divided into two parts. The first part isolates the effect of three core components, i.e., spatial features, spectral features, and the NPR module. The second part explores how patch size and the threshold τ influence performance, offering insight into the robustness and sensitivity of the model under different configurations.

Spatial, Spectral and Temporal Components. Table 6 presents an ablation study quantifying the individual and combined contributions of the spatial, spectral, and temporal components in our framework. Removing any one component leads to a noticeable drop in both accuracy and AUC, confirming the complementary benefits of each design. Notably, removing the spectral module degrades

Table 5: Resource allocation comparison on FF++ (DF, F2F). Training time(s), inference time(ms), and memory (MB). The row Saved indicates the relative reduction in resource consumption of SSTGNN compared to the second-best method.

Methods	Training		Inference		Memory	
	DF	F2F	DF	F2F	DF	F2F
STIL	5986	6043	12	12	18758	18756
DCL	6236	6348	12	13	24312	24334
MINTIME	14760	15326	23	21	22964	23067
TALL	7210	7367	14	14	28351	28439
SSTGNN	4698	4746	10	11	8546	8540
Saved (%)	27.4	27.4	20.0	9.1	119.6	119.7

Table 6: Ablation: spatial, spectral and temporal components.

Temporal	Spectral	Spatial	Accuracy	AUC
	✓	✓	95.08	98.30
✓		✓	93.78	97.33
✓	✓		92.90	98.03
✓	✓	✓	95.63	98.76

the performance at the most, with near 2% and 1% in Accuracy and AUC respectively. This greatly demonstrates the effectiveness of our learnable filter designs. Meanwhile, the negative edges involved also help the model to enhance its performance. These findings validate that integrating spatial, spectral, and temporal cues is critical for maximizing detection performance in deepfake video analysis.

Patch size and graph thresholds. Figure 6 examines how different patch sizes and threshold τ values affect the model’s performance. Among patch sizes, 32×32 achieves the best balance between detail and contextual information, reaching 95.63% accuracy and 98.76% AUC. Interestingly, increasing to 56×56 further improves AUC slightly (99.03%) but offers no significant gain in accuracy, suggesting a saturation point. As for the threshold τ , which controls decision confidence, the best performance is observed at $\tau = 0.6$, aligning with the main setting of model. Both too high ($\tau = 0.8$) and too low ($\tau = 0.2$) degrade accuracy, indicating that overly strict or overly permissive thresholds can harm decision reliability. These results demonstrate that the model is relatively robust but still benefits from careful tuning.

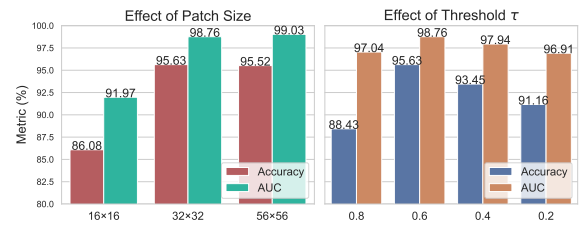


Figure 6: Ablation: patch size and graph thresholds τ .

5 Conclusion

This paper presents SSTGNN, a unified framework that seamlessly integrates spatial, temporal, and spectral information for robust deepfake detection. By transforming videos into expressive graph structures and employing learnable spectral filters alongside temporal differential cues, SSTGNN effectively captures subtle manipulation artifacts that traditional methods often overlook. The empirical results, demonstrated across a diverse range of datasets, highlight SSTGNN's superior generalization and adaptability, achieving significantly fewer parameters than existing baselines.

Beyond its technical innovations, SSTGNN marks a paradigm shift in deepfake detection by modeling videos as dynamic, interconnected graphs. This approach not only improves detection accuracy but also provides interpretable insights into the relational dynamics of multimedia data. The interpretability, combined with SSTGNN's lightweight architecture, makes it an ideal solution for real-world deployment, where the unpredictability of forgery types demands both robustness and efficiency.

