Exposing and Mitigating Temporal Attack in Deepfake Video Detection

Zheyuan Gu[1]* **Minghao Shao**[2]* **Zhen Wang**[3] **Keyu Mao**[4] **Ailiang Lin**[4]
Mingyou Liang[5] **Mingkun Xu**[6] **Shijie Zhang**[1] **Hao Jiang**[1] **Yusong Wang**[7]†

[1]Peking University [2]New York University [3]Huzhou Normal University
[4]Institute of Science Tokyo [5]Tongji University
[6]Guangdong Institute of Intelligence Science and Technology
[7]Hangzhou Tuomai Information Technology

Abstract

While spatiotemporal deepfake detectors achieve high AUC, our experiments reveal their susceptibility to evasion attacks. These models tend to overfit on fragile temporal spectrum cues, rather than learning robust semantic causality. To mitigate this vulnerability, we propose SpInShield, a temporal spectral-invariant defense framework explicitly designed to decouple semantic motion from manipulatable spectral artifacts. We propose a learnable spectral adversary that dynamically synthesizes severe spectral deformations, simulating extreme attack scenarios. By employing a shortcut suppression optimization strategy, SpInShield compels the encoder to extract reliable forensic cues while purging unstable spectral statistics from the latent space. Experiments show that SpInShield obtains competitive performance on widely used datasets and outperforms the strongest baseline by 21.30 percentage points in AUC under simulated amplitude spectral attacks.

1 Introduction

The rise of hyper-realistic AI-generated faces (deepfakes) undermines digital media credibility [37]. This technology lowers the barrier for malicious actors to fabricate convincing videos and biometric bypass and political disinformation campaigns [10, 48]. These capabilities can destabilize digital media's evidentiary value, demanding robust, automated countermeasures. In response, the community has deployed various detection mechanisms, mainly leveraging deep learning models to distinguish synthetic videos from authentic videos [50]. Because dynamic videos contain temporal cues distinct from static images, modern defenses adopt data-driven architectures that mine spatiotemporal representations, making them the primary defense against digital manipulation [63, 28, 39].

Existing deepfake detection methods fall into static frame-level analysis and dynamic video-level modeling [50, 36]. The first category extracts individual frame artifacts, treating video inputs as independent images. They use CNN backbones or spectral transformations to capture pixel-level anomalies and spatial frequency fingerprints, relying solely on intra-frame features [44, 33, 30]. The second category exploits temporal correlations via spatiotemporal coherence, overlooked by static analyzers. 3D-CNNs [49] or Video Transformers [2] analyze motion continuity and inter-frame relationships, revealing spatiotemporal inconsistencies that expose manipulation [70, 64, 39, 28].

Despite high performance on standard benchmarks, these spatiotemporal models risk overfitting specific temporal spectral patterns. Figure 1 shows that selectively suppressing specific temporal frequency bands with a temporal notch filter can substantially reduce detection performance, measured

*Equal contribution.
†Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

by the area under the receiver operating characteristic curve (AUC), indicating sensitivity to these spectral components. Whether implicitly modeled by neural networks or explicitly extracted via Discrete Fourier Transform (DFT), this rigid dependence on fixed temporal patterns makes the detectors vulnerable. The influence on the temporal spectrum, such as the aforementioned suppression, spectral tilt, and codec distortion, is common in the real world, arising from temporal filtering, global denoising, and video re-encoding. In such scenarios, an adversary can easily employ various perturbations to modify these temporal spectrum patterns and evade detection, raising significant security concerns.

However, building robust detectors faces three challenges. First, separating malicious temporal spectrum artifacts from legitimate motion is complex. The temporal frequency of forgeries overlaps with genuine facial dynamics, such as micro-expressions or blinking. Naively suppressing specific frequency bands to avoid overfitting risks losing critical semantic information, reducing detection AUC on real videos. Second, modern architectures exhibit a strong inductive bias toward low-level statistics. Models like 3D-CNNs and Transformers are driven by optimization to find the easiest solution. Without explicit constraints, they naturally gravitate towards brittle temporal spectrum shortcuts rather than learning the complex, high-level semantic causality of video content. Third, capturing spectrum-invariant representations requires a shift in learning paradigm. The lack of ground-truth annotations on spectral robustness prevents direct supervision to penalize the model for relying on brittle features.

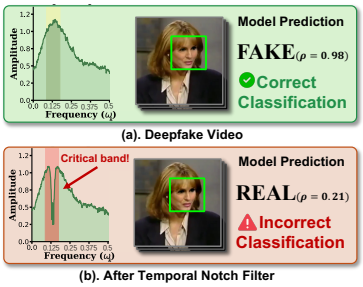


Figure 1: SLF [11] relies on temporal spectral cues for detection, and fails with misclassifications when these cues are suppressed.

To counter temporal spectral attacks and address these challenges, we propose **SpInShield** (Temporal Spectral Invariance Protection), a spectral invariance defense framework. First, to separate malicious spectrum artifacts from legitimate motion, we introduce a Learnable Spectral Adversary (LSA). This generative mechanism synthesizes severe spectral amplitude deformations while preserving the phase of nonzero Fourier coefficients before clipping, simulating worst case attacks. Second, to overcome the inductive bias toward low level statistics, we employ a weight sharing Siamese architecture. By processing the original videos and their spectrally attacked counterparts simultaneously, it forces the encoder to discard brittle spectrum shortcuts and extract robust forensic cues within a shared representation space. Finally, to achieve spectrum invariant representations without labels, we introduce a shortcut suppression optimization strategy. By enforcing adversarial spectral blindness at the representation level and symmetric prediction invariance at the prediction level, this optimization autonomously purges environment specific spectral statistics from the latent space. As a result, the model mitigates frequency shortcuts and maintains consistent classification decisions under temporal spectral attacks. In summary, contributions of this work are as follows:

- We identify that existing spatiotemporal deepfake detectors rely on amplitude-specific shortcuts in the temporal spectrum, creating an exploitable attack surface to evade detection.
- We propose SpInShield as a spectral-invariant detector by learnable adversarial amplitude spectrum synthesis and shortcut suppression optimization, reducing reliance on unstable spectral statistics.
- Experiments under simulated temporal spectral attacks demonstrate that SpInShield outperforms existing methods with a 21.30 percentage-point average AUC gain, validating its effectiveness.

2 Background

Existing deepfake detection methods operate primarily at the frame or video level. Methods at the frame level focus on spatial artifacts and high frequency anomalies [44, 42, 9] but fail to model temporal dynamics explicitly. Detectors at the video level address this by capturing spatiotemporal inconsistencies [55, 39, 11], frequently relying on temporal frequency statistics. However, temporal frequency is susceptible to corruptions and adversarial perturbations [65, 66]. Driven by shortcut learning [19], neural networks tend to exploit these spectral artifacts to minimize training loss rather than learning semantic causality [18]. Consequently, spatiotemporal encoders often regress to unstable spectral dependencies, resulting in generalization degradation under distribution shifts [20].

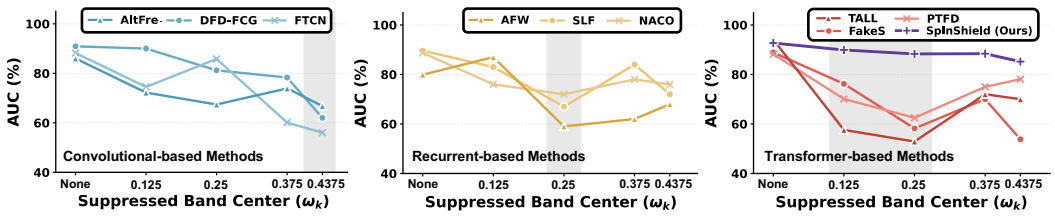


Figure 2: AUC under temporal notch suppression at representative DFT-bin frequencies. The x-axis shows the no-suppression and selected suppressed band centers $\omega_k = k/T$. Gray regions indicate the most vulnerable bands, where existing detectors degrade severely, while SpInShield remains stable.

3 Preliminary

In video analysis, temporal frequency analysis decomposes facial dynamics into two distinct components: **amplitude** and **phase** spectra. Amplitude represents the energy distribution of a temporal signal and reflects low-level statistical attributes such as global illumination shifts and compression artifacts [17, 41, 16]. In contrast, phase encodes the temporal ordering and continuous semantic dynamics in facial motion, such as lip synchronization and blinking [23].

Given a clip of T frames sampled at fps, where fps is set to 25 by default, we partition the facial ROI into M patches and compute the mean intensity of each patch to obtain temporal signals $x_m(t) \in \mathbb{R}$. The patch-wise mean intensity is not used as a detection feature; it only parameterizes a phase-preserving spectral transform that generates the environmental view. We apply DFT to obtain:

$$X_m(\omega_k) = A_m(\omega_k) e^{j P_m(\omega_k)}, \quad \omega_k = k/T,\ k = 0, \ldots, \left\lfloor \frac{T}{2} \right\rfloor, \tag{1}$$

where $A_m(\omega_k)$ and $P_m(\omega_k)$ denote amplitude and phase, respectively. Both phase and amplitude can be discriminative for deepfake detection. However, artifacts from forgery generators typically manifest in the amplitude spectrum as statistical anomalies and other low-level spectral features [45, 57, 42]. Compared with learning about the complex semantics like facial motion dynamics and micro-expression causality, models can more easily learn amplitude statistics directly [51, 40]. Moreover, for real-world attackers, manipulating phase while preserving realism is costly [54, 53]. We therefore hypothesize that current detectors rely primarily on amplitude-based shortcuts, making the amplitude spectrum the main source of overfitting and the most vulnerable attack surface.

4 Empirical Analysis

In this section, we address a central question: Do existing temporal-based detectors exhibit shortcut learning on amplitude-specific features? We use the widely adopted FaceForensics++ (FF++) dataset [44] (c23) and report the AUC. To cover diverse temporal modeling paradigms, we evaluate nine representative methods, grouped by architecture: *Convolution-based Methods*: use 3D conv./temporal shift to extract spatiotemporal features, including DFD-FCG [21], FTCN [70], AltFreezing (AltFre) [55]; *Recurrent-based Methods*: rely on Recurrent Neural Networks to model frame-to-frame continuity, including SLF [11], NACO [68], AFW [38]; *Transformer-based Methods:* use self-attention to capture global temporal dependencies, including FakeSTormer (FakeS) [39], PTFD [28], TALL [59]. To quantify the sensitivity illustrated in Figure 1, we evaluate temporal notch suppression across detector architectures, with results reported in Figure 2. Using the spectral representation in Eq. (1), we apply a notch mask $M_{\text{notch}}(\omega_k)$ to the amplitude spectrum while retaining the original phase terms:

$$\tilde{x}_m(t) = \mathcal{F}_t^{-1}\Big(\big(A_m(\omega_k) \odot M_{\text{notch}}(\omega_k)\big)\, e^{j\, P_m(\omega_k)} \Big), \tag{2}$$

where $M_{\text{notch}}(\omega_k)$ specifies stopbands on discrete DFT bins. We evaluate representative suppressed DFT-bin frequencies $\omega_k = k/T$. Experimental results reveal a shortcut learning tendency: their temporal modules rely on low-level features of amplitude spectrum. Specifically, Recurrent-based methods degrade the most, with AUC dropping by up to 41.2% relative to the no-suppression baseline. This likely occurs as their temporal modules tend to track artifacts in specific frequency bands. Similarly, Convolution-based methods and Transformer-based methods also drop by up to 36.8% and 30.6%, respectively. Thus, models may fail when the amplitude spectrum is attacked.

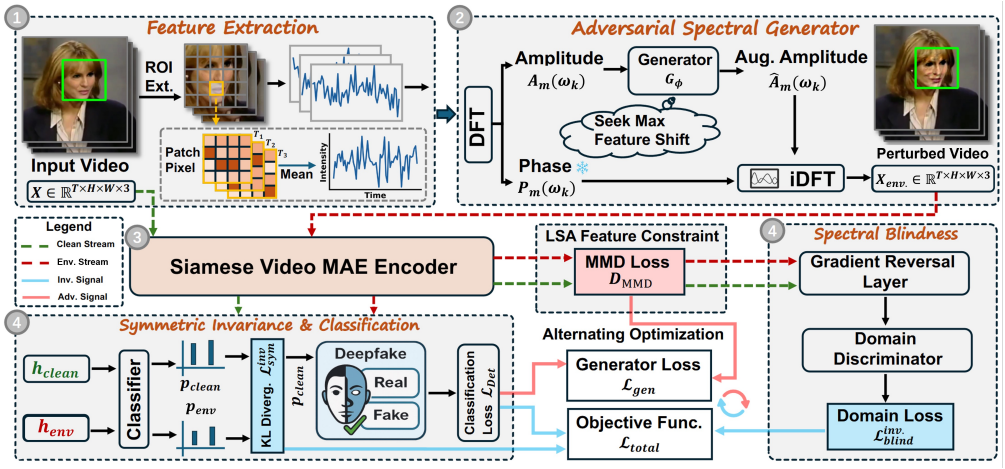


Figure 3: The framework comprising four interconnected modules: Feature Extraction, Adversarial Spectral Generator, Siamese Video MAE Encoder, and Shortcut Suppression Optimization.

