Video Forensic Self-Descriptions: Leveraging Temporally Distributed Forensic Microstructures for Zero-Shot Detection of AI-Generated Videos

Aref Azizpour[*]
Drexel University
aa4639@drexel.edu

Tai D. Nguyen[*]
Drexel University
tdn47@drexel.edu

Matthew C. Stamm
Drexel University
mcs382@drexel.edu

Abstract

AI-generated videos are becoming increasingly realistic, yet most detection methods focus on spatial artifacts within individual frames and require synthetic training data from known generators, causing them to fail on unseen generators. We identify two previously unexplored *temporally distributed forensic microstructures* that reveal whether a video was AI-generated. These traces arise from how a video's visual field and motion evolve over time, capturing subtle deviations that generators introduce when approximating real-world temporal dynamics. To expose them, we train two simple predictors on real videos only, modeling visual evolution (VE-FSD) and motion evolution (ME-FSD), and analyze their prediction residuals. We then fit 3D spatio-temporal autoregressive models to these residuals, producing compact descriptors called Video Forensic Self-Descriptions (VFSD). VFSDs of real videos naturally cluster together and separate from AI-generated ones, enabling both zero-shot detection without synthetic training data and source attribution. Experiments across multiple datasets and 35 generators demonstrate that VFSD achieves state-of-the-art performance on both tasks.

Project page and code: https://arefazizpour.github.io/VFSD-NeurIPS2026/

1 Introduction

Modern AI video generators [1–5] produce videos with convincing visual appearance, complex motion, and coherent scene dynamics, making them increasingly difficult for humans to distinguish from real recordings. While enabling creative applications, these technologies introduce risks, as AI-generated videos can be misused for financial fraud, impersonation attacks, and disinformation campaigns [6, 7]. These risks have motivated the development of forensic techniques for detecting AI-generated videos. In this work, we focus exclusively on fully synthesized videos (e.g., those produced by text-to-video models), distinct from partially manipulated media such as face-swaps or localized deepfakes.

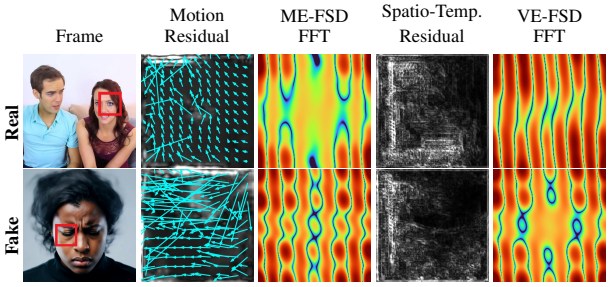


Figure 1: Real and AI-generated videos show distinct temporally distributed forensic microstructures.

Most existing detectors rely on supervised learning [8–10], training on real and synthetic videos from known generators to learn forensic traces associated with those generators. However, this

[*]Equal contribution.

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

paradigm suffers from a critical limitation. Detectors trained on a fixed set of generators often fail to generalize to unseen generators [11, 12]. Because new models are introduced rapidly and existing ones are frequently updated [13, 14], detectors tied to a fixed training set can experience significant performance degradation in practice.

Existing approaches rely on spatially distributed forensic traces within individual frames [15–18], while temporally distributed forensic evidence, inherent to a video's evolution, has received less attention [19]. Real-world videos are governed by physical processes that impose statistical structure on both visual scene evolution and motion dynamics. AI video generators attempt to approximate these dynamics [20, 4, 21], but cannot perfectly reproduce the underlying processes, causing subtle but structured deviations in the temporal evolution of generated videos.

We identify two *temporally distributed forensic microstructures* that can be used to detect AI-generated videos. The first arises from how a video's visual field evolves over time, capturing inconsistencies in the temporal evolution of visual appearance. The second arises from how motion evolves over time, capturing differences in motion dynamics produced by generative models compared to real videos. To expose these microstructures, we train simple predictors exclusively on real videos to forecast future frames and motion fields from past observations. When applied to real videos, the prediction residuals primarily reflect stochastic variations in real-world dynamics. When applied to AI-generated videos, the residuals additionally contain structured patterns introduced by the generative process, revealing the temporally distributed forensic microstructures.

To leverage these signals, we introduce Video Forensic Self-Description (VFSD) as a framework for constructing compact representations of the temporally distributed forensic microstructures within a video. Our approach is inspired by forensic self-descriptions for images [22], which characterize spatial forensic traces using parametric models. We extend this to video, modeling prediction residuals as stochastic processes and fitting spatio-temporal autoregressive models to characterize their correlation structure. The resulting parameters form a compact descriptor summarizing the temporal forensic microstructures.

We demonstrate that VFSD enables both zero-shot detection of AI-generated videos *without ever seeing synthetic data* and source generator attribution. Extensive experiments across multiple real datasets and 35 generators, including Sora [1], Veo [2], and Midjourney [23], show that our approach achieves state-of-the-art performance. Our main contributions are summarized as follows:

- We identify two unexplored temporally distributed forensic traces, one in visual evolution and one in motion evolution, that reveal whether a video is AI-generated. These traces are exposed by modeling a video as a dynamic system and analyzing prediction residuals.
- We propose Video Forensic Self-Descriptions (VFSD), compact parametric models of these traces built using 3D spatio-temporal autoregressive models fitted to residuals from predictors trained on real videos only.
- We show that VFSDs of real videos naturally cluster away from those of synthetic videos, enabling both zero-shot detection without synthetic training data and source generator attribution.
- We conduct extensive experiments across multiple datasets and 35 generators, demonstrating state-of-the-art performance for both zero-shot detection and source attribution.

2 Background

Synthetic Video Generation. Video generation has progressed from GANs [24, 25] and VAEs [26] through temporal extensions of image diffusion models [27, 20, 28, 29] to latent video diffusion [4, 30]. Recent text-to-video systems [1–3, 31, 5, 32–35] now produce content that is difficult to distinguish from real footage. Despite diverse architectures, all learn to approximate, rather than replicate, the temporal dynamics of physical scenes.

Synthetic Image Detection. Supervised image detectors [15, 17, 36, 18] train on real and synthetic data but struggle to generalize to unseen generators. Generalization-oriented approaches analyze pixel relationships [16], reconstruction errors [37, 38], spectral properties [39, 40], fingerprints [41], or CLIP features [42]. Most relevant to our work, Nguyen *et al*. [22] proposed Forensic Self-Descriptions (FSD), fitting autoregressive models to spatial prediction residuals for zero-shot detection without

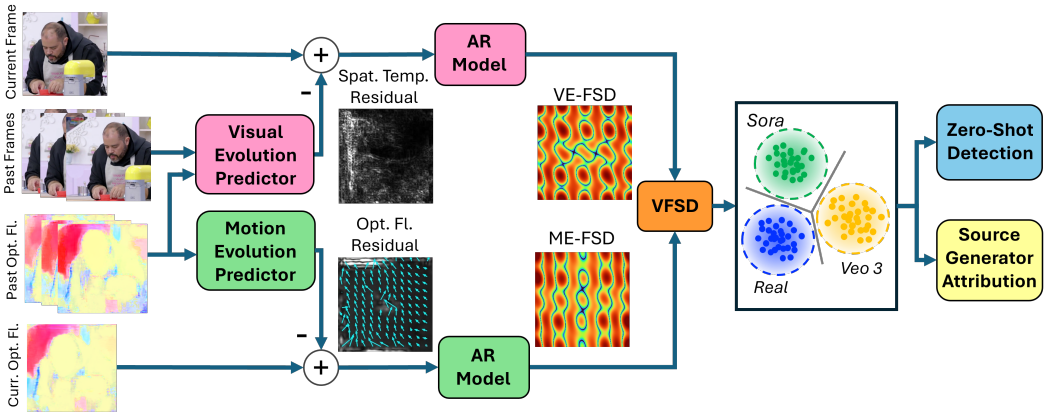


Figure 2: VFSD extracts spatio-temporal and motion residuals via real-trained predictors, then fits 3D AR models to produce self-descriptions for detection and attribution.

synthetic training data. However, Vahdati *et al*. [19] showed that image-level detectors fail on AI-generated videos because video generators leave fundamentally different traces.

Synthetic Video Detection. Early video forensic methods targeted face-swap deepfakes [43–45, 9, 46]. Recent supervised methods address fully synthesized video using temporal convolutions [8], state-space models [10], AI-generated features [47], or universal detection frameworks [48]. Training-free approaches include D3 [49], which computes second-order statistics of deep features. However, supervised methods remain tied to training generators, and existing training-free methods rely on summary statistics of pretrained features rather than explicitly modeling the spatio-temporal correlation structure of forensic traces [50, 51].

3 Proposed Approach

Prior work has shown that spatially distributed forensic microstructures are introduced into video frames as a consequence of a generator's architecture and synthesis pipeline [52–54, 11]. We demonstrate the existence of a complementary class of forensic evidence: *temporally distributed forensic microstructures*, traces that emerge not from frame formation but from how a scene evolves over time. Below, we describe how these microstructures can be identified by modeling the video scene as a dynamic system, compactly parameterized, and leveraged for zero-shot detection and source attribution. An overview is provided in Figure 2.

3.1 Temporal Evolution and Forensic Microstructures

We model a video scene as a dynamic system characterized by two evolving quantities: the visual field, captured by each video frame I_t, and the motion, estimated by the optical flow O_t between consecutive frames. We define the past L observations of the visual field and optical flow as $\mathbf{I}_{t-1}^{(L)} = \{I_{t-1}, \ldots, I_{t-L}\}$ and $\mathbf{O}_{t-1}^{(L)} = \{O_{t-1}, \ldots, O_{t-L}\}$.

Real video as a dynamic system. Under this model, we assume that there exist unknown system functions h_I and h_O that govern the temporal evolution of the visual field and its motion:

$$I_t = h_I(\mathbf{I}_{t-1}^{(L)}, \mathbf{O}_{t-1}^{(L)}) + \sigma_t^I, \qquad O_t = h_O(\mathbf{O}_{t-1}^{(L)}) + \sigma_t^O \tag{1}$$

where h_I and h_O govern the temporal evolution of a scene (*e.g.*, object motion and lighting changes) and σ_t^I, σ_t^O are i.i.d. white noise terms capturing imaging noise and scene innovations.