References

- [1] 2024. Zeroscope-v2-xl. https://huggingface.co/cerspense/zeroscope_v2_XL. Accessed: 2025-06-26.
- [2] Darius Afchar, Vincent Nozick, Junichi Yamagishi, and Isao Echizen. 2018. Mesonet: a compact facial video forgery detection network. In *2018 IEEE international workshop on information forensics and security (WIFS)*. IEEE, 1–7.
- [3] Anurag Arnab, Chen Sun, and Cordelia Schmid. 2021. Unified graph structured models for video understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 8117–8126.
- [4] Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. 2023. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127* (2023).
- [5] Junyi Cao, Chao Ma, Taiping Yao, Shen Chen, Shouhong Ding, and Xiaokang Yang. 2022. End-to-end reconstruction-classification learning for face forgery detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 4113–4122.
- [6] Han Chen, Yuzhen Lin, Bin Li, and Shunquan Tan. 2022. Learning features of intra-consistency and inter-diversity: Keys toward generalizable deepfake detection. *IEEE Transactions on Circuits and Systems for Video Technology* 33, 3 (2022), 1468–1480.
- [7] Haoxin Chen, Menghan Xia, Yingqing He, Yong Zhang, Xiaodong Cun, Shaoshu Yang, Jinbo Xing, Yaofang Liu, Qifeng Chen, Xintao Wang, et al. 2023. Videocrafter1: Open diffusion models for high-quality video generation. *arXiv preprint arXiv:2310.19512* (2023).
- [8] Xinyuan Chen, Yaohui Wang, Lingjun Zhang, Shaobin Zhuang, Xin Ma, Jiashuo Yu, Yali Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. 2023. Seine: Short-to-long video diffusion model for generative transition and prediction. In *The Twelfth International Conference on Learning Representations*.
- [9] François Chollet. 2017. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1251–1258.
- [10] Davide Alessandro Coccomini, Giorgos Kordopatis Zilos, Giuseppe Amato, Roberto Caldelli, Fabrizio Falchi, Symeon Papadopoulos, and Claudio Gennaro. 2024. MINTIME: multi-identity size-invariant video deepfake detection. *IEEE Transactions on Information Forensics and Security* (2024).
- [11] Brian Dolhansky, Russ Howes, Ben Pfau, Nicole Baram, and Cristian Canton Ferrer. 2019. The deepfake detection challenge (dfdc) preview dataset. *arXiv preprint arXiv:1910.08854* (2019).
- [12] Shichao Dong, Jin Wang, Renhe Ji, Jiajun Liang, Haoqiang Fan, and Zheng Ge. 2023. Implicit identity leakage: The stumbling block to improving deepfake detection generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 3994–4004.
- [13] Xiaoyi Dong, Jianmin Bao, Dongdong Chen, Ting Zhang, Weiming Zhang, Nenghai Yu, Dong Chen, Fang Wen, and Baining Guo. 2022. Protecting celebrities from deepfake with identity consistency transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 9468–9478.
- [14] Ricard Durall, Margret Keuper, and Janis Keuper. 2020. Watch your up-convolution: Cnn based generative deep neural networks are failing to reproduce spectral distributions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 7890–7899.
- [15] Patrick Esser, Johnathan Chiu, Parmida Atighehchian, Jonathan Granskog, and Anastasis Germanidis. 2023. Structure and content-guided video synthesis with diffusion models. In *Proceedings of the IEEE/CVF international conference on computer vision*. 7346–7356.
- [16] Joel Frank, Thorsten Eisenhofer, Lea Schönherr, Asja Fischer, Dorothea Kolossa, and Thorsten Holz. 2020. Leveraging frequency analysis for deep fake image recognition. In *International conference on machine learning*. PMLR, 3247–3258.
- [17] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. *Advances in neural information processing systems* 27 (2014).
- [18] Zhihao Gu, Yang Chen, Taiping Yao, Shouhong Ding, Jilin Li, Feiyue Huang, and Lizhuang Ma. 2021. Spatiotemporal inconsistency learning for deepfake video detection. In *Proceedings of the 29th ACM international conference on multimedia*. 3473–3481.