5 Methodology

This section presents SpInShield shown in Figure 3, covering the Learnable Spectral Adversary, Siamese Spatiotemporal Feature Encoding, and shortcut-suppression optimization.

5.1 Learnable Spectral Adversary

Current detectors are highly sensitive to the amplitude statistics in specific temporal frequency bands, leading them to internalize spurious shortcut correlations that link spectral statistics directly to ground truth labels. To mitigate this fragility, we propose the LSA component. It uses a generative network to synthesize the adversarial spectral deformation, encouraging the detector to learn forensic signatures that are invariant to unstable spectral statistics. Following Eq. (1), we define an environment transform $\mathcal{T}_e$ that perturbs only amplitude, with $P_m(\omega_k)$ fixed:

$$\widehat{X}_m(\omega_k) = \mathcal{T}_e(A_m(\omega_k))e^{jP_m(\omega_k)}. \tag{3}$$

For window- and frame-rate-invariant parameterization, we use discrete normalized frequency bins $\Omega = \{\omega_k = k/T \mid k = 0, \ldots, \lfloor T/2 \rfloor\}$. To find the most disruptive spectral deformations beyond hand-crafted heuristics, we parameterize the environment transform with a lightweight generator G_ϕ. Given a clip, we extract patch temporal profiles $\{x_m(t)\}_{m=1}^{M}$ from the facial ROI and apply a 1D temporal DFT to obtain the one-sided spectrum $X_m(\omega_k)$ as in Eq. (1), evaluated on the discrete bins $\omega_k \in \Omega$. Under *phase consistency*, we fix $P_m(\omega_k)$ and perturb only the amplitude spectrum. For each sample i in a mini-batch of size B, the generator maps the per-clip min-max normalized amplitude $\breve{A}_{m,i}(\omega_k)$ (normalized over patches and frequencies) to a frequency-wise modulation field over DFT bins. To enforce non-negativity of the amplitude spectrum and avoid implicit phase inversions, we use an exponential parameterization:

$$\widehat{A}_{m,i}(\omega_k) = A_{m,i}(\omega_k) \odot \exp\Big(\alpha \cdot \tanh\left(G_\phi(\breve{A}_{m,i}(\omega_k))\right)\Big) \tag{4}$$

where $\odot$ denotes element-wise multiplication and α controls the perturbation strength. The DC gain is fixed to one in implementation; pixel-wise RGB reconstruction is detailed in Appendix B.3. This ensures $\widehat{A}_{m,i}(\omega_k) \geq 0$ for all $\omega_k \in \Omega$. We then construct the perturbed spectrum $\widehat{X}_{m,i}(\omega_k) = \widehat{A}_{m,i}(\omega_k)e^{jP_{m,i}(\omega_k)}$ and reconstruct the adversarial temporal signal $\widehat{x}_{m,i}(t)$ via Inverse Discrete Fourier Transform (iDFT). To ensure $\widehat{x}_{m,i}(t) \in \mathbb{R}$, we mirror the one-sided spectrum to negative frequencies to enforce Hermitian symmetry. We formulate a minimax game between the generator and the detector. The generator searches for adversarial spectral environments that maximally shift the detector's latent features, while the detector learns invariance to these shifts. Let $f_\theta(\cdot)$ be the detector's feature extractor, and let $\{(X_{\text{clean},i}, y_i)\}_{i=1}^{B}$ denote a mini-batch of labeled videos. We

measure the distributional discrepancy between original and adversarial features using Maximum Mean Discrepancy (MMD) as:

$$\mathcal{D}_{\mathrm{MMD}} = \mathrm{MMD}\Big(\{f_\theta(X_{\mathrm{clean},i})\}_{i=1}^{B}, \{f_\theta(X_{\mathrm{env},i})\}_{i=1}^{B}\Big). \tag{5}$$

We define the effective modulation mask per sample, patch, and frequency as:

$$\mathcal{M}_{m,i}(\omega_k) = \frac{\widehat{A}_{m,i}(\omega_k)}{A_{m,i}(\omega_k) + \delta}, \tag{6}$$

where δ is a small constant for numerical stability. The generator objective is:

$$\max_{\phi} \ \mathcal{L}_{\mathrm{gen}} = \frac{1}{B}\sum_{i=1}^{B} \mathcal{L}_{\mathrm{CE}}\big(p(X_{\mathrm{env},i}), y_i\big) + \gamma\, \mathcal{D}_{\mathrm{MMD}} - \frac{\lambda_{\mathrm{mask}}}{N}\big\|\mathcal{M} - \mathbf{1}\big\|_F^2, \tag{7}$$

where $\gamma \geq 0$ weights the feature-distribution shift term, $\lambda_{\mathrm{mask}} \geq 0$ regularizes the modulation mask toward identity to prevent trivial large perturbations, Ω denotes the discrete normalized DFT bins, $\mathcal{M} \in \mathbb{R}^{B\times M\times|\Omega|}$ is the modulation mask from Eq. (6), $\mathbf{1}$ is an all-ones tensor of the same shape as $\mathcal{M}$, and $N = BM|\Omega|$ is the total number of elements. The detector minimizes classification loss over original and adversarial samples:

$$\min_{\theta} \ \mathcal{L}_{\mathrm{det}} = \frac{1}{B}\sum_{i=1}^{B} \Big(\mathcal{L}_{\mathrm{CE}}\big(p(X_{\mathrm{clean},i}), y_i\big) + \mathcal{L}_{\mathrm{CE}}\big(p(X_{\mathrm{env},i}), y_i\big)\Big). \tag{8}$$

We optimize ϕ and θ alternately. This adversarial training drives G_ϕ to expose the detector's reliance on spurious amplitude spectrum statistics, urging it to learn robust forensic representations that are invariant to spectral environments.

5.2 Siamese Spatiotemporal Feature Encoding

To capture long-range temporal dependencies, we adopt a Siamese design with a ViT-based [14] VideoMAE V2 [52] backbone encoder, initialized from a checkpoint pretrained on large-scale video data. Given an input clip $X \in \mathbb{R}^{T\times H\times W\times 3}$, VideoMAE V2 tokenizes X into non-overlapping spatiotemporal tubelets. Let X_{clean} denote the original clip and X_{env} the spectrally perturbed clip generated by LSA. The tokens pass through Transformer blocks with joint space and time attention, allowing global interactions across spatiotemporal tokens and capturing temporal inconsistencies over the full clip. We obtain clip-level representations $h_{\mathrm{clean}} \in \mathbb{R}^D$ and $h_{\mathrm{env}} \in \mathbb{R}^D$ by global average pooling over output tokens from the final layer of the clean and perturbed views, respectively. During training, we use a weight-sharing Siamese strategy to map both views into a shared representation space. The shared parameter constraint encourages the encoder to suppress environment-dependent spectral artifacts and retain robust cues that remain consistent across phase-preserving environments. The dual-stream design is used only during training. At inference, we only use X_{clean}.

5.3 Optimization for Shortcut Suppression

To avoid VideoMAE's shortcut learning from spurious amplitude spectrum statistics and improve out-of-distribution (OOD) forensic generalization, we optimize the detector with a compound objective: (*i*) symmetric prediction invariance at the prediction level and (*ii*) spectral blindness at the representation level.

Spectral Blindness. Prediction consistency alone does not prevent the encoder from encoding environment-specific amplitude statistics in latent space. To suppress them, we enforce domain confusion between the clean and perturbed views. We introduce a lightweight domain discriminator $q(\cdot)$ to predict a binary domain label $d \in \{0, 1\}$, where $d = 0$ denotes the clean view and $d = 1$ denotes the perturbed view. With a Gradient Reversal Layer (GRL), we compute domain prediction:

$$\hat{d}_{\mathrm{clean}} = q(\mathrm{GRL}(h_{\mathrm{clean}})), \quad \hat{d}_{\mathrm{env}} = q(\mathrm{GRL}(h_{\mathrm{env}})). \tag{9}$$

We then minimize the domain classification loss:

$$\mathcal{L}_{\mathrm{blind}}^{\mathrm{inv}} = \mathcal{L}_{\mathrm{CE}}(\hat{d}_{\mathrm{clean}}, 0) + \mathcal{L}_{\mathrm{CE}}(\hat{d}_{\mathrm{env}}, 1). \tag{10}$$

During backpropagation, the GRL reverses the sign of gradients into the encoder, equivalent to solving $\min_q \max_\theta \mathcal{L}_{\text{blind}}^{\text{inv}}$, driving the encoder to make h_{clean} and h_{env} indistinguishable so that controllable spectral cues are discarded and only phase-consistent evidence is retained.

Symmetric Invariance. We align the predicted class distributions of X_{clean} and its spectrally perturbed counterpart X_{env} to encourage invariance to amplitude perturbations. A prediction mismatch indicates sensitivity to the applied spectral perturbation. We adopt a symmetric Kullback–Leibler (KL) divergence as a consistency regularizer:

$$\mathcal{L}_{\text{sym}}^{\text{inv}} = \frac{1}{2}\Big(\mathcal{D}_{\text{KL}}(p_{\text{clean}} \| p_{\text{env}}) + \mathcal{D}_{\text{KL}}(p_{\text{env}} \| p_{\text{clean}})\Big), \tag{11}$$

where $p_{\text{clean}} = \text{softmax}(g(h_{\text{clean}}))$ and $p_{\text{env}} = \text{softmax}(g(h_{\text{env}}))$ denote class distributions for the clean and perturbed views, and $g(\cdot)$ is the classification head. Minimizing $\mathcal{L}_{\text{sym}}^{\text{inv}}$ promotes invariance to amplitude spectrum perturbations, reduces sensitivity to environment-dependent spectral artifacts.

Crucially, this prediction-level constraint remains essential even with adversarial representation learning. While the domain classification loss reduces attack-specific spectral information by aligning marginal distributions, adversarial training with a finite-capacity discriminator cannot guarantee strict feature alignment for each sample. Without $\mathcal{L}_{\text{sym}}^{\text{inv}}$, subtle residual feature discrepancies can be amplified by the classification head and lead to unpredictable changes in predictions for individual samples. $\mathcal{L}_{\text{sym}}^{\text{inv}}$ prevents this vulnerability at the decision level by serving as an instance-level anchor.

Total Objective. Let y be the manipulation label and let $d \in \{0, 1\}$ denote the domain label, where $d = 0$ corresponds to the clean view X_{clean} and $d = 1$ corresponds to the spectrally perturbed view X_{env} generated by LSA. The final training objective is defined as follows:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{det}} + \lambda_{\text{sym}}\mathcal{L}_{\text{sym}}^{\text{inv}} + \lambda_{\text{blind}}\mathcal{L}_{\text{blind}}^{\text{inv}}, \tag{12}$$

where λ_{sym} and λ_{blind} are non-negative hyperparameters that balance symmetric prediction invariance, spectral blindness, and the primary detection objective.

6 Experiment

6.1 Experiment Setting

Dataset. We evaluate SpInShield on the following datasets: Celeb-DF-v2 (CDF-v2) [31], DeepFakeDetection (DFD) dataset [15], Deeperforensics (DFo) dataset [26] and WildDeepFake (WDF) dataset [72] using a model trained on the FF++ [44]. To further assess the effects of GAN and Diffusion based generators, we add two evaluation setups using DiffSwap [69] and DaGAN [24] on the FFHQ dataset [27], each comprising 1,000 real samples and 1,000 fake samples.

Data Preprocessing. Following recent practices [11, 21], we utilize Dlib [29] to extract facial landmarks for alignment, and Retinaface [13] for face cropping. To mitigate spurious temporal artifacts caused by inter-frame misalignment, we enforce strict spatial consistency by fixing the bounding box coordinates to the initial frame. For temporal sampling, we randomly extract non-overlapping clips of 2 to 4 seconds from each video. From these clips, we extract a sequence of $T = 16$ consecutive frames to maintain the continuity of the frequency spectrum. Finally, these frames undergo a comprehensive video-level augmentation pipeline to simulate real-world variations. For more implementation details, please refer to Appendix B.