While h_I and h_O are unknown, we can train predictive models f_I and f_O to estimate the visual field and optical flow from past observations:

$$\hat{I}_t = f_I(\mathbf{I}_{t-1}^{(L)}, \mathbf{O}_{t-1}^{(L)}), \qquad \hat{O}_t = f_O(\mathbf{O}_{t-1}^{(L)}) \tag{2}$$

Examining the prediction residuals, we find

$$r_t^I = I_t - \hat{I}_t = \epsilon_t^I + \sigma_t^I, \qquad r_t^O = O_t - \hat{O}_t = \epsilon_t^O + \sigma_t^O \tag{3}$$

where $\epsilon_t^I = h_I(\cdot) - f_I(\cdot)$ and $\epsilon_t^O = h_O(\cdot) - f_O(\cdot)$ are stochastic processes capturing the model mismatch between the predictors and the true system functions.

Synthetic video as an imperfect approximation. A video generator learns its own approximations $g_I(\cdot)$ and $g_O(\cdot)$ of $h_I(\cdot)$ and $h_O(\cdot)$ from real-world training data. When we apply the same predictors f_I and f_O to a generated video, the residuals become:

$$\tilde{r}_t^I = \tilde{\epsilon}_t^I + \tilde{\sigma}_t^I, \qquad \tilde{r}_t^O = \tilde{\epsilon}_t^O + \tilde{\sigma}_t^O \tag{4}$$

where $\tilde{\epsilon}_t^I = g_I(\cdot) - f_I(\cdot)$ and $\tilde{\epsilon}_t^O = g_O(\cdot) - f_O(\cdot)$. Because no generator perfectly replicates h_I and h_O, the mismatch $g - f$ differs systematically from $h - f$, making real and synthetic residuals statistically distinguishable. Moreover, because each generator's approximation depends on its architecture, training data, and optimization, different generators produce different mismatch signatures. This is demonstrated in our experiments below.

We term these distinct residual patterns **temporally distributed forensic microstructures**, comprising **spatio-temporal traces** in r_t^I and **motion traces** in r_t^O. These microstructures constitute a new class of forensic evidence that we leverage for zero-shot detection and source attribution.

3.2 Self-Supervised Spatio-Temporal Residual Extraction

To extract these microstructures, we train two self-supervised linear convolutional predictors exclusively on real videos, one for visual content and one for motion. Each predictor learns to forecast the next state from previous ones, allowing us to leverage large-scale unlabeled video data. Training only on real data ensures the predictors learn a model of the temporal evolution of real videos without dependence on any generator.

Visual Evolution Predictor (f_I). We predict the next frame I_t from the immediately preceding frame and optical flow (L=1), since the flow already provides a first-order summary of temporal evolution, describing where each pixel will move, and gives the predictor an initial estimate to refine.

We concatenate (denoted $\oplus$) $I_{t-1} \in \mathbb{R}^{H\times W\times 3}$ and optical flow $O_{t-1} \in \mathbb{R}^{H\times W\times 2}$ (estimated via RAFT [55]) along the channel dimension to form $X_{t-1} = [I_{t-1} \oplus O_{t-1}] \in \mathbb{R}^{H\times W\times 5}$, and parameterize f_I as a linear convolution over a $B \times B$ spatial neighborhood $\mathcal{B}$ with weights $w_I \in \mathbb{R}^{B\times B\times 5\times 3}$:

$$\hat{I}_t(x, y, c) = \sum\nolimits_{(i,j)\in\mathcal{B}} \sum\nolimits_{\ell=1}^{5} w_I(i, j, \ell, c)\; X_{t-1}(x-i, y-j, \ell) \tag{5}$$

where $c \in \{1, 2, 3\}$ indexes the output RGB channels and ℓ indexes the input channels. The weights are optimized to minimize the total residual energy over a training set of N real video frames:

$$w_I^* = \operatorname*{argmin}_{w_I} \sum\nolimits_{n=1}^{N} \sum\nolimits_t \sum\nolimits_{x,y} \|I_t^{(n)}(x, y) - \hat{I}_t^{(n)}(x, y)\|_2^2 \tag{6}$$

The resulting **spatio-temporal residual** $r_t^I = I_t - \hat{I}_t$ measures, at each spatial location, how the actual visual evolution from $t-1$ to t deviates from the predicted evolution. The difference in forensic microstructures between real and generated video is visible in the bottom row of Figure 3, which visualizes the frequency response of AR coefficients fitted to these residuals. In this figure, we observe that real video produces broad, low-frequency structure, while each generator introduces distinct higher-frequency patterns.

Motion Evolution Predictor (f_O). Independently, f_O models how the motion field evolves, capturing anomalies in dynamics such as acceleration and direction shifts. Since no first-order approximation analogous to optical flow exists for motion evolution, f_O conditions on L consecutive flow fields. We concatenate them into $X_t^O = [O_{t-1} \oplus \cdots \oplus O_{t-L}] \in \mathbb{R}^{H\times W\times 2L}$ and parameterize f_O as a linear convolution over a $D \times D$ neighborhood $\mathcal{D}$ with weights $w_O \in \mathbb{R}^{D\times D\times 2L\times 2}$:

$$\hat{O}_t(x, y, c) = \sum\nolimits_{(i,j)\in\mathcal{D}} \sum\nolimits_{\ell=1}^{2L} w_O(i, j, \ell, c)\; X_t^O(x-i, y-j, \ell) \tag{7}$$

where $c \in \{1, 2\}$ indexes the output flow channels. The weights are optimized over real videos:

$$w_O^* = \operatorname*{argmin}_{w_O} \sum\nolimits_{n=1}^{N} \sum\nolimits_t \sum\nolimits_{x,y} \|O_t^{(n)}(x, y) - \hat{O}_t^{(n)}(x, y)\|_2^2 \tag{8}$$

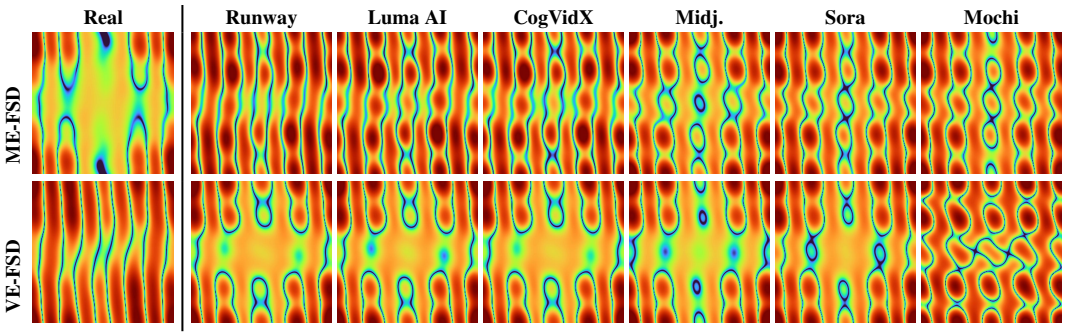


Figure 3: AR coefficient frequency response of the middle spatio-temporal slice for motion evolution FSDs (top) and visual evolution FSDs (bottom).

The resulting **motion residual** $r_t^O = O_t - \hat{O}_t$ measures how the actual motion evolution deviates from the predicted dynamics. The top row of Figure 3 confirms that these microstructures also differ between real and generated videos . We observe that real motion residuals exhibit broad, smoothly varying contours, whereas generators produce narrower, sharper structures with distinct patterns.

Importantly, both w_I^* and w_O^* are *learned once* from real videos and frozen. At test time, they are applied to any video to produce analyzable residuals.

3.3 Forensic Self-Descriptions via Autoregressive Modeling

The residuals r_t^I and r_t^O contain forensic microstructures that are latent within the high-dimensional residual signal. To obtain a compact representation, we fit a parametric model whose parameters serve as a description of the microstructure.

These microstructures are inherently three-dimensional. A residual value at location (x, y, t) is correlated with neighbors within the same frame and across adjacent frames. A purely spatial model such as 2D AR [22] would capture within-frame structure but discard cross-frame dependencies. Cross-frame dependencies carry significant discriminative power, as we show in our experiments.

To jointly capture spatial and temporal correlations, we model each residual as a 3D AR process. For each location (x, y, t) and channel c, the residual is a linear combination of channels at neighboring spatio-temporal locations:

$$r_t^I(x,y,c) = \sum\nolimits_{(i,j)\in\mathcal{A}_I} \sum\nolimits_{k=1}^{K_I} \sum\nolimits_{\ell=1}^{3} \phi(i,j,k,\ell,c)\; r_{t-k}^I(x-i, y-j, \ell) + \eta_t(x,y,c) \tag{9}$$

$$r_t^O(x,y,c) = \sum\nolimits_{(i,j)\in\mathcal{A}_O} \sum\nolimits_{k=1}^{K_O} \sum\nolimits_{\ell=1}^{2} \zeta(i,j,k,\ell,c)\; r_{t-k}^O(x-i, y-j, \ell) + \xi_t(x,y,c) \tag{10}$$

where η_t and ξ_t are innovation noise. The coefficient sets $\Phi = \{\phi(i,j,k,\ell,c)\}$ and $Z = \{\zeta(i,j,k,\ell,c)\}$ are estimated jointly across all channels using a sum-to-one constrained least-squares method [56]. These coefficients encode how the residual at each location depends on its spatio-temporal neighborhood. Because the underlying processes generating these dependencies differ between real and synthetic videos (Section 3.1), the resulting coefficients are discriminative.

Crucially, Φ and Z (referred to as VE-FSD and ME-FSD, respectively) are estimated per video at test time, producing a *spatio-temporal forensic self-description*: a compact parametric summary of the microstructure in its own residuals. This makes the method inherently generator-agnostic. Rather than learning what synthetic video looks like, we ask each video to *describe its own temporal microstructure* and determine whether that description is consistent with real video. A video from an unseen generator must still describe itself, and its description will differ from those of real videos.

4 Zero-Shot Detection and Source Attribution

We now describe how to leverage the self-descriptions Φ and Z for two tasks: zero-shot detection of AI-generated video, and source generator attribution.

4.1 Zero-Shot AI-Generated Video Detection

Our predictors are trained exclusively on real video (Section 3.1), so real video self-descriptions reflect a shared residual structure and naturally cluster in the self-description space. In contrast, synthetic videos follow generator-specific temporal patterns rather than real-video temporal evolution, causing their self-descriptions to lie outside this cluster. We exploit this separation by modeling the distribution of real self-descriptions and classifying outliers as AI-generated.