- [19] Alexandros Haliassos, Konstantinos Vougioukas, Stavros Petridis, and Maja Pantic. 2021. Lips don't lie: A generalisable and robust approach to face forgery detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 5039–5049.
- [20] Kai Han, Yunhe Wang, Jianyuan Guo, Yehui Tang, and Enhua Wu. 2022. Vision gnn: An image is worth graph of nodes. *Advances in neural information processing systems* 35 (2022), 8291–8303.
- [21] Yan Han, Peihao Wang, Souvik Kundu, Ying Ding, and Zhangyang Wang. 2023. Vision hgnn: An image is more than a graph of nodes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 19878–19888.
- [22] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.
- [23] Baojin Huang, Zhongyuan Wang, Jifan Yang, Jiaxin Ai, Qin Zou, Qian Wang, and Dengpan Ye. 2023. Implicit identity driven deepfake face swapping detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 4490–4499.
- [24] Taekhyun Jung, Sangwon Kim, and Keecheon Kim. 2020. Deepvision: Deepfakes detection using human eye blinking pattern. *IEEE Access* 8 (2020), 83144–83154.
- [25] Bachir Kaddar, Sid Ahmed Fezza, Wassim Hamidouche, Zahid Akhtar, and Abdenour Hadid. 2021. HCiT: Deepfake video detection using a hybrid model of CNN features and vision transformer. In *2021 International Conference on Visual Communications and Image Processing (VCIP)*. IEEE, 1–5.
- [26] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).
- [27] Jiaming Li, Hongtao Xie, Jiahong Li, Zhongyuan Wang, and Yongdong Zhang. 2021. Frequency-aware discriminative feature learning supervised by single-center loss for face forgery detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 6458–6467.
- [28] Lingzhi Li, Jianmin Bao, Ting Zhang, Hao Yang, Dong Chen, Fang Wen, and Baining Guo. 2020. Face x-ray for more general face forgery detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 5001–5010.
- [29] Yuezun Li, Xin Yang, Pu Sun, Honggang Qi, and Siwei Lyu. 2020. Celeb-df: A large-scale challenging dataset for deepfake forensics. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 3207–3216.
- [30] Ningyi Liao, Siqiang Luo, Xiang Li, and Jieming Shi. 2023. LD2: Scalable Heterophilous Graph Neural Network with Decoupled Embeddings. In *Advances in Neural Information Processing Systems* (New Orleans, LA, USA), A. Oh, T. Neumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (Eds.), Vol. 36. Curran Associates, Inc., 10197–10209. https://proceedings.neurips.cc/paper_files/paper/2023/file/206191b9b7349e2743d98d855dec9e58-Paper-Conference.pdf
- [31] Ningyi Liao, Dingheng Mo, Siqiang Luo, Xiang Li, and Pengcheng Yin. 2022. SCARA: Scalable Graph Neural Networks with Feature-Oriented Optimization. In *Proceedings of the VLDB Endowment* (Sydney, Australia), Vol. 15. VLDB Endowment, 3240–3248. doi:10.14778/3551793.3551866
- [32] Honggu Liu, Xiaodan Li, Wenbo Zhou, Yuefeng Chen, Yuan He, Hui Xue, Weiming Zhang, and Nenghai Yu. 2021. Spatial-phase shallow learning: rethinking face forgery detection in frequency domain. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 772–781.
- [33] Haoyu Liu, Ningyi Liao, and Siqiang Luo. 2025. SIGMA: An Efficient Heterophilous Graph Neural Network with Fast Global Aggregation. In *2025 IEEE 41st International Conference on Data Engineering (ICDE)*. IEEE Computer Society, 1924–1937.
- [34] Yuchen Luo, Yong Zhang, Junchi Yan, and Wei Liu. 2021. Generalizing face forgery detection with high-frequency features. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 16317–16326.
- [35] Kevin Lutz and Robert Bassett. 2021. Deepfake detection with inconsistent head poses: Reproducibility and analysis. *arXiv preprint arXiv:2108.12715* (2021).
- [36] Iacopo Masi, Aditya Killekar, Royston Marian Mascarenhas, Shenoy Pratik Gurudatt, and Wael AbdAlmageed. 2020. Two-branch recurrent network for isolating deepfakes in videos. In *Computer vision—ECCV 2020: 16th European conference, glasgow, UK, August 23–28, 2020, proceedings, part VII 16*. Springer, 667–684.