Baselines. For a comprehensive evaluation, we compare SpInShield with the following advanced and representative baselines, which are categorized into: *Frame-level methods*: SLADD [8], SBI [47], UCF [60], IID [25], LSDA [61], ProDet [9], and CDFA [32]; *Video-level methods*: TALL [59], SLF [11], NACO [68], AFW [38], AltFreezing [55], FTCN [70], PTFD [28], FakeSTormer [39], and DFD-FCG [21]. All baseline numbers reported in this paper are obtained from our own evaluations. We use the corresponding implementations and model-specific preprocessing, and evaluate each method under the protocol specified for the experiment.

6.2 Non-Adversarial Evaluation

This section validates SpInShield's robustness against standard deepfake generators as a baseline. In Table 1, SpInShield obtains state-of-the-art (SOTA) results in cross-domain evaluation, obtaining an

Table 1: Cross-dataset evaluation on FF++ (c23). **Avg.** excludes in-domain results. DiF. (DiffSwap) and DaG. (DaGAN) denote evaluation sets constructed using FFHQ and the respective generators, each containing 1,000 real and 1,000 fake samples. * indicates statistically significant improvements over the strongest baseline ($p < 0.05$, two-tailed t-test) on the average metric.

	SLADD	SBI	UCF	IID	LSDA	ProDet	CDFA	TALL	SLF	NACO	AFW	AltFre	FTCN	PTFD	FakeST	DFD-FCG	**Ours**
FF++ *(In-D.)*	.984	.997	.995	.993	.990	.992	.993	**.998**	.984	.997	.980	.995	.997	.984	.995	.993	.996
CDFv2	.837	.886	.837	.838	.875	.926	.938	.871	.890	.895	.824	.890	.869	.914	.924	**.949**	.948
DFD	.904	.827	.867	.939	.881	.901	.954	.892	.961	.970	.860	.902	.879	.920	**.985**	.928	.981
DFo	.800	.899	.808	.808	.768	.879	.973	**.996**	.990	.995	.819	.990	.988	.954	.960	.994	.987
WDF	.690	.703	.774	.666	.797	.781	.796	.680	.741	.686	.693	.773	.791	.741	.766	.875	**.889**
DiffSwap	.691	.704	.784	.729	.703	.701	.750	.689	.748	.722	.692	.745	.752	.820	.845	.872	**.894**
DaGAN	.730	.743	.722	.683	.793	.693	.728	.700	.831	.741	.753	.802	.783	.874	.820	.884	**.897**
Avg.	.775	.794	.799	.777	.803	.814	.857	.805	.860	.835	.774	.850	.844	.871	.883	.917	**.933***

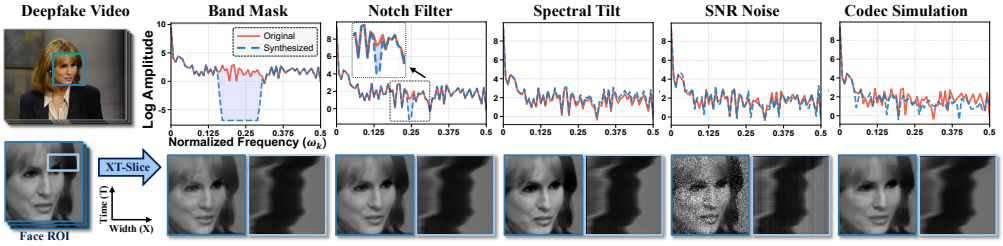


Figure 4: Visualization of amplitude spectral attack synthesis. Facial XT-slices are extracted from the input video to generate diverse attacks. The top row compares the log-amplitude spectra of the original (red solid) and synthesized (blue dashed) signals over normalized DFT-bin frequencies ω_k. The bottom row shows the corresponding visual effects. Perturbations are amplified for visualization only.

average AUC of 93.3%. This surpasses the second-best competitor (DFD-FCG) by a margin of 1.6%. This generalization stems from our temporal spectral synthesis design. By explicitly exposing the model to diverse simulated spectral environments in training, we encourage the encoder to discount spurious amplitude statistics that vary across datasets and recording pipelines. Furthermore, the shortcut suppression optimization urges the model to effectively learn phase spectrum information. A matched VideoMAE V2 baseline further indicates that the clean generalization gains cannot be attributed to backbone pretraining alone (Appendix C.3). Additional evaluations on FFIW and a DF40 talking-face subset show comparable clean performance to DFD-FCG and higher robustness under spectral perturbations (Appendix C.2).

6.3 Resilience to Temporal Spectral Attack

Robustness Against Synthetic Spectral Attacks. To evaluate the generalizability against unseen amplitude spectral conditions, we define a set of spectral attacks $\mathcal{E}$ as benchmarks. We construct these attacks following Section 5.1 with specific, fixed transformations $\mathcal{T}_e$ to the amplitude spectrum $A(\omega_k)$, and reconstruct the evaluation samples via iDFT. Each attack $e \in \mathcal{E}$ is categorized into Type-I: de-correlation via masking/reshaping, and Type-II: robustness via channel/platform distortions: *Random Band Masking:* We suppress $B \sim \text{Unif}\{1, 2, 3\}$ randomly placed contiguous DFT-bin bands with Tukey-tapered masks, where each band covers $K_{\text{band}} \sim \text{Unif}\{1, 2, 3, 4\}$ bins; *Narrow-band Notch:* We attenuate localized DFT-bin regions using a smooth cosine-tapered notch mask. The center bin is sampled uniformly from non-DC and non-Nyquist bins, with a total width of $K_{\text{notch}} \sim \text{Unif}\{1, 2\}$ bins; *Spectral Tilt:* We re-weight each DFT bin in the log-amplitude domain as $\log \widehat{A}(\omega_k) = \log(A(\omega_k) + \epsilon_0) + \beta_1\omega_k + \beta_2\omega_k^2$, with $\beta_1, \beta_2 \sim U(-1.5, 1.5)$; *SNR-based Noise:* We inject per-bin Gaussian noise in the log-amplitude domain: $\log \widehat{A}(\omega_k) = \log(A(\omega_k) + \epsilon_0) + \eta(\omega_k)$, where $\eta(\omega_k) \sim \mathcal{N}(0, \sigma^2)$. Figure 4 presents the amplitude spectral deviations of each attack.

Table 2 reports AUC under four amplitude spectral attacks across six datasets. SpInShield consistently outperforms across all attacks and datasets, with average AUC scores of 89.35, 90.47, 88.47, and 87.43, respectively. In contrast, baselines suffer substantial degradation under these attacks, with

Table 2: Robustness evaluation under amplitude spectrum attacks. * indicates statistically significant improvements over the strongest baseline via a two-tailed t-test ($p < 0.05$) on the average metric.

Method	CDFv2	DFD	DFo	WDF	DiF.	DaG.	Avg.	CDFv2	DFD	DFo	WDF	DiF.	DaG.	Avg.
	(a) Random Band Mask							(b) Narrow-band Notch						
FakeST	70.21	73.35	75.46	58.52	58.39	56.65	65.43	72.55	76.68	78.53	60.71	60.45	59.80	68.12
AltFre	65.23	73.55	76.22	60.19	62.50	68.59	67.71	68.48	76.55	79.23	62.82	64.80	71.23	70.52
PTFD	66.18	71.21	72.85	59.52	68.54	74.50	68.80	69.24	74.53	75.63	61.54	70.24	76.80	71.33
TALL	63.58	70.85	77.13	58.44	58.24	59.53	64.63	65.85	73.45	79.59	59.65	60.49	61.86	66.81
SLF	67.21	75.42	76.81	61.22	63.54	70.20	69.07	69.52	77.81	79.16	63.42	65.80	72.50	71.37
NACO	66.85	76.16	78.44	59.80	61.50	62.89	67.61	68.85	78.57	80.25	61.24	63.82	65.20	69.66
AFW	60.51	66.83	64.23	55.42	57.80	62.56	61.23	62.46	68.55	66.73	57.25	59.54	64.89	63.24
DFD-FCG	72.28	75.56	79.41	59.25	59.48	58.38	67.39	74.35	78.42	82.34	62.24	61.28	60.46	69.85
FTCN	64.52	68.25	75.53	58.89	64.20	66.58	66.33	66.55	71.41	78.20	60.52	66.54	68.87	68.68
VideoMAE(B)	60.15	74.25	55.42	54.80	54.08	56.51	59.20	61.83	76.51	56.16	56.06	55.81	58.23	60.77
SpInShield	**91.82**	**90.88**	**94.79**	**86.91**	**86.47**	**85.23**	**89.35** *	**92.61**	**91.98**	**96.12**	**87.91**	**87.88**	**86.32**	**90.47** *
	(c) Spectral Tilt							(d) SNR-based Noise						
FakeST	65.24	70.41	72.48	55.26	54.29	53.53	61.87	62.38	66.45	69.36	51.48	50.39	50.50	58.43
AltFre	62.44	70.58	75.65	58.24	59.61	64.50	65.17	60.15	68.27	72.56	55.43	56.40	61.24	62.34
PTFD	63.85	69.25	71.52	57.49	66.44	71.24	66.63	61.22	66.51	68.40	54.52	61.53	**66.40**	63.10
TALL	61.55	68.48	74.23	56.52	55.62	56.48	62.15	58.55	65.42	71.22	53.25	52.11	53.80	59.06
SLF	64.55	72.59	75.81	59.25	60.23	67.40	66.64	61.88	69.55	72.44	56.56	57.50	62.81	63.46
NACO	64.20	74.11	76.58	58.83	58.43	59.52	65.28	62.15	71.21	74.52	55.85	55.29	56.40	62.57
AFW	58.52	64.21	62.55	53.42	54.26	58.68	58.61	56.45	61.53	60.23	51.53	51.59	55.20	56.09
DFD-FCG	68.52	72.63	75.71	56.78	56.64	54.62	64.15	64.75	68.58	72.69	53.57	52.48	51.65	60.62
FTCN	62.52	67.51	73.45	57.82	61.53	63.40	64.37	60.40	65.25	70.57	55.24	58.20	60.59	61.71
VideoMAE(B)	57.43	70.12	53.81	53.21	52.37	53.51	56.74	55.20	66.80	52.13	51.46	50.81	51.20	54.60
SpInShield	**90.52**	**90.18**	**94.81**	**86.11**	**85.28**	**83.92**	**88.47** *	**89.82**	**89.48**	**93.11**	**85.41**	**84.18**	**82.58**	**87.43** *

Table 3: Robustness under real-world video processing pipelines (AUC %). * indicates statistically significant improvements over the strongest baseline via a two-tailed t-test ($p < 0.05$).

Pipeline	FakeST	AltFre	PTFD	TALL	SLF	NACO	AFW	DFD-FCG	FTCN	VideoMAE(B)	SpInShield
Black-Box Platform	82.38	80.63	82.97	78.32	83.11	81.23	73.73	76.08	81.02	70.94	**89.92***
Multi-stage Codec	77.53	76.37	78.72	73.99	78.58	77.02	69.16	75.27	76.54	65.64	**87.53***
Heavy Codec	71.56	70.66	72.81	67.52	72.95	70.89	63.71	75.38	71.17	58.80	**83.28***
Cross-Platform	75.53	73.71	76.27	71.47	76.31	74.43	66.84	75.61	74.34	64.24	**84.44***

most drop below AUC of 70%, while the VideoMAE baselines approach random-level performance under SNR-based Noise (54.60%). Notably, SpInShield maintains an average absolute improvement of over 20 percentage points in AUC compared with the strongest baselines across all attack types, showing that the invariance learned through LSA and Shortcut Suppression Optimization generalizes effectively to unseen, hand-crafted spectral attacks beyond those encountered during training. We assess the perceptual severity of the four spectral attacks using frame-level quality and optical-flow consistency metrics. The measurements suggest limited appearance and motion distortion under the evaluated budgets (Appendix C.1). Retraining on temporally smoothed FF++ improves baseline robustness but leaves a substantial gap to SpInShield (Appendix C.5).

Robustness in Real World Deployment Scenarios. We further evaluate SpInShield under real world video processing pipelines to assess its practical robustness under non differentiable post processing operations. To evaluate generalization in practical deployment, we assess our method under four real world video post processing pipelines: platform round trips, multi stage codec chains, heavy compression, and cross platform cascades. As Table 3 shows, SpInShield consistently achieves the highest average AUC across all settings, maintaining performance between 83.28% and 89.92%. While baseline methods suffer substantial performance drops under severe compression, the phase consistent forensic signals learned by SpInShield remain stable and outperform all competitors.