We model the distribution of real video self-descriptions using Gaussian Mixture Models [57]:

$$P(\Phi \mid \text{real}) = \sum_{m=1}^{M} \pi_m^I \mathcal{N}(\mu_m^I, \Sigma_m^I), \qquad P(Z \mid \text{real}) = \sum_{m=1}^{M} \pi_m^O \mathcal{N}(\mu_m^O, \Sigma_m^O) \tag{11}$$

where $\{\pi_m, \mu_m, \Sigma_m\}$ are mixture weights, means, and covariances estimated from real videos only. At test time, we compute the combined log-likelihood $\mathcal{L} = \log P(\Phi \mid \text{real}) + \log P(Z \mid \text{real})$ and classify the video as AI-generated if $\mathcal{L}$ falls below a threshold τ.

4.2 Source Attribution

We additionally aim to identify which generator produced a given synthetic video. Although this task requires labeled synthetic training data, the same self-descriptions can be reused directly.

Generators approximate h_I and h_O in different ways, yielding differences that are captured in the self-descriptions. We define the combined feature vector $\Omega = [\Phi \oplus Z]$ (referred to as VFSD) by concatenating the ℓ_2-normalized VE-FSD and ME-FSD, and train a classifier (SVM or KNN) on Ω with generator labels.

Visualization of Self-Description Space. Figure 4 visualizes the self-description space via t-SNE [58]. In the bottom row, real video self-descriptions form a distinct cluster well-separated from all generators, confirming the separation exploited by our zero-shot detector. Each generator forms its own cluster, explaining why the same self-descriptions support source attribution. The top row shows that existing forensic descriptors (NPR, FSD) and general-purpose representations (CLIP-ViT) fail to separate sources, which remain heavily entangled.

5 Experiments and Results

5.1 Implementation Details

We compute optical flow between consecutive frames using RAFT [55]. The VE-FSD predictor uses a spatial neighborhood of size B=11 and the ME-FSD predictor uses D=11 with a temporal history of L=5 flow fields. Both predictors are trained on 5000 videos from the MIT dataset [59]. For the 3D autoregressive self-descriptions, we use spatial neighborhoods $|\mathcal{A}_I|$=11 and $|\mathcal{A}_O|$=11 with temporal depths K_I=3 and K_O=3. For zero-shot detection, we fit Gaussian Mixture Models (GMMs) with M=10 components to the real video self-descriptions. For source attribution, we train a linear SVM with C=1.0 on the ℓ_2-normalized combined descriptor Ω. All of our experiments are conducted on a machine with four NVIDIA RTX A6000 GPUs. At test time, residual extraction takes $\sim$10 ms/frame and AR fitting $\sim$90 ms/frame, both scaling linearly with frame count.

5.2 Datasets

We evaluate on three real video datasets: Moments-In-Time (MIT) [59], Video-ACID [60], and GenVideo [10]. For synthetic data, we collected videos from 35 generative models spanning diffusion, autoregressive, and hybrid architectures, including Sora [1], Veo 3 [2], Kling [32], HunyuanVideo [34], CogVideoX [5], Runway Gen-3 [3], Luma [31], Midjourney [23], Pika [33], Mochi [35], and AnimateDiff [21]. Each real dataset contains 1000 videos for fitting the GMMs, and 1000 videos for testing. Additionally, MIT includes a separate set of 5000 videos to train the predictors, which are kept frozen for all evaluations across all datasets. Details and visualizations are provided in Appendix B.

5.3 Zero-Shot Detection Evaluation

Setup. We benchmark VFSD against state-of-the-art baselines. VFSD operates in a fully zero-shot setting, having never seen any synthetic video during training, whereas competing methods use their

Table 1: Zero-shot detection (AUC) across 35 generators per real dataset.

Method	GenV.	V.ACID	MIT	Avg.
Image-Based Detectors				
NPR [16]	0.839	0.548	0.730	0.706
RINE [37]	0.744	0.825	**0.879**	0.816
FSD [22]	<u>0.999</u>	0.755	0.737	0.830
Video-Based Detectors				
DeMamba [10]	0.451	0.568	0.534	0.527
D3 [49]	0.928	0.844	0.688	0.820
AIGVdet [47]	0.875	0.777	0.805	0.819
FTCN [8]	0.579	0.596	0.548	0.547
Vahdati [19]	0.868	0.770	0.827	0.822
Ours				
ME-FSD	0.998	0.932	0.750	0.893
VE-FSD	**1.000**	0.930	<u>0.831</u>	**0.920**
VFSD	**1.000**	**0.957**	0.791	<u>0.916</u>

Table 2: Per-generator zero-shot detection (AUC) on Video-ACID.

	Generative Source								
Method	Sora	Runway	Pika	Luma	CogVid.	Midj.	LTX	Veo 3	Avg. ± Std
Image-Based Detectors									
NPR [16]	0.480	0.563	0.713	0.548	0.478	0.600	0.979	0.447	0.601 ± 0.175
RINE [37]	0.663	0.860	0.695	0.823	0.751	0.687	0.997	0.787	0.783 ± 0.111
FSD [22]	0.875	0.635	0.600	0.758	0.613	0.765	0.912	0.552	0.714 ± 0.134
Video-Based Detectors									
DeMamba [10]	0.620	0.551	0.581	0.601	0.506	0.641	0.511	0.491	0.563 ± 0.057
D3 [49]	0.774	0.932	0.925	0.948	0.723	0.836	0.865	0.889	0.862 ± 0.080
AIGVdet [47]	0.637	0.863	0.849	0.794	0.730	0.874	0.817	0.835	0.800 ± 0.080
FTCN [8]	0.569	0.610	0.636	0.606	0.545	0.554	0.629	0.505	0.582 ± 0.046
Vahdati [19]	0.846	0.735	0.805	0.740	0.597	0.868	1.000	0.687	0.785 ± 0.123
Ours									
ME-FSD	0.988	0.963	0.850	0.964	0.970	0.979	0.999	0.873	0.948 ± 0.055
VE-FSD	0.994	0.984	0.876	0.951	0.977	0.943	**1.000**	0.878	0.950 ± 0.049
VFSD	**0.999**	**0.993**	**0.915**	**0.987**	**0.994**	**0.993**	**1.000**	**0.927**	**0.976 ± 0.034**

officially released pre-trained weights. Our evaluation encompasses 105 real-versus-synthetic pairs (3 real datasets $\times$ 35 generators). Full per-generator results are in the supplementary material.

Competing Methods. We compare against video detection methods (DeMamba [10], D3 [49], Vahdati [19], FTCN [8], AIGVdet [47]) and image detection methods (FSD [22], NPR [16], RINE [37]).

Metrics. We report the Area Under the ROC Curve (AUC). Accuracy is reported in Appendix D.

Results. Table 1 reports aggregated results across all generators. VFSD achieves the highest average AUC of 0.916 across all three datasets, while VE-FSD alone reaches 0.920, compared to 0.830 for the best baseline FSD. Image-based detectors (NPR, RINE, FSD) cannot capture the temporal forensic traces that distinguish real from generated video. Although RINE achieves the highest single-dataset AUC on MIT (0.879), it drops to 0.744 on GenVideo, highlighting the inconsistency of purely spatial approaches. Among baselines that utilize temporal information, DeMamba and FTCN achieve near-chance performance (0.527 and 0.547 AUC). This is because they learn visual artifacts and temporal inconsistencies specific to generators seen during training, resulting in low generalizability [19, 22]. D3 operates training-free but computes hand-crafted statistics of pretrained deep features rather than modeling prediction residuals, reaching 0.820 AUC. By using temporal forensic microstructures in both visual evolution and motion evolution residuals, VFSD achieves accurate zero-shot detection without any synthetic training data.

The average AUC of VE-FSD alone (0.920) slightly exceeds that of the combined VFSD (0.916) because ME-FSD lowers performance on MIT (0.791 vs. 0.831). However, integrating ME-FSD improves results on Video-ACID (0.957 vs. 0.930), where motion traces are more informative. Different generators leave stronger traces in different components, so combining both provides more reliable detection across the full range of generators.

Table 2 provides a per-generator breakdown on Video-ACID. VFSD achieves the highest mean AUC (0.976), maintaining $\geq$0.915 on every generator. The closest spatial baselines, FSD and Vahdati [19], both exhibit high variance. FSD fluctuates between 0.552 (Veo 3) and 0.912 (LTX Video), while Vahdati ranges from 0.597 to 1.000. Even FSD's best case (0.912) falls below our worst case (0.915). This consistent performance stems from a limitation shared by all generators. Real video exhibits complex, non-linear temporal variation that current generators fail to faithfully capture (Section 6), making the resulting self-descriptions discriminative regardless of the generative architecture.

5.4 Source Generator Attribution Evaluation

Setup. We evaluate whether our self-descriptions contain generator-specific fingerprints by testing source attribution. Since no standard benchmark exists, we extract embeddings from each competing method and apply the same classifiers (linear SVM and KNN with $K{=}5$) to all for a fair comparison. We use videos from 7 sources (6 generative models and real videos from MIT), with 1000 videos per source in an $80\%/20\%$ train/test split.

Competing Methods. We compare against vision embeddings (CLIP-ViT [61], DINOv2 [62], ResNet-50 [63]) and forensic embeddings (NPR [16], D3 [49], Vahdati [19], FSD [22]).

Metric. We report classification accuracy.

Table 3: Source attribution accuracy (KNN/SVM) across 7 sources.