- [37] F. Nazren. 2024. *Lee Hsien Loong urges S'poreans not to believe deepfake video of him 'promoting' investment product*. <https://mothership.sg/2024/06/lee-hsien-loong-deepfake-scam/> Accessed: 2024-07-12.
- [38] Utkarsh Ojha, Yuheng Li, and Yong Jae Lee. 2023. Towards universal fake image detectors that generalize across generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 24480–24489.
- [39] Trevine Oorloff, Surya Koppiseti, Nicolò Bonettini, Divyraj Solanki, Ben Colman, Yaser Yacoob, Ali Shahriyari, and Gaurav Bharaj. 2024. Avff: Audio-visual feature fusion for video deepfake detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 27102–27112.
- [40] OpenAI. 2024. *Sora*. <https://openai.com/index/sora/> Accessed: 2025-05-25.
- [41] Xiangyu Peng, Zangwei Zheng, Chenhui Shen, Tom Young, Xinying Guo, Binluo Wang, Hang Xu, Hongxin Liu, Mingyan Jiang, Wenjun Li, et al. 2025. Open-sora 2.0: Training a commercial-level video generation model in 200 k. *arXiv preprint arXiv:2503.09642* (2025).
- [42] Pika art. 2022. <https://pika.art/>. Accessed: 2025-06-26.
- [43] Andreas Rossler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Nießner. 2019. Faceforensics++: Learning to detect manipulated facial images. In *Proceedings of the IEEE/CVF international conference on computer vision*. 1–11.
- [44] Runway. 2024. *Gen-2: The next step forward for generative AI*. <https://research.runwayml.com/gen2> Accessed: 2024-05-25.
- [45] Huimin She, Yongjian Hu, Beibei Liu, Jicheng Li, and Chang-Tsun Li. 2024. Using Graph Neural Networks to Improve Generalization Capability of the Models for Deepfake Detection. *IEEE Transactions on Information Forensics and Security* (2024).
- [46] Ke Sun, Taiping Yao, Shen Chen, Shouhong Ding, Jilin Li, and Rongrong Ji. 2022. Dual contrastive learning for general face forgery detection. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 36. 2316–2324.
- [47] Chuangchuan Tan, Yao Zhao, Shikui Wei, Guanghua Gu, Ping Liu, and Yunchao Wei. 2024. 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*. 28130–28139.
- [48] Lingfeng Tan, Yunhong Wang, Junfu Wang, Liang Yang, Xunxun Chen, and Yuanfang Guo. 2023. Deepfake video detection via facial action dependencies estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 37. 5276–5284.
- [49] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. In *International Conference on Learning Representations*.
- [50] Yaohui Wang, Xinyuan Chen, Xin Ma, Shangchen Zhou, Ziqi Huang, Yi Wang, Ceyuan Yang, Yinan He, Jiahuo Yu, Peiqing Yang, et al. 2025. Lavie: High-quality video generation with cascaded latent diffusion models. *International Journal of Computer Vision* 133, 5 (2025), 3059–3078.
- [51] Yuan Wang, Kun Yu, Chen Chen, Xiyuan Hu, and Silong Peng. 2023. Dynamic graph learning with content-guided spatial-frequency relation reasoning for deepfake detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 7278–7287.
- [52] Zhendong Wang, Jianmin Bao, Wengang Zhou, Weilun Wang, and Houqiang Li. 2023. Altfreezing for more general video face forgery detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 4129–4138.
- [53] Deressa Wodajo and Solomon Atnafu. 2021. Deepfake video detection using convolutional vision transformer. *arXiv preprint arXiv:2102.11126* (2021).
- [54] Felix Wu, Amauri Souza, Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Weinberger. 2019. Simplifying graph convolutional networks. In *International conference on machine learning*. Pmlr, 6861–6871.
- [55] Haiyang Xu, Qinghao Ye, Xuan Wu, Ming Yan, Yuan Miao, Jiabo Ye, Guohai Xu, Anwen Hu, Yaya Shi, Guangwei Xu, et al. 2023. Youku-mplug: A 10 million large-scale chinese video-language dataset for pre-training and benchmarks. *arXiv preprint arXiv:2306.04362* (2023).
- [56] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. 2016. Msr-vtt: A large video description dataset for bridging video and language. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 5288–5296.
- [57] Hanqing Zhao, Wenbo Zhou, Dongdong Chen, Tianyi Wei, Weiming Zhang, and Nenghai Yu. 2021. Multi-attentional deepfake detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2185–2194.
- [58] Xiaohui Zhao, Yang Yu, Rongrong Ni, and Yao Zhao. 2022. Exploring complementarity of global and local spatiotemporal information for fake face video detection. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2884–2888.
- [59] Yinglin Zheng, Jianmin Bao, Dong Chen, Ming Zeng, and Fang Wen. 2021. Exploring temporal coherence for more general video face forgery detection. In *Proceedings of the IEEE/CVF international conference on computer vision*. 15044–15054.
- [60] Bojia Zi, Minghao Chang, Jingjing Chen, Xingjun Ma, and Yu-Gang Jiang. 2020. Wilddeepfake: A challenging real-world dataset for deepfake detection. In *Proceedings of the 28th ACM international conference on multimedia*. 2382–2390.