6.4 Adaptive Evaluation against SpInShield

We conduct an adaptive white box evaluation where attackers optimize video specific amplitude modulation masks. As Table 4 shows, Baseline models degrade significantly, but SpInShield retains the highest average AUC of 66.01%. We additionally evaluate pixel-space PGD attacks not used during training and observe higher AUC than DFD-FCG at both tested perturbation budgets (Appendix C.4).

Table 4: Adaptive white-box evaluation. VMAE denotes the plain VideoMAE.

	FakeST	SLF	DFD-FCG	VMAE	Ours
CDFv2	54.27	55.74	60.43	47.83	**67.74**
DFD	58.83	60.18	65.78	53.62	**66.42**
DFo	60.14	62.45	64.92	45.91	**72.86**
WDF	47.62	49.61	53.47	43.27	**61.83**
DiffSwap	48.91	51.36	55.86	44.58	**63.27**
DaGAN	47.36	56.27	56.21	45.18	**63.91**
Avg.	52.86	55.94	59.45	46.73	**66.01**

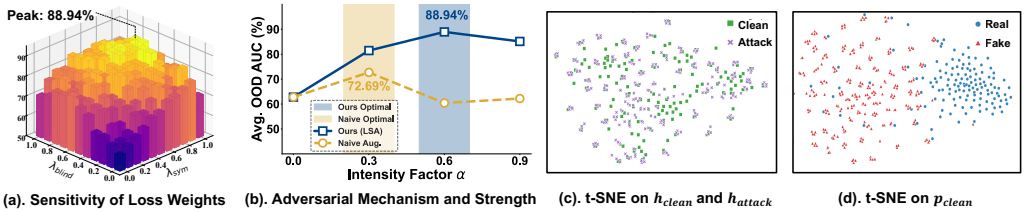
Table 5: (a) Robustness under full information and phase-only input. (b) Component contributions on in-domain performance and OOD robustness.

(a) Full info. vs. phase-only.

	Full	Phase	Δ
FakeST.	88.3	60.2	-28.1
AltFre.	85.0	58.7	-26.3
PTFD	87.1	57.5	-29.6
TALL	80.5	55.6	-24.9
SLF	86.0	59.4	-26.6
NACO	83.5	58.2	-25.3
AFW	77.4	52.1	-25.3
DFD-FCG	91.7	64.5	-27.2
FTCN	84.4	57.8	-26.6
Ours	**93.3**	**79.6**	**-13.7**

(b) Ablation study.

Method	In-Dom.	OOD						
	FF++	CDF-v2	DFD	DFo	WDF	DS	DaGAN	Avg.
VideoMAE(B)	86.89	58.66	71.93	54.38	53.88	53.27	54.86	57.83
w/ Naive	94.81	62.19	65.98	64.32	62.68	59.29	64.87	63.22
w/o $\mathcal{L}_{\text{sym}}^{\text{inv}}$	96.10	80.35	88.65	81.45	74.60	73.80	80.51	79.89
w/o $\mathcal{L}_{\text{blind}}^{\text{inv}}$	97.23	85.65	90.20	86.80	80.50	71.54	74.30	81.50
Complete	**99.60**	**91.26**	**90.60**	**94.75**	**86.63**	**85.93**	**84.45**	**88.94**



(a). Sensitivity of Loss Weights (b). Adversarial Mechanism and Strength (c). t-SNE on h_{clean} and h_{attack} (d). t-SNE on p_{clean}

Figure 5: Quantitative and qualitative evaluation: (a) Joint impact of hyperparameters λ_{sym} and λ_{blind} on OOD generalization performance. (b) Impact of the amplitude spectral attack intensity α on the OOD generalization capability. (c) Feature space distribution visualization between clean and perturbed samples. (d) Decision space clustering visualization of real and fake video samples.

6.5 Phase-Only Evaluation

To validate whether SpInShield uses phase information more effectively, we evaluate the performance in a setting where all amplitude spectrum information is removed. Table 5 (a) reports the performance. Existing baselines suffer significant performance degradation with phase spectrum information only, exhibiting AUC drops from 24.9% to 29.6%. This performance gap demonstrates their heavy reliance on amplitude spectrum information. Conversely, SpInShield maintains an AUC of 79.6% with the minimum performance gap of 13.7% among all evaluated methods. These results suggest that SpInShield effectively uses phase information for reliable deepfake detection.

6.6 Ablation Study

To isolate each component's contribution within SpInShield, we conduct an ablation study on FF++ for in-domain evaluation, and report OOD performance as average AUC under four amplitude spectral attacks. In Table 5 (b), we add components on top of the VideoMAE backbone: Naive Spectral Augmentation with random Gaussian noise on amplitude spectrum (w/o LSA), $\mathcal{L}_{\text{sym}}^{\text{inv}}$, and $\mathcal{L}_{\text{blind}}^{\text{inv}}$. The baseline shows in-domain performance with an AUC of 86.89% but degrades severely in OOD settings with 57.83% on average, confirming the amplitude shortcut problem. Naive spectral augmentation improves marginally with +5.39%, as random noise fails to synthesize targeted adversarial deformations that expose amplitude shortcuts effectively. Removing $\mathcal{L}_{\text{sym}}^{\text{inv}}$ or $\mathcal{L}_{\text{blind}}^{\text{inv}}$ individually yields further gains but leaves a substantial gap, showing that prediction-level consistency and feature-level domain confusion are complementary.

A matched-objective comparison with fixed spectral augmentation further supports the benefit of the learnable adversary under comparable perturbation strength.

Matched-objective comparison. To isolate the contribution of the Learnable Spectral Adversary (LSA), we compare SpInShield with a Fixed-SA variant. Fixed-SA retains the same Siamese training pipeline, classification objective, prediction-consistency loss $\mathcal{L}_{\text{sym}}^{\text{inv}}$, and domain-confusion loss $\mathcal{L}_{\text{blind}}^{\text{inv}}$. The backbone, optimizer, learning rate, training schedule, data split, and evaluation protocol are also unchanged. The only difference is that LSA is replaced by fixed spectral augmentation.

To construct Fixed-SA, we apply the converged LSA from the full SpInShield model to the validation split and collect the mean and standard deviation of its log-amplitude masks for each temporal-frequency bin. These statistics parameterize a fixed mask-sampling distribution during Fixed-SA

training. No test data are used to calibrate this distribution. Both variants use the same amplitude-only modulation, spectral reconstruction, and clipping procedure. The measured mean perturbation-strength ratio of Fixed-SA to LSA is 0.97, indicating comparable perturbation strength.

Table 6: Matched-objective ablation of LSA. Both variants use the same losses and training settings with comparable perturbation strength. Results are average OOD AUC under the four spectral attacks.

Variant	OOD Avg. AUC (%)
Fixed-SA	78.37
SpInShield	**88.94**

As shown in Table 6, SpInShield outperforms Fixed-SA by 10.57 AUC percentage points. Unlike Fixed-SA, whose sampling distribution remains unchanged throughout training, LSA is jointly optimized against the evolving detector. This matched-objective comparison supports the benefit of adaptive spectral environment generation beyond fixed augmentation with comparable perturbation strength.

6.7 Sensitivity Analysis

Figure 5 (a) demonstrates that SpInShield performs best when $\lambda_{\text{blind}} = 0.7$ and $\lambda_{\text{sym}} = 0.9$. It illustrates that optimizing both attack invariance and spectral blindness can suppress amplitude spectrum shortcuts under a proper loss weight setup as insufficient adversarial weight setup fails to eliminate amplitude spectrum shortcuts. Figure 5 (b) shows that SpInShield obtains a peak AUC of 88.94% when $\alpha = 0.6$, illustrating the impact of the perturbation intensity. Insufficient perturbation fails to suppress amplitude shortcuts, whereas excessive intensity destroys forensic cues and degrades performance. Moreover, LSA shows stronger performance compared to naive augmentation.

6.8 Latent Space Visualization

We visualize feature distributions and validate SpInShield's spectrum shortcut suppression via t-SNE. Figure 5 (c) plots the encoder representations h_{clean} and h_{env}. The overlap between these two distributions shows the encoder is amplitude-blind, mapping both clean and perturbed inputs into a shared invariant latent space. Figure 5 (d) illustrates that h_{clean} forms distinct clusters for real and fake samples in the decision space. This clear separation demonstrates that the model retains strong discriminative capability for forgery classification after suppressing amplitude spectrum shortcuts.

7 Conclusion

This work addresses the limitations of existing spatiotemporal deepfake detectors in relying on shortcut learning with fragile temporal spectrum cues. We propose SpInShield framework, building a learnable spectral adversary component to synthesize severe amplitude spectrum deformations that expose latent vulnerabilities. Building upon LSA, we introduce a dual-objective shortcut suppression strategy: spectral blindness removes amplitude-specific information from the latent space, while symmetric invariance anchors prediction consistency at the decision level, collectively compelling the encoder to prioritize phase-consistent forensic cues. SpInShield achieves advanced performance under simulated temporal spectral attacks, demonstrating its effectiveness.

Acknowledgments and Disclosure of Funding

The authors received no third-party funding or support for the work reported in this paper. The authors further declare that they have no competing interests, financial or otherwise, that could reasonably be perceived to have influenced the conduct of the research, the interpretation of its findings, or the presentation of its conclusions.

References

[1] N. Ahmed, T. Natarajan, and K.R. Rao. Discrete cosine transform. *IEEE Transactions on Computers*, C-23(1):90–93, 1974. doi: 10.1109/T-C.1974.223784.

[2] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6816–6826. IEEE, 2021.

[3] Zhongjie Ba, Qingyu Liu, Zhenguang Liu, Shuang Wu, Feng Lin, Li Lu, and Kui Ren. Exposing the deception: Uncovering more forgery clues for deepfake detection. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pages 719–728, 2024.

[4] Sara Beery, Grant van Horn, and Pietro Perona. Recognition in terra incognita, 2018. URL `https://arxiv.org/abs/1807.04975`.

[5] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *Proceedings of the International Conference on Machine Learning (ICML)*, July 2021.

[6] Junyi Cao, Chao Ma, Taiping Yao, Shen Chen, Shouhong Ding, and Xiaokang Yang. End-to-end reconstruction-classification learning for face forgery detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4113–4122, June 2022.

[7] Nicholas Carlini and David Wagner. Audio adversarial examples: Targeted attacks on speech-to-text. In *2018 IEEE Security and Privacy Workshops (SPW)*, pages 1–7, 2018. doi: 10.1109/SPW.2018.00009.

[8] Liang Chen, Yong Zhang, Yibing Song, Lingqiao Liu, and Jue Wang. Self-supervised learning of adversarial example: Towards good generalizations for deepfake detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18689–18698. IEEE, 2022.

[9] Jikang Cheng, Zhiyuan Yan, Ying Zhang, Yuhao Luo, Zhongyuan Wang, and Chen Li. Can we leave deepfake data behind in training deepfake detector? In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS '24, Red Hook, NY, USA, 2024. Curran Associates Inc. ISBN 9798331314385.

[10] Bobby Chesney and Danielle Citron. Deep fakes: A looming challenge for privacy, democracy, and national security. *Calif. L. Rev.*, 107:1753, 2019.

[11] Jongwook Choi, Taehoon Kim, Yonghyun Jeong, Seungryul Baek, and Jongwon Choi. Exploiting style latent flows for generalizing deepfake video detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1133–1143. IEEE, 2024.

[12] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1800–1807. Ieee, 2017.

[13] Jiankang Deng, Jia Guo, Evangelos Ververas, Irene Kotsia, and Stefanos Zafeiriou. Retinaface: Single-shot multi-level face localisation in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5202–5211. IEEE, 2020.

[14] Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[15] Nick Dufour and Andrew Gully. Contributing data to deepfake detection research. `https://ai.googleblog.com/2019/09/contributing-data-to-deepfake-detection.html`, 9 2019. Google AI Blog. Accessed: 2023-07-30.

[16] Tarik Dzanic, Karan Shah, and Freddie Witherden. Fourier spectrum discrepancies in deep network generated images. *Advances in neural information processing systems*, 33:3022–3032, 2020.

[17] David J Field and Damon M Chandler. Method for estimating the relative contribution of phase and power spectra to the total information in natural-scene patches. *Journal of the Optical Society of America A*, 29(1):55–67, 2011.

[18] Joel Frank, Thorsten Eisenhofer, Lea Schönherr, Asja Fischer, Dorothea Kolossa, and Thorsten Holz. Leveraging frequency analysis for deep fake image recognition. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org, 2020.

[19] Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020.

[20] David Güera and Edward J. Delp. Deepfake video detection using recurrent neural networks. In *2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–6, 2018. doi: 10.1109/AVSS.2018.8639163.