Method	Video Source							Avg.
	Real (MIT)	Sora	Runway	Luma	CogVid.	Midj.	Mochi	
Vision Embeddings								
ResNet-50	0.52/0.67	0.36/0.54	0.62/0.66	0.54/0.60	0.61/0.68	0.27/0.65	0.48/0.63	0.49/0.63
CLIP-ViT	0.80/0.88	0.58/**0.88**	0.62/0.82	0.56/0.72	**0.80**/0.77	0.63/0.82	0.58/0.78	0.65/0.81
DINOv2	0.66/0.74	0.39/0.58	0.57/0.61	0.51/0.59	0.71/0.60	0.48/0.73	0.48/0.56	0.54/0.63
Forensic Embeddings								
NPR	0.74/0.91	0.35/0.78	0.39/0.89	0.29/0.77	0.31/0.84	0.36/0.74	0.27/0.89	0.39/0.83
D3	0.78/0.87	0.60/0.84	0.69/0.77	0.54/0.74	**0.80**/0.80	0.55/0.86	0.61/0.86	0.65/0.82
Vahdati	0.41/0.37	0.53/0.62	0.35/0.53	0.29/0.34	0.29/0.39	0.35/0.63	0.36/0.48	0.37/0.48
FSD	0.44/0.76	0.77/0.78	0.51/0.70	0.49/0.70	0.39/0.59	0.80/0.84	0.89/0.97	0.61/0.77
Ours								
ME-FSD	**0.99/0.99**	0.67/0.85	**0.79**/0.97	**0.94**/0.99	0.56/0.93	0.93/0.92	0.94/**0.98**	0.83/0.95
VE-FSD	0.98/0.99	**0.83**/0.85	0.53/0.88	0.80/0.98	0.46/0.86	0.90/0.93	0.94/0.94	0.78/0.92
VFSD	**0.99**/0.98	0.80/**0.88**	0.72/**0.97**	0.93/**0.99**	0.58/**0.96**	**0.94/0.96**	**0.98**/0.95	**0.85/0.96**

Results. As shown in Table 3, VFSD achieves the highest attribution accuracy (0.959 with SVM, 0.851 with KNN), substantially outperforming the best baselines of NPR+SVM (0.834) and CLIP-ViT+KNN / D3+KNN (0.654). Forensic embeddings consistently outperform general-purpose vision embeddings under SVM, confirming that forensic representations carry more source-discriminative information than semantic features.

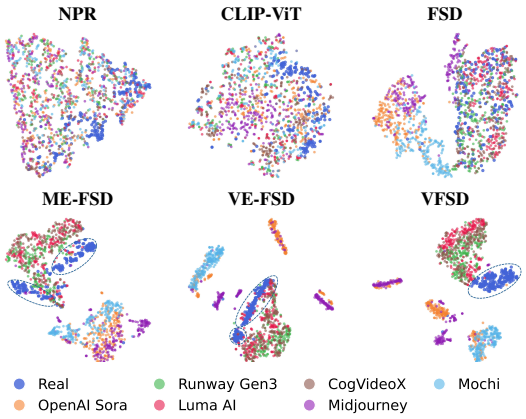


Figure 4: 2D t-SNE projections. Baseline embeddings (top) show heavy source entanglement. Our self-descriptions (bottom) separate each generator into distinct clusters and isolate real videos.

KNN performance reveals the intrinsic structure of each embedding space. All baselines degrade significantly under KNN (*e.g.*, NPR falls from 0.834 to 0.391, D3 from 0.824 to 0.654), indicating that their embeddings do not naturally separate sources. In contrast, VFSD achieves 0.851 under KNN, surpassing the best baseline's SVM accuracy (0.834), because each generator approximates h_I and h_O differently (Section 3.1) and these differences are directly encoded in the self-descriptions. This natural clustering carries practical advantages, as new generators can be incorporated by simply computing self-descriptions and fitting a lightweight classifier, without retraining any neural network. Supporting both zero-shot detection and source attribution confirms these temporal forensic microstructures are detection-discriminative and source-specific.

These findings are qualitatively confirmed by the t-SNE plots in Figure 4. Baseline embeddings exhibit heavy source entanglement, consistent with their poor KNN performance, whereas our self-descriptions map each source to a distinct, well-separated cluster. The combined VFSD projection (bottom right) yields the sharpest separation, cleanly isolating real videos from all synthetic sources.

6 Discussion

Why Temporal Residuals Are Discriminative. Because video generators enforce temporal consistency through 3D convolutions and cross-frame attention [4, 21, 20, 64], the resulting synthetic videos do not contain the complex, non-linear frame-to-frame variation of real scenes. A counterintuitive consequence is that our motion predictor f_O achieves *lower* prediction error on synthetic videos than on real videos. Figure 5 confirms that all generators exhibit lower mean motion residual energy than real videos.

However, total residual energy alone is insufficient for full discriminative power. The forensic microstructures manifest as complex 3D spatio-temporal dependencies captured by the AR coefficients (Section 3.3), which encode how each residual value relates to its neighbors rather than merely summarizing magnitude. For real video, these dependencies reflect the model-mismatch process ϵ_t from Section 3.1; for generated video, they reflect the generator's temporal conditioning patterns. This difference is visible in Figure 3, where real videos produce broad, smoothly varying AR frequency responses, while each generator introduces distinct higher-frequency patterns.

VE-FSD and ME-FSD Encode Different Signals. To verify that the two components capture different forensic signals, we evaluate VE-FSD and ME-FSD independently on 9,000 videos (1,000 real, 8,000 synthetic across eight generators). This results in VE-FSD producing 228 errors and ME-FSD 437 errors. Detection errors made by the two components are nearly disjoint. Among the videos that VE-FSD misclassifies, ME-FSD correctly classifies 93.9%. Among those that ME-FSD misclassifies, VE-FSD correctly classifies 96.8%, yielding a Phi coefficient of -0.96. We can observe this difference in Figure 4 (bottom row), where each component separates a different subset of generators. The combined VFSD projection resolves both with the sharpest separation.

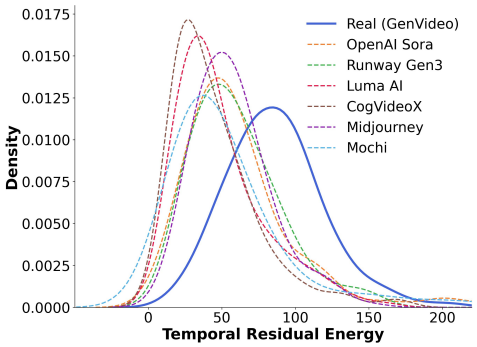


Figure 5: Motion residual energy distributions for real vs AI-generated videos.

We further quantify the practical impact of this difference by repeating the same experiment measuring the detection performance of two strategies: joint distribution (a single GMM on the concatenated descriptor) vs. product of marginals (separate GMMs on Φ and Z combined by summing log-likelihoods). We found that the product of marginals outperforms the joint distribution (AUC 0.992 vs. 0.970, balanced accuracy 96.8% vs. 92.9%). This is likely because concatenation doubles the dimensionality without adding samples, causing overfitting, while the marginal approach loses nothing given the near-independence shown above.

Robustness to Video Compression. We evaluate the robustness of our forensic self-descriptions to lossy video compression. To do this, we re-encode all test videos using H.264 [65] at Constant Rate Factors of $\{12, 23, 32\}$. As shown in Table 4, VFSD drops from 0.991 to only 0.978 AUC under heavy compression. VE-FSD is more stable (0.980 AUC at CRF 32) than ME-FSD (0.953 to 0.929), because heavy compression smooths the motion fields ME-FSD relies on. The combined VFSD remains above 0.978 across all conditions, confirming that temporal forensic microstructures persist even after lossy compression.

Table 4: Detection AUC vs. compression

		CRF		
Method	None	12 (L)	23 (M)	32 (H)
VE-FSD	0.979	0.989	0.991	0.980
ME-FSD	0.953	0.952	0.953	0.929
VFSD	0.991	0.991	0.991	0.978

Distinction from Spatial FSD. We verify that VFSD captures a different forensic signal than spatial FSD [22] by computing the Phi correlation between their detection errors. The correlations are strongly negative (-0.76 for ME-FSD, -0.92 for VE-FSD, -0.82 for VFSD), confirming that temporal forensic traces from how frames evolve over time are fundamentally distinct from the spatial microstructures that FSD extracts from individual frames.

Limitations and Future Work. Our method has several limitations. First, it depends on optical flow quality. Under severe motion blur or low light, flow estimation degrades, weakening ME-FSD. Second, if a generator produces erratic camera motion, the resulting residuals may resemble those of real videos, risking false negatives. Third, the linear predictors capture only first-order temporal relationships. While sufficient for current generators, future generators that better approximate real-world dynamics may require higher-order or nonlinear predictors to maintain discriminative residuals. These limitations are partially mitigated by our dual-component design.

Several future directions follow from these limitations. Incorporating a confidence-aware flow estimator that down-weights unreliable flow regions could improve ME-FSD robustness in challenging conditions. Our analysis above (Distinction from Spatial FSD) shows that temporal and spatial forensic signals are largely independent, suggesting that combining VFSD with spatial FSD [22] could yield a more powerful detector that captures both classes of forensic evidence. Finally,

extending the framework to model longer-range temporal dependencies is a promising direction for future investigation.

7 Ablation Study

We conduct comprehensive ablations on the MIT dataset to validate our design choices. Table 5 summarizes the results. We perform ablations on a subset of 5 generators: CogVideoX-1.5, IPOW, Luma AI, Midjourney, and Mochi-1. All experiments use the same baseline configuration unless otherwise specified, with the defaults highlighted.

Table 5: Ablation on MIT. Highlighted cells indicate chosen hyperparameters.

Component	Config.	AUC
Neighbor. Size ($\mathcal{A}$)	7×7	0.961
	9×9	0.989
	11×11	**0.992**
Temporal Depth (K)	$K = 1$	0.983
	$K = 3$	**0.992**
	$K = 5$	0.967
Detector	OC-SVM	0.991
	Isol. Forest	0.977
	GMM	**0.992**
Video Clip Length (T)	5 frames	0.990
	7 frames	**0.992**
	9 frames	0.982

Spatial Neighborhood Size ($\mathcal{A}$). We evaluate window sizes of 7×7, 9×9, and 11×11 pixels. Larger windows provide richer spatial context, leading to improved discrimination. The 11×11 window achieves 0.992 AUC compared to 0.961 for 7×7. This suggests that temporal microstructures manifest themselves in sufficiently large spatial neighborhoods.

Temporal Depth (K). We test temporal depths $K \in \{1, 3, 5\}$ frames. The optimal configuration uses $K = 3$ frames (0.992 AUC), balancing temporal consistency modeling with computational efficiency. Surprisingly, $K = 5$ degrades performance to 0.967, likely due to the introduction of redundant parameters.

Anomaly Detector. We compare three anomaly detectors: GMM, One-Class SVM (OC-SVM), and Isolation Forest. GMM achieves the best performance (0.992 AUC), demonstrating that explicit density modeling is well-suited for this task. OC-SVM achieves comparable performance (0.991). Isolation Forest underperforms (0.977), suggesting that tree-based methods may not capture the subtleties of residual distributions as effectively.