A Appendix

A.1 Detail of Datasets

Table 7 summarizes the video datasets used in our study, covering a wide spectrum of both real and fake sources. These datasets span traditional deepfake benchmarks, modern generative video datasets, and large-scale real-world corpora. For facial forgery detection, we include classic datasets such as FF++ and Celeb-DF, which are based on video-to-video (V2V) or image-to-video (I2V) manipulations like DeepFakes and FaceSwap, typically with moderate resolution and short clip lengths. Wild-DF and DFDC further introduce more realistic variations in lighting, occlusion, and subject diversity. Recent large-scale generative datasets—including SEINE, SVD, and Pika—reflect the latest advancements in image- and text-driven video generation, with high-quality outputs and broad content coverage. These are used extensively for training in multi-source setups due to their volume and diversity. To evaluate generalization to unseen generative patterns, we adopt newer open-world benchmarks such as OpenSora, ZeroScope, Crafter, Gen2, and Lavie. These datasets are built from emerging text-to-video (T2V) models and differ in frame rates, resolutions, and generation styles. In particular, our 3-to-3 evaluation trains on SEINE, SVD, and Pika, and tests on Crafter, Gen2, and Lavie. For authentic real-world reference, we use MSR-VTT and Youku-mPLUG—two diverse, large-scale datasets originally designed for video retrieval and understanding—serving as clean real samples to benchmark false positives and open-set robustness. Altogether, our dataset suite consists of 13 datasets covering V2V, I2V, and T2V paradigms, ranging from 2 to 120 seconds in duration, 240p to 2K in resolution, and including over 500,000 fake videos and 20,000+ real clips. This broad coverage enables robust training and evaluation across both in-domain and cross-domain scenarios.

Table 7: Video Datasets Overview.

Video source	Type	FPS	Length	Resolution	Size	
					Real	Fake
FF++ [43]	I2V & V2V	≤30	5-15s	≤1280×720	1,000	4,000
Celeb-DF (CD) [29]	V2V	≤30	13s	≤1280×720	590	5,639
Wild-DF [60]	V2V	≤25	5-15s	≤854×480	707	707
DFDC [11]	V2V	30	3-15s	≤1280×720	2,000	3,000
SEINE [8]	I2V	8	2-4s	1024×576	-	24,737
SVD [4]	I2V	8	4s	1280×720	-	149,026
Pika [42]	T2V & I2V	24	3s	1088×640	-	98,377
OpenSora [41]	T2V & I2V	24	3s	1088×640	-	177,410
ZeroScope [1]	T2V	8	3s	1024×576	-	133,169
Crafter [7]	T2V	8	4s	256×256	-	1,400
Gen2 [15]	T2V & I2V	24	4s	896×512	-	1,380
Lavie [50]	T2V	8	2s	1280×2048	-	1,400
MSR-VTT [56]	-	30	10-30s	320×240	10,000	-
Youku-mPLUG [55]	-	25-30	10-120s	≤1920×1080	10,000	-