[21] Yue-Hua Han, Tai-Ming Huang, Kai-Lung Hua, and Jun-Cheng Chen. Towards more general video-based deepfake detection through facial component guided adaptation for foundation model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 22995–23005. IEEE, 2025.

[22] Katherine L. Hermann, Hossein Mobahi, Thomas Fel, and Michael C. Mozer. On the foundations of shortcut learning, 2024. URL `https://arxiv.org/abs/2310.16228`.

[23] Omar Hommos, Silvia L Pintea, Pascal SM Mettes, and Jan C van Gemert. Using phase instead of optical flow for action recognition. In *European Conference on Computer Vision*, pages 678–691. Springer, 2018.

[24] Fa-Ting Hong, Longhao Zhang, Li Shen, and Dan Xu. Depth-aware generative adversarial network for talking head video generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3397–3406, 2022.

[25] Baojin Huang, Zhongyuan Wang, Jifan Yang, Jiaxin Ai, Qin Zou, Qian Wang, and Dengpan Ye. Implicit identity driven deepfake face swapping detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4490–4499, 2023. doi: 10.1109/CVPR52729.2023.00436.

[26] Liming Jiang, Ren Li, Wayne Wu, Chen Qian, and Chen Change Loy. Deeperforensics-1.0: A large-scale dataset for real-world face forgery detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2886–2895. IEEE, 2020.

[27] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4396–4405. IEEE, 2019.

[28] Taehoon Kim, Jongwook Choi, Yonghyun Jeong, Haeun Noh, Jaejun Yoo, Seungryul Baek, and Jongwon Choi. Beyond spatial frequency: Pixel-wise temporal frequency-based deepfake video detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 11198–11207, October 2025.

[29] Davis E. King. Dlib-ml: A machine learning toolkit. *J. Mach. Learn. Res.*, 10:1755–1758, December 2009. ISSN 1532-4435.

[30] Hanzhe Li, Jiaran Zhou, Yuezun Li, Baoyuan Wu, Bin Li, and Junyu Dong. Freqblender: enhancing deepfake detection by blending frequency knowledge. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS '24, Red Hook, NY, USA, 2024. Curran Associates Inc. ISBN 9798331314385.

[31] Yuezun Li, Xin Yang, Pu Sun, Honggang Qi, and Siwei Lyu. Celeb-df: A large-scale challenging dataset for deepfake forensics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3204–3213. IEEE, 2020.

[32] Yuzhen Lin, Wentang Song, Bin Li, Yuezun Li, Jiangqun Ni, Han Chen, and Qiushi Li. Fake it till you make it: Curricular dynamic forgery augmentations towards general deepfake detection, 2024. URL `https://arxiv.org/abs/2409.14444`.

[33] Honggu Liu, Xiaodan Li, Wenbo Zhou, Yuefeng Chen, Yuan He, Hui Xue, Weiming Zhang, and Nenghai Yu. Spatial-phase shallow learning: rethinking face forgery detection in frequency domain. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 772–781. IEEE, 2021.

[34] Yuyang Long, Qilong Zhang, Boheng Zeng, Lianli Gao, Xianglong Liu, Jian Zhang, and Jingkuan Song. Frequency domain model augmentation for adversarial attack. In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, *Computer Vision – ECCV 2022*, pages 549–566, Cham, 2022. Springer Nature Switzerland. ISBN 978-3-031-19772-7.

[35] Wei Mao, Miaomiao Liu, Mathieu Salzmann, and Hongdong Li. Learning trajectory dependencies for human motion prediction, 2020. URL `https://arxiv.org/abs/1908.05436`.

[36] Momina Masood, Mariam Nawaz, Khalid Mahmood Malik, Ali Javed, Aun Irtaza, and Hafiz Malik. Deepfakes generation and detection: state-of-the-art, open challenges, countermeasures, and way forward. *Applied intelligence*, 53(4):3974–4026, 2023.

[37] Yisroel Mirsky and Wenke Lee. The creation and detection of deepfakes. *ACM Computing Surveys (CSUR)*, 54:1 – 41, 2020.

[38] Daniel Mas Montserrat, Hanxiang Hao, Sri Kalyan Yarlagadda, Sriram Baireddy, Ruiting Shao, János Horváth, Emily Bartusiak, Justin Yang, David Güera, Fengqing Zhu, et al. Deepfakes detection with automatic face weighting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 2851–2859. IEEE, 2020.

[39] Dat Nguyen, Marcella Astrid, Anis Kacem, Enjie Ghorbel, and Djamila Aouada. Vulnerability-aware spatio-temporal learning for generalizable deepfake video detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10786–10796, 2025.

[40] Thanh Thi Nguyen, Quoc Viet Hung Nguyen, Dung Tien Nguyen, Duc Thanh Nguyen, Thien Huynh-The, Saeid Nahavandi, Thanh Tam Nguyen, Quoc-Viet Pham, and Cuong M. Nguyen. Deep learning for deepfakes creation and detection: A survey. *Computer Vision and Image Understanding*, 223:103525, 2022. ISSN 1077-3142. doi: https://doi.org/10.1016/j.cviu.2022.103525. URL `https://www.sciencedirect.com/science/article/pii/S1077314222001114`.

[41] A.V. Oppenheim and J.S. Lim. The importance of phase in signals. *Proceedings of the IEEE*, 69(5):529–541, 1981. doi: 10.1109/PROC.1981.12022.

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

[43] Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred A. Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks, 2019. URL `https://arxiv.org/abs/1806.08734`.

[44] Andreas Rössler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Nießner. Faceforensics++: Learning to detect manipulated facial images. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1–11, 2019.

[45] Michael Rubinstein. *Analysis and visualization of temporal variations in video*. PhD thesis, Massachusetts Institute of Technology, 2014.

[46] Harshay Shah, Kaustav Tamuly, Aditi Raghunathan, Prateek Jain, and Praneeth Netrapalli. The pitfalls of simplicity bias in neural networks, 2020. URL `https://arxiv.org/abs/2006.07710`.

[47] Kaede Shiohara and Toshihiko Yamasaki. Detecting deepfakes with self-blended images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18699–18708, 2022. doi: 10.1109/CVPR52688.2022.01816.

[48] Ruben Tolosana, Ruben Vera-Rodriguez, Julian Fierrez, Aythami Morales, and Javier Ortega-Garcia. Deepfakes and beyond: A survey of face manipulation and fake detection. *Information fusion*, 64:131–148, 2020.

[49] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4489–4497, 2015.

[50] Luisa Verdoliva. Media forensics and deepfakes: An overview. *IEEE Journal of Selected Topics in Signal Processing*, 14:910–932, 2020.

[51] Neal Wadhwa, Michael Rubinstein, Frédo Durand, and William T. Freeman. Phase-based video motion processing. *ACM Trans. Graph.*, 32(4), July 2013. ISSN 0730-0301. doi: 10.1145/2461912.2461966. URL https://doi.org/10.1145/2461912.2461966.

[52] Limin Wang, Bingkun Huang, Zhiyu Zhao, Zhan Tong, Yinan He, Yi Wang, Yali Wang, and Yu Qiao. Videomae v2: Scaling video masked autoencoders with dual masking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14549–14560. IEEE, 2023.

[53] Weihong Wang and Hany Farid. Exposing digital forgeries in video by detecting double mpeg compression. In *Proceedings of the 8th Workshop on Multimedia and Security*, MM&Sec '06, page 37–47, New York, NY, USA, 2006. Association for Computing Machinery. ISBN 1595934936. doi: 10.1145/1161366.1161375. URL https://doi.org/10.1145/1161366.1161375.

[54] Yan Wang, Qindong Sun, Dongzhu Rong, and Rong Geng. Multi-domain awareness for compressed deepfake videos detection over social networks guided by common mechanisms between artifacts. *Computer Vision and Image Understanding*, 247:104072, 2024. ISSN 1077-3142. doi: https://doi.org/10.1016/j.cviu.2024.104072. URL https://www.sciencedirect.com/science/article/pii/S107731422400153X.

[55] Zhendong Wang, Jianmin Bao, Wengang Zhou, Weilun Wang, and Houqiang Li. Altfreezing for more general video face forgery detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4129–4138. IEEE, 2023.

[56] Chao-Yuan Wu, Manzil Zaheer, Hexiang Hu, R. Manmatha, Alexander J. Smola, and Philipp Krähenbühl. Compressed video action recognition, 2018. URL https://arxiv.org/abs/1712.00636.

[57] Hao-Yu Wu, Michael Rubinstein, Eugene Shih, John Guttag, Frédo Durand, and William Freeman. Eulerian video magnification for revealing subtle changes in the world. *ACM Trans. Graph.*, 31(4), July 2012. ISSN 0730-0301. doi: 10.1145/2185520.2185561. URL https://doi.org/10.1145/2185520.2185561.

[58] Xingjian Wu, Xiangfei Qiu, Zhengyu Li, Yihang Wang, Jilin Hu, Chenjuan Guo, Hui Xiong, and Bin Yang. CATCH: Channel-aware multivariate time series anomaly detection via frequency patching. In *ICLR*, 2025.

[59] Yuting Xu, Jian Liang, Gengyun Jia, Ziming Yang, Yanhao Zhang, and Ran He. Tall: Thumbnail layout for deepfake video detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 22658–22668, 2023.

[60] Zhiyuan Yan, Yong Zhang, Yanbo Fan, and Baoyuan Wu. Ucf: Uncovering common features for generalizable deepfake detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 22355–22366. IEEE, 2023.

[61] Zhiyuan Yan, Yuhao Luo, Siwei Lyu, Qingshan Liu, and Baoyuan Wu. Transcending forgery specificity with latent space augmentation for generalizable deepfake detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8984–8994. IEEE, 2024.

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

[63] Zhiyuan Yan, Jiangming Wang, Peng Jin, Ke-Yue Zhang, Chengchun Liu, Shen Chen, Taiping Yao, Shouhong Ding, Baoyuan Wu, and Li Yuan. Orthogonal subspace decomposition for generalizable AI-generated image detection. In *Forty-second International Conference on Machine Learning*, 2025.

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

[65] Chenyu Yi, Siyuan Yang, Haoliang Li, Yap peng Tan, and Alex Kot. Benchmarking the robustness of spatial-temporal models against corruptions, 2022. URL `https://arxiv.org/abs/2110.06513`.

[66] Dong Yin, Raphael Gontijo Lopes, Jonathon Shlens, Ekin D. Cubuk, and Justin Gilmer. A fourier perspective on model robustness in computer vision, 2020. URL `https://arxiv.org/abs/1906.08988`.

[67] John R. Zech, Marcus A. Badgeley, Manway Liu, Anthony B. Costa, Joseph J. Titano, and Eric Karl Oermann. Variable generalization performance of a deep learning model to detect pneumonia in chest radiographs: A cross-sectional study. *PLOS Medicine*, 15(11):1–17, 11 2018. doi: 10.1371/journal.pmed.1002683. URL `https://doi.org/10.1371/journal.pmed.1002683`.

[68] Daichi Zhang, Zihao Xiao, Shikun Li, Fanzhao Lin, Jianmin Li, and Shiming Ge. Learning natural consistency representation for face forgery video detection. In *European Conference on Computer Vision*, pages 407–424. Springer, 2024.

[69] Wenliang Zhao, Yongming Rao, Weikang Shi, Zuyan Liu, Jie Zhou, and Jiwen Lu. Diffswap: High-fidelity and controllable face swapping via 3d-aware masked diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8568–8577. IEEE, 2023.

[70] Yinglin Zheng, Jianmin Bao, Dong Chen, Ming Zeng, and Fang Wen. Exploring temporal coherence for more general video face forgery detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15024–15034. IEEE, 2021.

[71] Tianfei Zhou, Wenguan Wang, Zhiyuan Liang, and Jianbing Shen. Face forensics in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5778–5788, June 2021.

[72] Bojia Zi, Minghao Chang, Jingjing Chen, Xingjun Ma, and Yu-Gang Jiang. Wilddeepfake: A challenging real-world dataset for deepfake detection. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 2382–2390, 2020.

A Related Work

A.1 Deepfake Detection Approaches

Frame-Level Detection. These methods focus on spatial artifacts within individual frames. FaceForensics++[44] popularized Xception network [12] as a core benchmark. This architecture uses depthwise separable convolutions to decouple channel wise and spatial correlations, capturing subtle pixel-level texture anomalies. To disentangle semantic content from manipulation traces, later work incorporates spatial frequency domain analysis. F3Net [42] applies Discrete Cosine Transform (DCT) [1] to decompose images into frequency bands and adopts a dual-stream Xception network to learn discrepancies in the frequency domain. Similarly, transformations in the spatial frequency

domain are used to expose subtle high-frequency artifacts [33, 3, 6]. Cheng et al. [9] use progressive transitions to incorporate diverse forgery patterns. However, frame-level methods do not explicitly model temporal dynamics, limiting their ability to capture abnormal temporal patterns across frames.