Video Clip Length (T). We evaluate the effect of varying the input clip length over $T \in \{5, 7, 9\}$ consecutive frames used to construct the residual volume. The 7-frame setting optimally balances temporal context and computational efficiency (0.992 AUC). While a shorter 5-frame clip retains strong performance (0.990), extending the clip to 9 frames degrades performance to 0.982, as excessively long sequences accumulate scene motion variations and dilute localized temporal microstructures.

8 Conclusion

We proposed VFSD, a zero-shot framework that detects AI-generated videos by exposing temporally distributed forensic microstructures. By modeling real video as a dynamic system and measuring how prediction residuals deviate from learned temporal evolution patterns, our method captures forensic traces without ever training on synthetic data. Experiments across 35 generators demonstrate state-of-the-art zero-shot detection performance, accurate source attribution, and robustness to video compression.

References

[1] Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, Clarence Ng, Ricky Wang, and Aditya Ramesh. Video generation models as world simulators. `https://openai.com/index/video-generation-models-as-world-simulators/`, 2024. OpenAI Technical Report.

[2] Google DeepMind. Veo 3 technical report. Technical report, Google DeepMind, 2025.

[3] Runway Research. Introducing Gen-3 Alpha. `https://runwayml.com/research/introducing-gen-3-alpha`, 2024.

[4] Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your latents: High-resolution video synthesis with latent diffusion models. In *CVPR*, pages 22563–22575, 2023.

[5] Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, Da Yin, Yuxuan Zhang, Weihan Wang, Yean Cheng, Bin Xu, Xiaotao Gu, Yuxiao Dong, and Jie Tang. CogVideoX: Text-to-video diffusion models with an expert transformer. In *ICLR*, 2025.

[6] Europol Innovation Lab. Facing reality? Law enforcement and the challenge of deepfakes. Technical report, Europol, 2022.

[7] Luisa Verdoliva. Media forensics and deepfakes: an overview. *IEEE Journal of Selected Topics in Signal Processing*, 14(5):910–932, 2020.

[8] Yinglin Zheng, Jianmin Bao, Dong Chen, Ming Zeng, and Fang Wen. Exploring temporal coherence for more general video face forgery detection. In *ICCV*, pages 15044–15054, 2021.

[9] Zhihao Gu, Yang Chen, Taiping Yao, Shouhong Ding, Jilin Li, Feiyue Huang, and Lizhuang Ma. Spatiotemporal inconsistency learning for deepfake video detection. In *ACM Intl. Conf. Multimedia*, pages 3473–3481, 2021.

[10] Haoxing Chen, Yan Hong, Zizheng Huang, Zhuoer Xu, Zhangxuan Gu, Yaohui Li, Jun Lan, Huijia Zhu, Jianfu Zhang, Weiqiang Wang, and Huaxiong Li. DeMamba: AI-generated video detection on million-scale GenVideo benchmark. *arXiv preprint arXiv:2405.19707*, 2024.

[11] Riccardo Corvi, Davide Cozzolino, Giada Zingarini, Giovanni Poggi, Koki Nagano, and Luisa Verdoliva. On the detection of synthetic images generated by diffusion models. In *ICASSP*, pages 1–5, 2023.

[12] Diego Gragnaniello, Davide Cozzolino, Francesco Marra, Giovanni Poggi, and Luisa Verdoliva. Are GAN generated images easy to detect? A critical analysis of the state-of-the-art. In *IEEE Intl. Conf. Multimedia and Expo*, pages 1–6, 2021.

[13] Diangarti Tariang, Riccardo Corvi, Davide Cozzolino, Giovanni Poggi, Koki Nagano, and Luisa Verdoliva. Synthetic image verification in the era of generative artificial intelligence: What works and what isn't there yet. *IEEE Security & Privacy*, 22(3):37–49, 2024.

[14] Li Lin, Neeraj Gupta, Yue Zhang, Hainan Ren, Chun-Hao Liu, Feng Ding, Xin Wang, Xin Li, Luisa Verdoliva, and Shu Hu. Detecting multimedia generated by large AI models: A survey. *arXiv preprint arXiv:2402.00045*, 2024.

[15] Sheng-Yu Wang, Oliver Wang, Richard Zhang, Andrew Owens, and Alexei A. Efros. CNN-generated images are surprisingly easy to spot... for now. In *CVPR*, pages 8695–8704, 2020.

[16] Chuangchuang Tan, Yao Zhao, Shikui Wei, Guanghua Gu, Ping Liu, and Yunchao Wei. Rethinking the up-sampling operations in CNN-based generative network for generalizable deepfake detection. In *CVPR*, pages 28130–28139, 2024.

[17] Utkarsh Ojha, Yuheng Li, and Yong Jae Lee. Towards universal fake image detectors that generalize across generative models. In *CVPR*, pages 24480–24489, 2023.

[18] Chuangchuang Tan, Yao Zhao, Shikui Wei, Guanghua Gu, and Yunchao Wei. Learning on gradients: Generalized artifacts representation for GAN-generated images detection. In *CVPR*, pages 12105–12114, 2023.

[19] Danial Samadi Vahdati, Tai D. Nguyen, Aref Azizpour, and Matthew C. Stamm. Beyond deepfake images: Detecting AI-generated videos. In *CVPRW*, pages 4397–4408, 2024.

[20] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David Fleet. Video diffusion models. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 8633–8646. Curran Associates, Inc., 2022. doi: 10.52202/068431-0628. URL `https://proceedings.neurips.cc/paper_files/paper/2022/file/39235c56aef13fb05a6adc95eb9d8d66-Paper-Conference.pdf`.

[21] Yuwei Guo, Ceyuan Yang, Anyi Rao, Zhengyang Liang, Yaohui Wang, Yu Qiao, Maneesh Agrawala, Dahua Lin, and Bo Dai. AnimateDiff: Animate your personalized text-to-image diffusion models without specific tuning. In *ICLR*, 2024.

[22] Tai D. Nguyen, Aref Azizpour, and Matthew C. Stamm. Forensic self-descriptions are all you need for zero-shot detection, open-set source attribution, and clustering of AI-generated images. In *CVPR*, pages 3040–3050, 2025.

[23] Midjourney. Midjourney. `https://www.midjourney.com`. Accessed: 2025.

[24] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

[25] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of StyleGAN. In *CVPR*, pages 8110–8119, 2020.

[26] Diederik P. Kingma and Max Welling. Auto-encoding variational Bayes. In *ICLR*, 2014.

[27] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, pages 6840–6851, 2020.

[28] Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P. Kingma, Ben Poole, Mohammad Norouzi, David J. Fleet, and Tim Salimans. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022.

[29] Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, Devi Parikh, Sonal Gupta, and Yaniv Taigman. Make-a-video: Text-to-video generation without text-video data. In *ICLR*, 2023.

[30] Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, Varun Jampani, and Robin Rombach. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023.

[31] Luma Labs. Dream machine. `https://lumalabs.ai/dream-machine`, 2024.

[32] Kuaishou Technology. Kling. `https://kling.kuaishou.com`, 2024.

[33] Pika Labs. Pika. `https://pika.art`, 2024.

[34] Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, Jianwei Zhang, et al. HunyuanVideo: A systematic framework for large video generative models. *arXiv preprint arXiv:2412.03603*, 2024.

[35] Genmo. Mochi 1. `https://www.genmo.ai`, 2024.

[36] Davide Cozzolino, Giovanni Poggi, Riccardo Corvi, Matthias Nießner, and Luisa Verdoliva. Raising the bar of AI-generated image detection with CLIP. In *CVPRW*, pages 4356–4366, 2024.

[37] Christos Koutlis and Symeon Papadopoulos. Leveraging representations from intermediate encoder-blocks for synthetic image detection. In *ECCV*, pages 394–411, 2024.

[38] Jonas Ricker, Denis Lukovnikov, and Asja Fischer. AEROBLADE: Training-free detection of latent diffusion images using autoencoder reconstruction error. In *CVPR*, pages 9130–9140, 2024.

[39] Ricard Durall, Margret Keuper, and Janis Keuper. Watch your up-convolution: CNN based generative deep neural networks are failing to reproduce spectral distributions. In *CVPR*, pages 7890–7899, 2020.

[40] Quentin Bammey. Synthbuster: Towards detection of diffusion model generated images. *IEEE Open Journal of Signal Processing*, 5:1–9, 2024.

[41] Sergey Sinitsa and Ohad Fried. Deep image fingerprint: Towards low budget synthetic image detection and model lineage analysis. In *WACV*, pages 4067–4076, 2024.

[42] Davide Cozzolino, Giovanni Poggi, Matthias Nießner, and Luisa Verdoliva. Zero-shot detection of AI-generated images. In *ECCV*, pages 54–72, 2024.

[43] Andreas Rossler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Nießner. FaceForensics++: Learning to detect manipulated facial images. In *ICCV*, pages 1–11, 2019.

[44] 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.

[45] Irene Amerini, Leonardo Galteri, Roberto Caldelli, and Alberto Del Bimbo. Deepfake video detection through optical flow based CNN. In *ICCV Workshops*, 2019.

[46] Umur Aybars Ciftci, Ilke Demir, and Lijun Yin. Fakecatcher: Detection of synthetic portrait videos using biological signals. *IEEE transactions on pattern analysis and machine intelligence*, 2020.

[47] Jianfa Bai, Man Lin, Gang Cao, and Zijie Lou. Ai-generated video detection via spatial-temporal anomaly learning. In *Chinese Conference on Pattern Recognition and Computer Vision (PRCV)*, pages 460–470. Springer, 2024.

[48] Rohit Kundu, Hao Xiong, Vishal Mohanty, Athula Balachandran, and Amit K. Roy-Chowdhury. Towards a universal synthetic video detector: From face or background manipulations to fully AI-generated content. In *CVPR*, pages 28050–28060, 2025.

[49] Chende Zheng, Ruiqi Suo, Chenhao Lin, Zhengyu Zhao, Le Yang, Shuai Liu, Minghui Yang, Cong Wang, and Chao Shen. D3: Training-free AI-generated video detection using second-order features. In *ICCV*, pages 12852–12862, 2025.

[50] Matthew C. Stamm, Min Wu, and K. J. Ray Liu. Information forensics: An overview of the first decade. *IEEE Access*, 1:167–200, 2013.

[51] Owen Mayer and Matthew C. Stamm. Forensic similarity for digital images. *IEEE Trans. Inform. Forensics Security*, 15:1331–1346, 2020.

[52] Davide Cozzolino and Luisa Verdoliva. Noiseprint: A CNN-based camera model fingerprint. *IEEE Trans. Inform. Forensics Security*, 15:144–159, 2020.

[53] Belhassen Bayar and Matthew C. Stamm. Constrained convolutional neural networks: A new approach towards general purpose image manipulation detection. *IEEE Trans. Inform. Forensics Security*, 13(11):2691–2706, 2018.

[54] Francesco Marra, Diego Gragnaniello, Luisa Verdoliva, and Giovanni Poggi. Do GANs leave artificial fingerprints? In *IEEE Conf. Multimedia Inform. Processing and Retrieval*, pages 506–511, 2019.

[55] Zachary Teed and Jia Deng. RAFT: Recurrent all-pairs field transforms for optical flow. In *ECCV*, pages 402–419, 2020.

[56] Simon S. Haykin. *Adaptive Filter Theory*. Prentice Hall, 4th edition, 2002.

[57] Geoffrey McLachlan and Kaye Basford. *Mixture Models: Inference and Applications to Clustering*. Marcel Dekker, 1988.

[58] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

[59] Mathew Monfort, Alex Andonian, Bolei Zhou, Kandan Ramakrishnan, Sarah Adel Bargal, Tom Yan, Lisa Brown, Quanfu Fan, Dan Gutfreund, Carl Vondrick, and Aude Oliva. Moments in time dataset: One million videos for event understanding. *IEEE TPAMI*, 42(2):502–508, 2020.

[60] Brian C. Hosler, Xinwei Zhao, Owen Mayer, Chen Chen, James A. Shackleford, and Matthew C. Stamm. The video authentication and camera identification database: A new database for video forensics. *IEEE Access*, 7:76937–76948, 2019.

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

[62] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jégou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024.

[63] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[64] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, pages 10684–10695, 2022.

[65] Iain E. Richardson. *The H.264 Advanced Video Compression Standard*. Wiley, 2nd edition, 2010.

[66] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

[67] Changli Tang, Yixuan Li, Yudong Yang, Jimin Zhuang, Guangzhi Sun, Wei Li, Zejun Ma, and Chao Zhang. video-SALMONN 2: Caption-enhanced audio-visual large language models. *arXiv preprint arXiv:2506.15220*, 2025.

[68] Yihan Wang, Lahav Lipson, and Jia Deng. SEA-RAFT: Simple, efficient, accurate raft for optical flow. In *Computer Vision – ECCV 2024: 18th European Conference, Milan, Italy, September 29–October 4, 2024, Proceedings, Part VII*, pages 36–54, Berlin, Heidelberg, 2024. Springer-Verlag. ISBN 978-3-031-72666-8. doi: 10.1007/978-3-031-72667-5_3. URL https://doi.org/10.1007/978-3-031-72667-5_3.

[69] Yihan Wang and Jia Deng. WAFT: Warping-alone field transforms for optical flow. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=HTqGE0KcuF.

[70] Lukas Ruff, Robert Vandermeulen, Nico Goernitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Alexander Binder, Emmanuel Müller, and Marius Kloft. Deep one-class classification. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4393–4402. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/ruff18a.html.

[71] Zhikang Liu, Yiming Zhou, Yuansheng Xu, and Zilei Wang. SimpleNet: A simple network for image anomaly detection and localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20402–20411, 2023.

Video Forensic Self-Descriptions: Leveraging Temporally Distributed Forensic Microstructures for Zero-Shot Detection of AI-Generated Videos

Supplementary Material

Page	Appendix	Title
1	A	Implementation Details
2	B	Dataset Details
2	C	Complete Zero-Shot Detection Results
4	D	Threshold Sensitivity and Cross-Dataset Transferability
7	E	Optical Flow Estimator Invariance
7	F	Anomaly Detection Architecture Ablation
7	G	Predictor Architecture Ablation
8	H	Additional Robustness Experiments
8	I	Time Complexity Analysis
8	J	GenVideo Results
9	K	AR Frequency Response Computation
9	L	Broader Societal Impacts

A Implementation Details

Predictor Architectures. Both f_I and f_O are implemented as single convolutional layers with an 11×11 kernel and no activation function (*i.e.*, purely linear filters). The Visual Evolution Predictor f_I takes the 5-channel concatenation $X_{t-1} = [I_{t-1} \oplus O_{t-1}] \in \mathbb{R}^{H \times W \times 5}$ — three RGB channels and two optical flow channels — and outputs the predicted frame $\hat{I}_t \in \mathbb{R}^{H \times W \times 3}$. The Motion Evolution Predictor f_O takes the $2L$-channel concatenation $X_t^O = [O_{t-1} \oplus \cdots \oplus O_{t-L}] \in \mathbb{R}^{H \times W \times 10}$ with $L = 5$ past flow fields and outputs the predicted flow field $\hat{O}_t \in \mathbb{R}^{H \times W \times 2}$. Optical flow is estimated from consecutive frame pairs using RAFT [55] (large model). Frames with any dimension exceeding 1080 pixels are downsampled to fit within that limit while preserving aspect ratio, with both dimensions rounded to multiples of 8 as required by RAFT.

Predictor Training. Both f_I and f_O are trained on 5,000 videos from the MIT Moments in Time dataset [59], with frames center-cropped to 512×512. Both predictors are optimized with the MSE loss between their predictions and ground-truth targets (Eqs. (6) and (8) of the main paper). Per training epoch, the VE-FSD predictor samples 5 consecutive frame pairs per video, and the ME-FSD predictor samples 3 sequences of $L + 2 = 7$ consecutive frames per video, each yielding $L + 1 = 6$ consecutive optical flows from which one prediction residual is computed. Both use AdamW [66] with learning rate 10^{-4}, weight decay 10^{-4}, and gradient clipping with maximum norm 1.0. A learning rate scheduler reduces the learning rate by a factor of 0.75 when validation loss plateaus. Training runs for up to 50 epochs with early stopping (patience of 10 epochs). Once trained, both predictors are frozen for all subsequent experiments.

3D AR Self-Description Fitting. For each video at test time, the first $T = 7$ frames are extracted. The frozen VE predictor is applied to each consecutive pair (I_{t-1}, O_{t-1}) to produce the spatio-temporal residual sequence r^I, and the frozen ME predictor is applied to each consecutive sequence of L flows to produce the motion residual sequence r^O. All residual frames are resized to 512 on the smaller edge while maintaining aspect ratio and then center-cropped to 512×512.

The AR coefficients Φ are estimated by fitting Eq. (9) to the volume r^I. All overlapping spatio-temporal patches of size $K_I \times |\mathcal{A}_I| \times |\mathcal{A}_I| = 3 \times 11 \times 11$ are extracted from the residual volume with unit stride in all three dimensions. Within each patch, the element at temporal index $\lfloor K_I/2 \rfloor = 1$ and spatial index $(5, 5)$ serves as the regression target; the remaining $K_I \cdot |\mathcal{A}_I|^2 - 1 = 362$ elements serve as predictors. The coefficient set Φ is estimated from all patch observations jointly using



Figure S1: Video frames in our dataset from real (top) and the synthetic sources (bottom).

a least-squares method [56]. The motion coefficients Z are estimated identically from r^O using $|\mathcal{A}_O| = 11$ and $K_O = 3$.

Zero-Shot Detection. For each of three real video datasets — MIT Moments in Time [59], GenVideo [10], and Video-ACID [60] — two GMMs with $M = 10$ full-covariance components are fitted to the self-descriptions extracted from 1,000 real training videos, one for $P(\Phi \mid \text{real})$ and one for $P(Z \mid \text{real})$, following Eq. (11). At test time, the combined log-likelihood $\mathcal{L} = \log P(\Phi \mid \text{real}) + \log P(Z \mid \text{real})$ is computed per video and thresholded at τ for binary prediction, with AUC reported over all values of τ. Each detector is evaluated on 1,000 held-out real validation videos (disjoint from the 1,000 used to fit the GMMs) and 1,000 AI-generated videos from each of 35 generators, processed independently for each real dataset.

B Dataset Details

B.1 Synthetic Data Creation

To ensure a rigorous and unbiased evaluation of our zero-shot detection framework, we constructed the synthetic videos to closely mirror the semantic distribution of our real benchmarks. We began by randomly sampling reference videos from the Video-ACID and MIT datasets. To extract high-quality text descriptions of the real videos, we processed each video using the Video-SALMONN 2 vision-language model [67] to generate descriptive captions. These generated captions were then utilized directly to prompt the generative models detailed in Table S1.

This protocol minimizes the influence of semantic domain shift between the real and synthetic evaluation sets and ensures that the performance of VFSD is driven by the detection of temporally distributed forensic microstructures, rather than relying on visual content.

For all generative models, we utilized their standard inference configurations, avoiding generator-specific prompt engineering or negative prompting that might artificially suppress forensic traces. Data visualizations are provided in Figure S1.

Synthetic Video Datasets. We evaluate on 1,000 videos from each of 35 AI video generators, listed in Table S1. Importantly, for zero-shot detection experiments, AI-generated videos are solely used for evaluation purposes and are not used for training the predictors or to fit the GMMs. For source attribution experiments, the videos from each generator are split into 80%/20% train/test splits.

Real Video Datasets. Table S2 summarizes the three real video datasets used for predictor training, GMM fitting, and evaluation. For MIT [59], we use 5,000 videos to train the VE-FSD and ME-FSD predictors (Section 3.2). For each dataset, a disjoint set of 1,000 videos is used to fit the GMMs $P(\Phi \mid \text{real})$ and $P(Z \mid \text{real})$, and a separate 1,000 videos serve as the held-out real test set for performance evaluation.

C Complete Zero-Shot Detection Results

Per-generator AUC results for all 35 AI video generators evaluated against each of the three real-video test sets are reported in Table S3–S5.

Generator-specific difficulty. Table S3 shows that methods fail on different generators: no single generator is hard for all approaches. On MIT, our temporal methods (ME-FSD, VE-FSD) score substantially below their averages on Kling (VE-FSD: 0.40, ME-FSD: 0.31), Wan 2.1 (VE-FSD: 0.58, ME-FSD: 0.33), Wan 2.2 (VE-FSD: 0.39, ME-FSD: 0.22), and Vidu Q1 (VE-FSD: 0.50) —

Table S1: Synthetic video datasets used for evaluation. Each generator contributes 1,000 test videos. Resolution is reported as width × height. Ranges indicate variation across videos within a generator family.