Video-Level Detection. These methods leverage both spatial appearance and temporal dynamics to determine whether a video has been manipulated. Based on temporal modeling, these methods can be categorized into three groups: Transformer-based, Recurrent-based, and Convolution-based. To reduce the spatial bias of 3D CNNs, Wang et al. [55] propose AltFreezing, which alternately freezes spatial and temporal weights. This encourages the model to capture temporal artifacts, such as flicker, rather than relying solely on spatial artifacts. Nguyen et al. [39] propose FakeSTormer, which adapts TimeSformer [5] to decouple spatio-temporal features. The framework incorporates auxiliary temporal branches to model temporal fragility and pinpoint high-frequency inconsistencies. Choi et al. [11] observe that generative models suppress style latent variance for stability. They introduce a StyleGRU with contrastive learning to capture abnormal temporal evolution. Despite architectural differences, many video-level detectors use temporal-frequency statistics as cues.

A.2 Temporal Frequency Based Approaches

Given its discriminative power, temporal frequency is widely used in deepfake detection and many other tasks. Wu et al. [58] introduce frequency patching to capture abnormal patterns in time-series anomaly detection. Similarly, Wu et al. [56] leverage DCT coefficients from compressed videos for efficient action recognition. Mao et al. [35] apply DCT to temporal trajectories for robust human motion prediction. Although widely used, temporal frequency is a fragile cue-susceptible to temporal corruptions, frequency biases, and adversarial perturbations [65, 66, 34, 7], so models that depend on it are prone to robustness degradation under distribution shifts, indicative of shortcut learning.

A.3 Shortcut Learning

Deep neural networks are prone to shortcut learning [19], in which models rely on spurious correlations to minimize training loss instead of learning robust semantic causality. Consequently, models tend to converge on the most accessible discriminative features, from macroscopic contexts to microscopic statistical irregularities, often at the expense of generalization capability [46, 22, 43, 4, 67]. In deepfake detection, a key shortcut learning issue is over-reliance on temporal spectral artifacts containing multiple temporal-frequency components collectively [18, 16]. Implicit spatiotemporal encoders (e.g., 3D-CNNs) naturally gravitate towards these brittle spectrum statistics and regress to the same rigid dependencies that explicit spectral methods hard code. Rather than modeling robust semantics, these detectors degenerate into narrow band frequency filters, identifying forgeries through specific, unstable spectral fingerprints that real-world perturbations can easily disrupt [20].

B Implementation and Evaluation Protocol

We provide the implementation details required to reproduce SpInShield and the robustness evaluation. All models are trained on FF++ (c23) and evaluated under the same cross-dataset protocol as in the main text. For SpInShield, videos natively encoded at a constant 25 FPS are decoded without resampling. Videos at other frame rates are sampled to 25 FPS by timestamp before face extraction, without re-encoding or recompression. Face regions are detected by RetinaFace and aligned with Dlib landmarks. To avoid introducing artificial temporal jitter, the face bounding box is fixed according to the first frame of each clip. We sample $T = 16$ consecutive frames from 2–4 second video segments and resize each frame to 224×224. During testing, we uniformly sample $K = 10$ clips per video at evenly spaced temporal positions and average their fake probabilities to obtain the video-level score. We report video-level AUC.

SpInShield uses a weight-sharing Siamese VideoMAE V2-Base encoder during training, producing 768-dimensional clip-level features. The backbone is initialized from the official pre-trained checkpoint, followed by global average pooling and a two-layer binary classifier. The LSA generator operates on the normalized one-sided temporal amplitude spectrum computed from $M = 64$ facial ROI patches arranged as an 8×8 grid. For each patch, we compute the temporal mean-intensity signal and apply rFFT, yielding a one-sided amplitude spectrum. It preserves the temporal phase and outputs a frequency-wise modulation field, which is applied using the exponential parameterization

Table 7: Implementation and evaluation protocol.

Item	Setting
Training dataset	FF++ (c23).
Evaluation datasets	CDFv2, DFD, DFo, WDF, DiffSwap, and DaGAN.
Video preprocessing	Native constant-25-FPS videos are decoded without resampling; other frame rates are sampled to 25 FPS by timestamp without re-encoding or recompression. Faces are detected by RetinaFace and aligned by Dlib landmarks. The first-frame bounding box is fixed throughout each clip.
Input clip	$T = 16$ consecutive frames sampled from 2–4 second video segments; input resolution 224×224.
Backbone	VideoMAE V2-Base initialized from the official pre-trained checkpoint; 768-dim clip-level features; global average pooling followed by a two-layer binary classifier.
LSA signal construction	The facial ROI is divided into $M = 64$ patches using an 8×8 grid. For each patch, a temporal mean-intensity signal is transformed by one-sided rFFT. LSA perturbs amplitude while preserving phase and reconstructs the attacked clip by iDFT with Hermitian symmetry.
LSA generator	Lightweight 1D conv network with 3 layers, hidden dimension 64, kernel size 3, LayerNorm, and GELU activation. The output modulation is applied as in Eq. (4), with $\alpha = 0.6$.
Domain discriminator	2-layer MLP with hidden dimension 256, GELU activation, and dropout rate 0.1, applied to clip-level representations through a gradient reversal layer. The GRL coefficient is fixed to 1.0.
MMD term	Gaussian kernel; bandwidth set by the median heuristic on each batch.
Loss weights	$\lambda_{\mathrm{sym}} = 0.9$, $\lambda_{\mathrm{blind}} = 0.7$, generator MMD weight $\gamma = 0.5$, mask regularization weight is set to $\lambda = 0.05$.
Optimization	AdamW optimizer; detector learning rate 1×10^{-5}; generator learning rate 1×10^{-4}; discriminator learning rate 1×10^{-4}; weight decay 0.05; batch size 16; training epochs 20; cosine learning-rate scheduler with 2 warm-up epochs.
Alternating update	For each mini-batch, update G_ϕ for 1 step with the detector frozen, then update the detector and discriminator for 1 step with G_ϕ frozen.
Compute resources	Single NVIDIA A100 GPU; approximately 6 hours per training run.
Spectral stress test	Random band masking, narrow-band notch, spectral tilt, and SNR-based log-amplitude noise. These fixed attacks are used only for evaluation and are not used as training augmentations.
Real-world processing	H.264/H.265 re-encoding, resizing followed by re-encoding, denoising, FPS conversion, and platform round-trip processing. These operations are applied to encoded videos before face extraction and are independent of LSA.
Adaptive attack	White-box attacker optimizes a video-specific amplitude modulation mask to maximize classification loss under norm and smoothness constraints; step size 0.05 in log-amplitude space, restarts 10, perturbation budget $\|\Delta\|_\infty \leq 0.5$.
Testing	LSA generator and domain discriminator are removed. We uniformly sample $K = 10$ clips per video and average clip-level fake probabilities as the video-level score. AUC is computed at the video level.

in Eq. (4). The LSA generator is implemented as a lightweight 1D convolutional network with 3 layers, hidden dimension 64, kernel size 3, LayerNorm, and GELU activation. We use $\alpha = 0.6$, $\lambda_{\mathrm{sym}} = 0.9$, and $\lambda_{\mathrm{blind}} = 0.7$. The MMD term in the generator objective uses a Gaussian kernel with bandwidth determined by the median heuristic on each batch. The domain discriminator is a 2-layer MLP with hidden dimension 256, GELU activation, dropout rate 0.1, and a gradient reversal layer with coefficient 1.0. The generator MMD weight is set to $\gamma = 0.5$, and the mask regularization weight is set to $\lambda = 0.05$. The detector, generator, and domain discriminator are optimized alternately using AdamW. The detector learning rate is 1×10^{-5}, while both the generator and discriminator use a learning rate of 1×10^{-4}. We use weight decay 0.05, batch size 16, and train for 20 epochs with a cosine learning-rate scheduler and 2 warm-up epochs. For each mini-batch, we update G_ϕ for one step with the detector frozen, and then update the detector and domain discriminator for one step with G_ϕ frozen. The detailed settings are summarized in Table 7. All experiments are conducted on a single NVIDIA A100 GPU, with each training run taking approximately 6 hours. At inference time, the LSA generator and domain discriminator are discarded, and only the clean-stream detector is used.

For robustness evaluation, all methods are evaluated on the same source videos. For the spectral stress test in Section 6.3, we first run each detector's own preprocessing pipeline and then apply the temporal spectral transformation directly to its face-ROI input tensor. These fixed hand-crafted transformations are used only for evaluation and are not used as training augmentation. To further decouple evaluation from the training-time LSA distribution, we additionally evaluate real-world video-processing pipelines, including H.264/H.265 re-encoding, resizing followed by re-encoding, denoising, FPS conversion, and black-box platform round-trip processing. These transformations are applied to encoded videos before face extraction and are independent of LSA. We also evaluate an adaptive white-box attack where the attacker directly optimizes a video-specific amplitude modulation mask against the target detector under norm and smoothness constraints. Specifically, we use 10 random restarts, and perturbation budget $\|\Delta\|_\infty \leq 0.5$. We verify that the attack loss converges within the optimization budget for all evaluated methods.

Table 8: Face-cropping and alignment audit on 500 FF++ c23 videos. A clip satisfies the coverage criterion only if all 68 landmarks in all 16 frames lie inside the aligned crop.

Metric	Result
Face detection success rate	100.00%
Landmark detection failure rate	0.00%
Clips satisfying the coverage criterion	99.62%
Mean face-box occupancy in aligned crop	73.38%

Table 9: Mean relative increase in temporal AC power after random frame removal, reported for the eight non-DC DFT bins of 16-frame clips.

DFT bin	1	2	3	4	5	6	7	8
Mean increase (%)	2.39	3.04	3.41	3.98	4.12	4.45	5.78	5.91

B.1 Audit of Face Cropping and Alignment

Our preprocessing fixes the crop bounding box to the first frame of each clip to reduce frame-to-frame detection jitter. Each cropped frame is subsequently aligned independently using Dlib facial landmarks before applying the fixed 8×8 patch grid. The grid therefore operates on approximately aligned facial coordinates.

To assess the stability of this procedure, we audit 500 videos from FF++ c23, comprising 100 real videos and 400 fake videos from the four manipulation methods. We use a strict clip-level coverage criterion: all 68 facial landmarks must remain inside the aligned crop in all 16 frames. We additionally measure the average occupancy of the per-frame RetinaFace bounding box within the final aligned crop.

Table 8 supports stable facial coverage for the short, face-centric clips in the audited setting. Per-frame alignment reduces global translation, rotation, and scale variation, but anatomical correspondence remains approximate under facial deformation, pose changes, and occlusion. Large head motion, profile views, or faces leaving the frame may require temporally smoothed tracking or landmark-guided adaptive cropping.

B.2 Effect of Frame Skipping on Temporal Spectra

Frame skipping may introduce temporal discontinuities and act as a dataset-specific spectral confounder. When the temporal DFT treats retained frames as uniformly spaced samples, a skipped frame causes the next retained frame to be interpreted as an adjacent time step despite the longer actual interval. This can introduce abrupt transitions and redistribute energy across frequency bins.

To examine this effect, we conduct a controlled frame-skipping test on 484 videos from the 1% source subset of DeeperForensics-1.0. For each video, we repeat random frame removal three times and construct 16-frame clips using a sliding window. We compute pixel-wise temporal DFTs on luminance and compare temporal AC power between clips with skipped frames and continuous clips. The median increase in total temporal AC power is 2.12%. Table 9 reports the mean relative power increase for each non-DC frequency bin.

The mean power increases are distributed across all non-DC bins, ranging from 2.39% to 5.91%. This pattern is consistent with broadband spectral changes rather than changes confined to a single narrow frequency band in this controlled setting. Moreover, spectral vulnerabilities are also observed on CDFv2, DFD, DFo, WDF, DiffSwap, and DaGAN, which use different data-generation pipelines. Together, these observations suggest that FF++-specific frame skipping is unlikely to be the sole explanation for the observed vulnerabilities, although its contribution cannot be completely excluded.

Table 10: Perceptual distortion of the four spectral attacks under the evaluation budgets. Arrows indicate the preferred direction. Flow cos. denotes optical-flow cosine similarity between clean and transformed clips.