Generator	Resolution	# Frames	Test Samples
AnimateDiff v1	512×512	16	1,000
AnimateDiff v2	512×512	16	1,000
CausVid	640×352	128	1,000
CogVideoX 1.5	1360×768	161	1,000
EasyAnimate v5.1	672×384	49	1,000
HunyuanVideo	1280×720	129	1,000
IPOC	1360×768	81	1,000
IPOW	832×480	81	1,000
JT-CV-9B	2158×1214	51	1,000
Kling 1.6	1280×720	241	1,000
LanDiff	720×480	49	1,000
LTX-Video v0.9.1	768×512	50–121	1,000
Luma	1360×752	121	1,000
Magi T2V	1280×720	96	1,000
Meta AI	768–1280×768–1280	64	1,000
Midjourney	448–1344×464–1360	125	1,000
MiniMax Video-01	1280×720	50–141	1,000
MiracleVision v5	720×480	48–121	1,000
Mochi 1	848×480	60–163	1,000
Open-Sora 2.0	1024×576	48–129	1,000
Pika 1.0	1280×720	72	1,000
Pika 1.5	1296×720	120	1,000
RepVideo	720×480	49	1,000
Runway Gen-3	1280×768	256	1,000
Sora	480–854×480–1080	132–600	1,000
Stable Video Diffusion 1.0	1024×576	25	1,000
StepVideo	992×544	204	1,000
STIV (Apple)	512×512	20–77	1,000
Vchitect 2.0	768×432	40	1,000
Veo 3	1280×720	192	1,000
VideoCrafter 1	512–1024×320–576	16	1,000
VideoCrafter 2	512×320	16	1,000
Vidu Q1	1280×720	125	1,000
Wan 2.1	1280×720	81	1,000
Wan 2.2	1280×720	81	1,000
Total			**35,000**

generators where RINE achieves 0.97, 0.90, 0.91, and 0.90, respectively. Conversely, on SVD, STIV, and Midjourney — where RINE scores 0.65, 0.71, and 0.76 — VE-FSD achieves 0.99, 1.00, and 0.99. These differences indicate that some generators induce forensic microstructures that are more separable from real video in spatial representations, while others are more separable in temporal statistics, but not necessarily in both.

Uniform performance of supervised detectors. DeMamba and FTCN show a distinct pattern across all three tables: their per-generator AUC falls within a narrow range of 0.47–0.62 (DeMamba, MIT) and 0.48–0.63 (FTCN, MIT), with per-dataset standard deviations of at most 0.04 in all tables. For comparison, RINE spans 0.65–0.98 across the same generators on MIT (Table S3). DeMamba and FTCN do not selectively succeed on some generators and fail on others; their scores are nearly the same for every generator. This shows that these detectors do not capture general forensic microstructures that would allow them to distinguish AI-generated videos from real videos in a zero-shot setting.

Table S2: Real video datasets used for predictor training, GMM fitting, and evaluation. "Predictor Train" videos are used to train the VE-FSD and ME-FSD predictors (MIT only). "GMM Train" and "Test" are disjoint splits used for fitting the GMMs and evaluating detection AUC, respectively. Resolution is reported as width × height; ranges reflect variation across videos.

Dataset	Resolution	# Frames	Predictor Train	GMM Train	Test
MIT Moments in Time [59]	480–3840 × 480–2160	15–360	5,000	1,000	1,000
GenVideo [10]	298 × 224	30–92	—	1,000	1,000
Video-ACID [60]	640–3840 × 480–2160	46–627	—	1,000	1,000

Table S3: Per-generator zero-shot detection (AUC) on MIT Moments in Time. Numbers are averaged over variants within the same generator family.

		Generative Source																														
Method	**Avg ± Std**	**ADiff-v1**	**ADiff-v2**	**CausVid**	**CogVidX**	**EasyAnim**	**Hunyuan**	**IPOC**	**IPOW**	**JT-CV-9B**	**Kling**	**LanDiff**	**LTX-Vid**	**Luma**	**MagiT2V**	**Meta A1**	**Midj.**	**MiniMax**	**Miracle**	**Mochi**	**O-Sora**	**Pika 1.0**	**Pika 1.5**	**RepVid**	**Runway**	**Sora**	**SVD**	**StepVid**	**STIV**	**Vchit.**	**Veo 3**	**VC1**	**VC2**	**Vidu Q1**	**Wan 2.1**	**Wan 2.2**
Image-Based Detectors																																			
NPR [16]	0.72 ± 0.18	0.49	0.53	0.95	0.70	0.89	0.69	0.91	0.80	0.90	0.72	0.68	0.68	0.73	0.65	0.96	0.80	0.80	0.72	0.63	0.67	0.73	0.88	0.80	0.77	0.71	0.26	0.79	0.12	0.48	0.71	0.78	0.56	0.76	0.87	**0.94**
RINE [37]	**0.88** ± 0.08	0.96	0.96	0.97	0.82	0.82	**0.81**	0.90	0.96	0.98	**0.97**	0.85	0.96	0.88	**0.80**	0.97	0.76	0.88	0.86	0.84	0.87	0.92	0.76	0.87	0.91	0.82	0.65	0.88	0.71	0.93	0.84	0.96	0.98	**0.90**	**0.90**	0.91
FSD [22]	0.78 ± 0.23	0.93	0.93	1.00	0.49	1.00	0.46	0.57	0.94	0.99	0.65	0.96	0.95	0.71	0.54	1.00	0.84	0.45	0.97	0.98	0.92	0.65	0.47	0.94	0.60	0.95	0.94	0.98	0.96	0.99	0.38	0.95	1.00	0.37	0.41	0.42
Video-Based Detectors																																			
DeMamba [10]	0.54 ± 0.04	0.55	0.59	0.60	0.48	0.56	0.51	0.58	0.52	0.59	0.53	0.57	0.49	0.58	0.50	0.51	0.62	0.56	0.56	0.50	0.54	0.59	0.55	0.57	0.50	0.55	0.48	0.47	0.48	0.53	0.48	0.54	0.56	0.51	0.51	0.56
D3 [49]	0.69 ± 0.14	0.38	0.47	0.83	0.53	0.75	0.68	0.72	0.85	0.70	0.76	0.77	0.69	0.82	0.77	0.84	0.63	0.82	0.76	0.78	0.88	0.83	0.76	0.69	0.77	0.61	0.31	0.74	0.38	0.50	0.77	0.54	0.69	0.58	0.74	0.65
AIGVDet [47]	0.80 ± 0.12	0.86	0.94	0.63	0.76	0.89	0.80	0.90	0.86	0.92	0.79	0.74	0.77	0.83	0.79	0.76	0.90	**0.90**	0.46	0.63	0.76	**0.93**	**0.88**	0.67	0.89	0.70	0.96	0.45	0.80	0.73	**0.85**	0.72	0.86	0.74	0.89	0.88
FTCN [8]	0.55 ± 0.03	0.61	0.56	0.54	0.50	0.52	0.51	0.55	0.59	0.58	0.54	0.57	0.57	0.56	0.50	0.51	0.50	0.56	0.63	0.55	0.52	0.56	0.58	0.58	0.56	0.50	0.59	0.57	0.57	0.58	0.48	0.58	0.56	0.54	0.55	0.55
Vahdati [19]	0.82 ± 0.12	0.86	0.92	0.72	0.66	0.54	0.71	0.82	0.92	0.94	0.91	0.83	0.93	0.81	0.64	0.99	0.91	0.81	0.91	0.69	0.80	0.78	0.87	0.81	0.81	0.91	0.99	0.79	0.94	0.60	0.81	0.87	0.60	0.74	0.88	0.92
Ours																																			
ME-FSD	0.75 ± 0.27	1.00	1.00	0.99	0.95	0.98	0.45	0.91	0.97	**1.00**	0.31	0.98	1.00	0.89	0.57	1.00	0.95	0.36	0.96	0.99	0.98	0.53	0.49	0.97	0.93	0.96	0.97	0.97	1.00	0.97	0.38	0.98	0.99	0.31	0.33	0.22
VE-FSD	0.83 ± 0.20	**1.00**	**1.00**	0.98	0.98	0.99	0.58	**0.98**	1.00	0.99	0.40	1.00	**1.00**	0.95	0.65	0.99	0.99	0.64	1.00	1.00	0.99	0.64	0.67	1.00	0.98	0.99	0.99	1.00	**1.00**	0.99	0.63	0.97	0.95	0.50	0.58	0.39
VFSD	0.79 ± 0.26	**1.00**	**1.00**	**1.00**	**0.98**	**1.00**	0.48	0.98	**1.00**	**1.00**	0.38	**1.00**	**1.00**	**0.97**	0.64	**1.00**	**1.00**	0.46	**1.00**	**1.00**	**1.00**	0.65	0.56	**1.00**	**0.98**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.43	**1.00**	**1.00**	0.31	0.39	0.24

Complementary roles of VE-FSD and ME-FSD. On MIT (Table S3), VE-FSD scores higher than ME-FSD on 26 of 35 generators. On Video-ACID (Table S5), the ranking reverses for a subset of generators: ME-FSD leads on SVD (0.99 vs. 0.91), VC1 (1.00 vs. 0.96), Kling (0.79 vs. 0.73), and Vidu Q1 (0.78 vs. 0.75), among others. The fact that which component performs better changes across datasets (and across generators within each dataset) indicates that visual evolution residuals and motion evolution residuals capture different forensic microstructures, which motivates their combination in VFSD.

D Threshold Sensitivity and Cross-Dataset Transferability of VFSD

For a detector to be practically viable, its operating threshold must be easily calibrated and, ideally, transferable across different real-video domains. Our approach requires only real videos for this calibration, as the threshold τ is simply set to a high percentile of the real-video anomaly scores. We investigate the cross-domain portability of this threshold through two complementary experiments. First, we examine whether the optimal threshold τ^* has similar values for GMMs that are trained on different datasets. Second, we quantify the detection performance when a GMM and its associated optimal threshold τ^* are taken from one dataset and applied to another, measuring how effectively a small calibration set of real target-domain videos can recover any performance gap.

D.1 Threshold Sensitivity

Setup. For cross-dataset deployment to be practical, the VFSD score distributions induced by different real datasets must be compatible. This means a threshold that works well when trained on one dataset should not require large adjustments on another. To assess this, we compute VFSD scores for every video in our three real-video datasets together with scores for all videos from the 35 generators in our synthetic datasets. All scores are mapped to a common $[0, 1]$ scale using a single min-max normalization computed over the union of real and synthetic scores across all three datasets. This normalization ensures that threshold values are directly comparable across datasets.