Transformation	PSNR (dB) ↑	SSIM ↑	LPIPS ↓	Flow cos. ↑
Random Band Masking	46.39	0.9709	0.0122	0.9518
Narrow-band Notch	54.92	0.9960	0.0013	0.9727
Spectral Tilt	56.21	0.9992	0.0003	0.9771
SNR-based Noise	53.41	0.9989	0.0002	0.9799

B.3 Pixel-wise RGB Reconstruction

The patch-mean temporal profiles parameterize the spectral modulation; their reconstructed scalar signals are not directly written back to RGB frames. The aligned 224×224 facial ROI is divided into an 8×8 grid of non-overlapping 28×28 patches. Each pixel belongs to exactly one patch, without overlap averaging or boundary interpolation.

For reconstruction, let $S_{m,i}(\omega_k)$ denote the positive exponential gain in Eq. (4), with the DC gain fixed to $S_{m,i}(0) = 1$. This reconstruction gain is distinguished from the effective ratio $\mathcal{M}_{m,i}$ defined in Eq. (6). For pixel p belonging to patch m and RGB channel c, we compute

$$Z_{i,p,c}(\omega_k) = \mathrm{rFFT}_t\left[X_{\mathrm{clean},i,p,c}\right](\omega_k),$$
$$X_{\mathrm{env},i,p,c}(t) = \mathrm{clip}_{\mathrm{RGB}}(\mathrm{irFFT}_T\left[S_{m,i} \odot Z_{i,p,c}\right](t)).$$

Here, irFFT_T reconstructs a real-valued sequence of length T, and $\mathrm{clip}_{\mathrm{RGB}}$ clamps values to the valid RGB range before the detector's standard input normalization. Each RGB channel is transformed independently, while all pixels and channels within a patch share the same frequency-wise gain.

Before clipping, the positive gains preserve the phase of every nonzero Fourier coefficient, and the unit DC gain preserves each pixel's temporal mean. The pixel-wise reconstruction involves no division or logarithm. Numerical stabilization is used when min-max normalizing the patch-mean amplitude spectrum supplied to the generator. All operations use float32, without intermediate uint8 quantization.

C Additional Robustness Evaluation

C.1 Perceptual Distortion of Spectral Attacks

We quantify the perceptual severity of the four spectral transformations using frame-level PSNR, SSIM, and LPIPS, together with optical-flow cosine similarity between clean and transformed clips. Table 10 reports the measurements under the perturbation budgets used in our robustness evaluation.

SNR-based Noise produces the lowest LPIPS distortion and the highest flow cosine similarity, yet leads to lower baseline AUCs than Random Band Masking (Table 2). This observation suggests that the performance degradation cannot be explained solely by the severity captured by these distortion metrics. The measurements support limited appearance and motion distortion under the evaluated budgets, but do not establish strict imperceptibility or preservation of all task-relevant forensic evidence.

C.2 Evaluation on Talking-Face and In-the-Wild Forgeries

To broaden the evaluation beyond the original benchmarks, we additionally compare SpInShield with DFD-FCG on FFIW [71] and a targeted talking-face subset of DF40 [62]. The selected DF40 subset includes Wav2Lip, SadTalker, and HeyGen, and is denoted as DF40-TalkingFace-3. We evaluate both clean performance and robustness under the same spectral perturbation setting for the two detectors.

As shown in Table 11, SpInShield achieves comparable clean performance to DFD-FCG on both benchmarks. Under spectral perturbations, SpInShield improves AUC over DFD-FCG by 19.1 percentage points on DF40-TalkingFace-3 and 17.7 percentage points on FFIW. These results provide additional evidence of robustness in talking-face generation and in-the-wild multi-person forgery

Table 11: Clean and perturbed performance on the selected DF40 talking-face subset and FFIW. All results are AUC (%).

Benchmark	Setting	DFD-FCG	SpInShield
DF40-TalkingFace-3	Clean	90.7	**91.0**
DF40-TalkingFace-3	Perturbed	67.0	**86.1**
FFIW	Clean	**91.6**	91.4
FFIW	Perturbed	69.2	**86.9**

Table 12: Clean cross-dataset generalization with a matched VideoMAE V2 backbone. Both models use the same pretrained initialization and matched training settings. Results are average AUC (%) on clean OOD data.

Model	Clean OOD Avg. AUC (%)
Plain VideoMAE V2	78.1
SpInShield	**93.3**

scenarios. Our evaluation remains focused on face-centric video forgeries with aligned facial regions; fully generated open-world text-to-video content remains outside its scope.

C.3 Clean Generalization with a Matched Backbone

To distinguish the contribution of our training framework from that of backbone pretraining, we evaluate a plain VideoMAE V2 baseline under matched settings. The baseline and SpInShield use the same pretrained initialization, training split, preprocessing, clip length, classifier architecture, optimizer, learning-rate schedule, number of epochs, batch size, and evaluation protocol. The plain baseline excludes the LSA-generated spectral view and the two invariance objectives, $\mathcal{L}_{\mathrm{sym}}^{\mathrm{inv}}$ and $\mathcal{L}_{\mathrm{blind}}^{\mathrm{inv}}$.

Table 12 shows that SpInShield improves clean OOD average AUC by 15.2 percentage points over the matched plain backbone. This comparison indicates that backbone pretraining alone does not account for the observed gains, and supports the contribution of the proposed spectral augmentation and invariance training. These results concern clean cross-dataset evaluation, whereas the OOD results in the component ablation (Table 5b) are averaged under the four spectral attacks.

C.4 Robustness to Pixel-Space Adversarial Attacks

To evaluate robustness beyond the LSA perturbation family, we conduct untargeted white-box pixel-space PGD attacks that are not used during training. Each model is attacked separately on its preprocessed clips using 20 optimization steps with random initialization. We evaluate perturbation budgets of $2/255$ and $4/255$ and compare SpInShield with DFD-FCG.

As shown in Table 13, SpInShield retains higher AUC at both perturbation budgets, exceeding DFD-FCG by 11.9 percentage points at $2/255$ and 7.9 percentage points at $4/255$. These results provide evidence that its robustness gains extend beyond amplitude-only spectral perturbations to the evaluated pixel-space attacks. Nevertheless, performance decreases substantially as the attack budget increases, indicating that vulnerability to white-box adversarial perturbations remains.

C.5 Effect of Temporal Smoothing during Training

To examine the influence of temporal smoothing on spectral robustness, we retrain the evaluated models on temporally smoothed FF++ while keeping the other training settings unchanged. We compare their perturbed OOD performance with that of models trained on the original FF++ videos.

As shown in Table 14, temporal smoothing improves the baseline AUCs by 1.4–3.7 percentage points, suggesting that it affects their learned temporal-frequency dependencies. However, their robustness remains substantially below that of SpInShield, whose AUC changes only slightly from 88.9 to

Table 13: White-box pixel-space PGD evaluation. Each detector is attacked separately using 20 steps with random initialization. Results are average AUC (%) under clean inputs and the two perturbation budgets.

Model	Clean	PGD $(2/255)$	PGD $(4/255)$
DFD-FCG	91.7	67.3	57.1
SpInShield	**93.3**	**79.2**	**65.0**

Table 14: Effect of temporal smoothing of the training videos. Each model is trained on either original or temporally smoothed FF++ under otherwise identical settings. Results are perturbed OOD average AUC (%).

Model	Original FF++	Smoothed FF++
VideoMAE(B)	57.8	60.4
FakeST	63.5	67.2
DFD-FCG	65.5	67.1
SLF	67.6	69.0
SpInShield	**88.9**	**89.2**

89.2. These results suggest that temporal smoothing of the training data alone does not account for SpInShield's robustness advantage under spectral perturbations.

C.6 Real-world video processing pipelines.

The spectral stress test in Section 6.3 is designed to isolate temporal-spectrum shortcut learning under controlled amplitude perturbations. However, real deployment scenarios usually involve non-differentiable and implementation-dependent video processing operations, such as platform transcoding, resizing, denoising, frame-rate conversion, and repeated compression. To evaluate the generalization of SpInShield in real-world applications, we further consider four real-world processing settings: (1) We evaluate Black-Box platform round-trip processing, where videos are uploaded to YouTube and TikTok and then downloaded after platform-side transcoding. (2) We construct a deterministic Multi-Stage Codec-Chain benchmark, which combines resizing, denoising, frame-rate conversion, and repeated H.264/H.265 re-encoding with different severity levels. (3) We evaluate a Heavy-compression variant of the Codec-Chain (CRF=33, $25 \rightarrow 10 \rightarrow 25$ FPS, three-pass H.264/H.265 with denoising) to examine whether robustness holds under aggressive spatiotemporal information loss. (4) we evaluate a Cross-Platform Cascade, where each video undergoes two consecutive real platform round-trips (YouTube followed by WhatsApp-style compressed sharing) before face extraction, simulating the multi-hop dissemination path commonly observed in social-media propagation of forged content. Unlike the amplitude-spectrum attacks used in the main stress test, these transformations are applied to encoded videos before face extraction and are never used during training. Therefore, this evaluation provides a distributionally separated test of robustness under realistic video post-processing pipelines.

Discussion of Real-world Pipeline Results. Table 15 reports the robustness of different detectors under realistic video processing pipelines. In the black-box platform round-trip setting in Table 15(a), SpInShield achieves the best average AUC of 89.92%, outperforming the strongest baseline (SLF) by 6.81 points. While individual baselines remain competitive on specific subsets—for example, DFD-FCG on DFD (90.84), TALL on DFo (93.62), FTCN on WDF (84.35), PTFD on DiffSwap (87.20), and SLF on DaGAN (85.20)—no baseline maintains balanced performance across all six datasets. SpInShield is the only method that performs uniformly well, with per-dataset AUCs all at or above 86%, indicating that the proposed phase-consistent representation is less dependent on dataset-specific compression artifacts and transfers consistently across heterogeneous post-processing pipelines.

In the standard multi-stage codec-chain setting in Table 15 (b), SpInShield further achieves 87.53% average AUC, surpassing the strongest baseline (PTFD) by 8.81 points. This setting introduces repeated compression and re-encoding, which weakens many amplitude- or frequency-shortcut based methods. SpInShield's advantage is especially pronounced on the harder OOD subsets: it improves

Table 15: Robustness under real-world video processing pipelines. * indicates statistically significant improvements over the strongest baseline via a two-tailed t-test ($p < 0.05$) on the average metric.

Method	CDFv2	DFD	DFo	WDF	DiF.	DaG.	Avg.	CDFv2	DFD	DFo	WDF	DiF.	DaG.	Avg.
	(a) Black-Box Platform Round-trip							(b) Multi-stage Codec-Chain						
FakeST	82.66	90.32	90.91	74.49	79.55	76.35	82.38	76.50	85.20	89.50	73.35	76.30	64.33	77.53
AltFre	79.36	87.56	88.35	72.15	77.94	78.42	80.63	76.18	83.62	86.83	67.34	72.11	72.14	76.37
PTFD	84.72	83.90	87.60	70.96	87.20	83.44	82.97	77.26	80.14	82.62	70.79	84.20	77.31	78.72
TALL	79.36	86.17	93.62	68.45	71.11	71.21	78.32	75.96	80.36	91.30	63.72	65.50	67.10	73.99
SLF	83.06	86.29	91.00	74.09	79.02	85.20	83.11	85.30	82.00	82.50	72.85	72.64	76.19	78.58
NACO	84.22	89.80	90.48	73.27	72.88	76.73	81.23	79.50	88.50	87.50	65.50	70.47	70.65	77.02
AFW	78.71	79.05	77.74	65.23	70.68	70.97	73.73	71.63	76.17	75.83	61.23	63.21	66.89	69.16
DFD-FCG	78.20	90.84	75.65	66.86	66.94	77.99	76.08	76.55	87.18	79.34	67.77	69.15	71.63	75.27
FTCN	77.18	82.62	86.03	84.35	76.97	78.97	81.02	77.06	82.48	83.76	70.74	73.00	72.20	76.54
VideoMAE(B)	73.78	79.43	71.69	66.03	67.32	67.39	70.94	67.72	75.92	65.71	59.06	62.07	63.36	65.64
SpInShield	**92.27**	**91.09**	**95.44**	**86.10**	**87.88**	**86.74**	**89.92** *	**87.79**	**88.59**	**92.99**	**85.15**	**84.80**	**85.86**	**87.53** *
	(c) Heavy Codec-Chain							(d) Cross-Platform Cascade						
FakeST	73.91	79.16	79.87	61.91	67.58	66.93	71.56	76.34	83.08	85.72	64.96	73.50	69.58	75.53
AltFre	71.22	75.80	80.19	61.35	65.96	69.44	70.66	75.01	78.64	82.89	66.30	69.20	70.22	73.71
PTFD	71.37	75.78	80.20	70.27	69.39	69.85	72.81	75.46	83.13	82.92	65.38	74.55	76.18	76.27
TALL	70.50	72.70	84.50	57.40	60.81	59.21	67.52	72.20	80.70	80.90	66.66	63.88	64.48	71.47
SLF	70.50	74.80	76.00	73.60	70.50	72.30	72.95	75.88	83.05	84.50	67.96	71.21	75.26	76.31
NACO	71.41	79.81	82.17	57.79	65.74	68.42	70.89	74.58	79.73	87.30	66.90	68.22	69.85	74.43
AFW	63.74	69.25	69.25	56.99	60.57	62.46	63.71	71.03	75.82	69.93	58.55	61.10	64.61	66.84
DFD-FCG	75.13	80.25	80.46	65.72	71.49	79.23	75.38	74.00	86.20	81.24	62.55	70.06	79.61	75.61
FTCN	71.37	75.54	81.14	62.00	64.20	72.77	71.17	73.46	75.22	80.10	75.26	70.40	71.60	74.34
VideoMAE(B)	61.05	71.95	54.86	52.89	56.12	55.93	58.80	66.68	74.90	64.49	58.90	59.96	60.51	64.24
SpInShield	**86.85**	**84.64**	**87.91**	**79.27**	**80.85**	**80.16**	**83.28** *	**85.84**	**86.58**	**87.84**	**80.35**	**83.15**	**82.88**	**84.44** *

the next-best result by 11.80 points on WDF (85.15 vs. FakeST 73.35) and by 8.55 points on DaGAN (85.86 vs. PTFD 77.31). On the easier in-distribution-style subsets, SpInShield also attains the highest score, leading the strongest baseline by 2.49 points on CDFv2 (87.79 vs. SLF 85.30), 1.69 points on DFo (92.99 vs. TALL 91.30), and 0.60 points on DiffSwap (84.80 vs. PTFD 84.20). This pattern indicates that SpInShield does not sacrifice performance on standard subsets to gain robustness on harder ones, but instead maintains the most balanced performance across the full evaluation suite.

Under the heavy codec-chain in Table 15(c), all methods degrade compared with the standard codec-chain because aggressive compression and frame-rate reduction remove substantial high-frequency spatiotemporal evidence. SpInShield decreases from 87.53% to 83.28%, but still preserves the highest average AUC and exceeds the strongest baseline (DFD-FCG) by 7.90 points. In comparison, VideoMAE(B) drops from 65.64% to 58.80%, showing that generic video representations are more sensitive to severe codec degradation. The advantage of SpInShield is especially pronounced on CDFv2 (86.85 vs. DFD-FCG 75.13, +11.72) and DiffSwap (80.85 vs. DFD-FCG 71.49, +9.36), and it also leads the strongest baseline on the remaining subsets, by 4.39 points on DFD (84.64 vs. DFD-FCG 80.25), 3.41 points on DFo (87.91 vs. TALL 84.50), 5.67 points on WDF (79.27 vs. SLF 73.60), and 0.93 points on DaGAN (80.16 vs. DFD-FCG 79.23). This consistency across subsets suggests that phase-consistent forensic cues retain discriminative information even when amplitude statistics are heavily corrupted.

SpInShield achieves 84.44% average AUC, outperforming the strongest baseline (SLF) by 8.13 points, and obtains the highest score on all six datasets, including CDFv2 (85.84, +9.50 over FakeST), WDF (80.35, +5.09 over FTCN), DiffSwap (83.15, +8.60 over PTFD), and DaGAN (82.88, +3.27 over DFD-FCG). On DFD and DFo, it also outperforms the strongest baselines by 0.38 points (86.58 vs. DFD-FCG 86.20) and 0.54 points (87.84 vs. NACO 87.30), respectively. These results suggest that phase-based forensic signals are more stable under repeated transcoding, whereas methods relying on platform- or codec-specific amplitude fingerprints degrade more substantially when those fingerprints are overwritten by successive re-encoding.

C.7 Adaptive Evaluation against SpInShield

The fixed spectral stress test in Section 6.3 evaluates whether detectors are robust to hand-crafted amplitude-spectrum perturbations. Although this setting is useful for diagnosing temporal-spectrum shortcut learning, it does not fully capture an adaptive attacker who is aware of the defense mechanism. We therefore conduct an adaptive white-box evaluation, where the attacker has full access to the target detector, preprocessing pipeline, LSA formulation, and training objectives. Instead of sampling

Table 16: Adaptive white-box evaluation. Adaptive Spectral denotes per-video optimization of an amplitude modulation mask against each target detector. * indicates statistically significant improvements over the strongest baseline via a two-tailed t-test ($p < 0.05$) on the average metric.

Method	CDFv2	DFD	DFo	WDF	DiffSwap	DaGAN	Avg.
FakeST	54.27	58.83	60.14	47.62	48.91	47.36	52.86
AltFre	52.48	57.91	61.73	48.27	50.43	53.18	54.00
PTFD	55.16	59.42	60.87	48.94	54.73	57.62	56.12
TALL	51.83	56.74	64.28	46.39	47.21	47.84	52.38
SLF	55.74	60.18	62.45	49.61	51.36	56.27	55.94
NACO	54.62	62.37	63.81	47.83	49.74	50.41	54.80
AFW	49.27	53.86	51.74	44.92	45.83	47.62	48.87
DFD-FCG	60.43	65.78	64.92	53.47	55.86	56.21	59.45
FTCN	53.18	57.46	60.83	48.74	51.92	53.47	54.27
VideoMAE(B)	47.83	53.62	45.91	43.27	44.58	45.18	46.73
SpInShield	**67.74**	**66.42**	**72.86**	**61.83**	**63.27**	**63.91**	**66.01** *

perturbations from the fixed benchmark, the attacker directly optimizes a video-specific amplitude modulation mask against each target detector. Given an input video X with label y, the attacker preserves the temporal phase and optimizes an amplitude-domain perturbation Δ in the log-amplitude space. The attacked video is generated as $\mathcal{T}_\Delta(X)$, and the perturbation is optimized by maximizing the classification loss under norm and smoothness constraints:

$$\max_{\Delta} \mathcal{L}_{\mathrm{CE}}\left(f_\theta(\mathcal{T}_\Delta(X)), y\right) - \lambda_{\mathrm{smooth}}\|\nabla_\omega \Delta\|_2^2 - \lambda_{\mathrm{norm}}\|\Delta\|_2^2, \tag{13}$$

where $\nabla_\omega \Delta$ penalizes abrupt frequency-wise changes and $\|\Delta\|_2^2$ constrains the overall attack magnitude. We verify that the attack loss converges within the optimization budget for all evaluated methods. All methods are attacked independently using the same optimization budget, and AUC is computed at the video level using the same testing protocol as in the main experiments. The results are reported in Table 16.

Table 16 shows that adaptive optimization substantially reduces the performance of all detectors compared with non-adaptive evaluation. Specifically, baselines without explicit spectral-invariance training drop close to chance-level performance (around 47-56% average AUC), confirming that fixed-pattern detectors collapse when the attacker can directly target their decision boundaries. Even strong baselines such as DFD-FCG degrade by over 30 points compared with their clean performance, indicating that adaptive optimization is a fundamentally stronger test than fixed hand-crafted perturbations. SpInShield also exhibits a notable performance drop under this stronger attack (from 93.3% on clean videos to 66.01% under adaptive attack), reflecting the inherent difficulty of defending against fully white-box adversaries. Nevertheless, SpInShield achieves the best average AUC and consistently outperforms all baselines by a clear margin (+6.56 points over the strongest baseline DFD-FCG). This suggests that the learned spectral-invariant representation provides robustness that generalizes beyond the fixed LSA perturbation distribution encountered during training, rather than merely memorizing it. We acknowledge that adaptive white-box attacks remain a challenging open problem and view the residual gap to clean-data performance as a direction for future work.

D Limitations

While SpInShield demonstrates strong empirical performance across multiple robustness benchmarks, several limitations remain. Although SpInShield substantially outperforms baselines under the adaptive white-box evaluation, it still exhibits a notable performance drop relative to its clean-data accuracy, indicating that shortcut suppression mitigates but does not fully eliminate the threat posed by an attacker with full knowledge of the defense. Preserving the Fourier phase of nonzero coefficients before clipping does not guarantee preservation of high-level facial-motion semantics, since amplitude reweighting and clipping can alter useful forensic evidence. Our evaluation primarily focuses on face-swap and face-reenactment forgeries, and whether the temporal-spectrum shortcut hypothesis transfers to fully generative video synthesis paradigms (e.g., text-to-video diffusion models) remains an open question. Finally, SpInShield operates exclusively on the visual modality and does not exploit

complementary cues such as audio-visual synchronization, which may offer further robustness in multi-modal forgery scenarios. We leave these directions to future work.

E Broader Impacts

SpInShield aims to improve the reliability of video deepfake detection under temporal spectral perturbations and common video-processing operations, potentially supporting media verification and reducing the harms of manipulated content. However, the spectral perturbation techniques studied in this work could also be misused to evade detection systems. Detection errors remain consequential: false positives may undermine trust in authentic content, while false negatives may allow manipulated videos to circulate. Performance may vary across populations, recording conditions, and manipulation methods beyond the evaluated benchmarks. Practical deployment should therefore include human review, validation under the intended operating conditions, and procedures for correcting erroneous decisions.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The main claims in the abstract and introduction are aligned with the paper's contributions: identifying temporal spectral shortcut learning in spatiotemporal deepfake detectors and proposing SpInShield to mitigate this vulnerability. These claims are supported by the empirical analysis in Section 4, the proposed method in Section 5, and the experimental results in Section 6.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: We discuss the limitations in appendix.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

Answer: [N/A]

Justification: The paper does not include theoretical results.

Guidelines:

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

4. **Experimental result reproducibility**

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

Answer: [Yes]

Justification: The paper describes the datasets, preprocessing pipeline, evaluated baselines, attack generation procedures, model architecture, and optimization objectives needed to reproduce the main experimental results. Additional implementation details such as exact training hyperparameters and environment settings are provided in the appendix.

Guidelines:

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

some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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

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

   Answer: [No]

   Justification: We currently do not provide open access to the full code and processed data needed to reproduce all experimental results. We rely on publicly available benchmark datasets and official baseline implementations where available, and plan to release code and reproduction instructions when permitted.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: Section 6.1 specifies the datasets, preprocessing steps, temporal sampling strategy, evaluation datasets, and baseline methods used in the experiments. The attack construction details are further described in Section 6.3, while the model architecture and objectives are given in Section 5.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: Tables 1 and 2 report statistically significant improvements over the strongest baseline using two-tailed t-tests ($p < 0.05$) on the average AUC.

   Guidelines:

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: We provide sufficient details about the compute workers, memory, runtime, or total compute required for each experiment in appendix.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: The research is intended to improve the robustness of deepfake detection systems and uses existing benchmark datasets and evaluation protocols. To the best of our knowledge, the study does not involve deception of participants, collection of new personal data, or procedures that violate the NeurIPS Code of Ethics.

   Guidelines:

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

10. **Broader impacts**

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

Answer: [Yes]

Justification: Appendix E discusses the potential benefits for media verification, misuse of spectral perturbations, harms arising from detection errors, and considerations for responsible deployment.

Guidelines:

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

11. **Safeguards**

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

Answer: [N/A]

Justification: The paper does not release a generative model, scraped dataset, or other high-risk asset that could directly enable deepfake generation or misuse. The proposed method is a detection and robustness framework, so this question is not applicable.

Guidelines:

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

12. **Licenses for existing assets**

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

Answer: [No]

Justification: The paper cites existing datasets and methods, but does not provide a complete asset-by-asset inventory of licenses and terms of use for the datasets, pretrained weights, and third-party code used in the experiments.

Guidelines:

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

13. **New assets**

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

Answer: [N/A]

Justification: The paper does not release a new dataset, benchmark, model checkpoint, or code package as a standalone asset in the current submission.

Guidelines:

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

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

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

Answer: [N/A]

Justification: The paper does not involve crowdsourcing experiments or newly conducted research with human subjects. All experiments are conducted on existing deepfake detection benchmarks and generated evaluation sets derived from existing datasets.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

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

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

    Answer: [N/A]

    Justification: The paper does not involve newly recruited human subjects, user studies, crowdsourcing experiments, or collection of new personal data. Therefore, IRB approval or equivalent human-subjects review is not applicable.

    Guidelines:

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

16. **Declaration of LLM usage**

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

    Answer: [N/A]

    Justification: The core method development, experiments, and analysis in this paper do not use LLMs as an important, original, or non-standard component. Any use of LLMs for writing, editing, or formatting does not affect the methodology, scientific rigor, or originality of the research.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.