Metric. For each dataset, we sweep the threshold τ and record the balanced accuracy $\mathrm{BA} = \frac{1}{2}(\mathrm{TPR} + \mathrm{TNR})$ averaged over all 35 generators; TPR is the rate of synthetic videos correctly flagged as synthetic, and TNR is the rate of real videos correctly classified as real.

Table S4: Per-generator zero-shot detection (AUC) on GenVideo. Numbers are averaged over variants within the same generator family.

		Generative Source																														
Method	**Avg ± Std**	**ADiff-v1**	**ADiff-v2**	**CausVid**	**CogVidX**	**EasyAnim**	**Hunyuan**	**IPOC**	**IPOW**	**JT-CV-9B**	**Kling**	**LanDiff**	**LTX-Vid**	**Luma**	**Magi T2V**	**Meta AI**	**Midj.**	**MiniMax**	**Miracle**	**Mochi**	**O-Sora**	**Pika 1.0**	**Pika 1.5**	**RepVid**	**Runway**	**Sora**	**SVD**	**StepVid**	**STIV**	**Vchit.**	**Veo 3**	**VC1**	**VC2**	**Vidu Q1**	**Wan 2.1**	**Wan 2.2**
Image-Based Detectors																																			
NPR [16]	0.83 ± 0.16	0.64	0.67	0.98	0.84	0.95	0.83	0.97	0.90	0.96	0.85	0.82	0.75	0.84	0.82	0.99	0.90	0.90	0.85	0.78	0.83	0.85	0.95	0.91	0.89	0.85	0.43	0.91	0.19	0.66	0.84	0.89	0.74	0.88	0.95	0.98
RINE [37]	0.74 ± 0.14	0.89	0.91	0.91	0.64	0.60	0.61	0.78	0.90	0.94	0.93	0.68	0.92	0.74	0.60	0.91	0.58	0.77	0.67	0.64	0.72	0.82	0.55	0.72	0.79	0.65	0.39	0.70	0.47	0.81	0.67	0.90	0.94	0.76	0.78	0.77
FSD [22]	1.00 ± 0.00	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	1.00	1.00	**1.00**	**1.00**	**1.00**
Video-Based Detectors																																			
DeMamba [10]	0.46 ± 0.04	0.47	0.51	0.52	0.40	0.48	0.43	0.50	0.44	0.50	0.45	0.48	0.41	0.49	0.42	0.43	0.53	0.47	0.47	0.42	0.46	0.51	0.47	0.49	0.42	0.47	0.40	0.39	0.40	0.45	0.40	0.45	0.48	0.43	0.43	0.48
D3 [49]	0.93 ± 0.04	0.83	0.86	0.97	0.88	0.94	0.93	0.93	0.97	0.94	0.95	0.94	0.93	0.96	0.95	0.97	0.93	0.96	0.96	0.95	0.97	0.96	0.95	0.92	0.96	0.91	0.83	0.94	0.84	0.88	0.95	0.87	0.94	0.90	0.94	0.92
AIGVDet [47]	0.87 ± 0.10	0.92	0.97	0.74	0.85	0.94	0.88	0.95	0.93	0.96	0.87	0.83	0.86	0.90	0.87	0.86	0.94	0.95	0.56	0.74	0.86	0.97	0.95	0.77	0.94	0.79	0.99	0.56	0.89	0.82	0.91	0.79	0.92	0.83	0.94	0.94
FTCN [8]	0.58 ± 0.04	0.64	0.60	0.58	0.53	0.53	0.53	0.58	0.63	0.60	0.57	0.59	0.61	0.59	0.52	0.54	0.54	0.60	0.68	0.60	0.56	0.58	0.61	0.61	0.57	0.52	0.62	0.61	0.60	0.61	0.51	0.61	0.58	0.57	0.58	0.59
Vahdati [19]	0.86 ± 0.11	0.91	0.97	0.78	0.73	0.58	0.77	0.88	0.95	0.98	0.94	0.87	0.95	0.86	0.69	1.00	0.95	0.87	0.97	0.73	0.86	0.84	0.93	0.88	0.86	0.95	1.00	0.84	0.97	0.64	0.87	0.91	0.64	0.80	0.93	0.96
Ours																																			
ME-FSD	1.00 ± 0.00	1.00	1.00	0.99	1.00	0.99	1.00	1.00	1.00	**1.00**	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.99	1.00	0.99	0.98	1.00	1.00	1.00
VE-FSD	1.00 ± 0.00	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	1.00	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
VFSD	**1.00** ± 0.00	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**

Table S5: Per-generator zero-shot detection (AUC) on Video-ACID. Numbers are averaged over variants within the same generator family.

		Generative Source																														
Method	**Avg ± Std**	**ADiff-v1**	**ADiff-v2**	**CausVid**	**CogVidX**	**EasyAnim**	**Hunyuan**	**IPOC**	**IPOW**	**JT-CV-9B**	**Kling**	**LanDiff**	**LTX-Vid**	**Luma**	**Magi T2V**	**Meta AI**	**Midj.**	**MiniMax**	**Miracle**	**Mochi**	**O-Sora**	**Pika 1.0**	**Pika 1.5**	**RepVid**	**Runway**	**Sora**	**SVD**	**StepVid**	**STIV**	**Vchit.**	**Veo 3**	**VC1**	**VC2**	**Vidu Q1**	**Wan 2.1**	**Wan 2.2**
Image-Based Detectors																																			
NPR [16]	0.53 ± 0.19	0.29	0.33	0.86	0.48	0.75	0.48	0.75	0.61	0.75	0.52	0.47	0.59	0.55	0.42	0.82	0.60	0.61	0.51	0.42	0.43	0.52	0.71	0.59	0.56	0.48	0.11	0.58	0.05	0.28	0.49	0.55	0.32	0.54	0.71	0.85
RINE [37]	0.82 ± 0.10	0.92	0.94	0.93	0.75	0.74	0.74	0.85	0.93	0.96	**0.95**	0.78	0.94	0.82	0.73	0.93	0.69	0.84	0.78	0.76	0.80	0.87	0.69	0.81	0.86	0.75	0.58	0.80	0.63	0.87	0.78	0.93	0.95	**0.84**	0.85	0.84
FSD [22]	0.78 ± 0.15	0.86	0.85	1.00	0.61	0.97	0.60	0.61	0.85	0.74	0.70	0.89	0.87	0.76	0.67	0.95	0.76	0.58	0.90	0.90	0.87	0.75	0.60	0.86	0.63	0.89	0.86	0.95	0.90	0.95	0.53	0.92	1.00	0.51	0.54	0.56
Video-Based Detectors																																			
DeMamba [10]	0.57 ± 0.04	0.57	0.61	0.64	0.51	0.59	0.54	0.62	0.55	0.62	0.58	0.60	0.52	0.60	0.54	0.55	0.64	0.58	0.57	0.53	0.57	0.64	0.58	0.64	0.55	0.58	0.51	0.51	0.51	0.57	0.52	0.57	0.59	0.55	0.55	0.60
D3 [49]	0.84 ± 0.12	0.56	0.64	0.97	0.72	0.88	0.85	0.86	0.95	0.89	0.92	0.88	0.85	0.95	0.91	0.97	0.84	**0.95**	0.93	0.89	0.96	0.94	**0.93**	0.83	0.93	0.81	0.51	0.88	0.57	0.68	**0.91**	0.70	0.88	0.76	**0.88**	0.84
AIGVDet [47]	0.77 ± 0.12	0.83	0.91	0.61	0.73	0.85	0.77	0.86	0.83	0.90	0.77	0.70	0.74	0.79	0.76	0.73	0.87	0.86	0.46	0.62	0.73	0.90	0.85	0.65	0.86	0.68	0.95	0.45	0.77	0.71	0.82	0.70	0.84	0.72	0.87	0.85
FTCN [8]	0.60 ± 0.04	0.66	0.61	0.58	0.55	0.56	0.55	0.60	0.65	0.63	0.58	0.62	0.62	0.61	0.54	0.56	0.55	0.60	0.69	0.60	0.57	0.61	0.64	0.63	0.61	0.54	0.65	0.62	0.62	0.63	0.52	0.63	0.60	0.58	0.60	0.60
Vahdati [19]	0.76 ± 0.13	0.80	0.87	0.65	0.60	0.48	0.64	0.75	0.88	0.91	0.85	0.77	0.90	0.74	0.58	0.97	0.87	0.74	0.85	0.62	0.75	0.70	0.80	0.75	0.73	0.87	0.98	0.72	0.91	0.54	0.74	0.81	0.53	0.67	0.82	**0.87**
Ours																																			
ME-FSD	0.93 ± 0.07	1.00	1.00	0.99	0.97	0.99	0.87	0.96	0.99	**1.00**	0.79	0.99	1.00	0.96	0.91	1.00	0.98	0.83	0.98	0.99	1.00	0.92	0.85	0.99	0.96	0.99	0.99	0.99	1.00	0.99	0.84	1.00	1.00	0.78	0.83	0.76
VE-FSD	0.93 ± 0.08	1.00	1.00	0.98	0.98	0.98	0.87	0.98	0.99	1.00	0.73	0.99	1.00	0.95	0.89	0.99	0.94	0.89	0.98	1.00	0.97	0.94	0.88	0.99	0.98	1.00	0.91	1.00	1.00	0.99	0.87	0.96	0.99	0.75	0.85	0.74
VFSD	**0.96** ± 0.06	**1.00**	**1.00**	**1.00**	**0.99**	**1.00**	**0.92**	**0.99**	**1.00**	1.00	0.85	**1.00**	**1.00**	**0.99**	**0.95**	**1.00**	**0.99**	0.90	**1.00**	**1.00**	**1.00**	**0.98**	0.91	**1.00**	**0.99**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.89	**1.00**	**1.00**	0.79	0.88	0.80

Results. The results of this experiment are in Figure S2. Despite each GMM being trained independently on a *single* real-video dataset, the balanced-accuracy curves peak in a similar region of the shared threshold axis. For VFSD, the optimal normalized threshold τ^* spans $\log_{10}\tau^* \in [-2.49, -1.92]$ across the three datasets, a range of 0.57 in log-scale. Noticeably, the optimal thresholds in log space for MIT (-2.32) and Video-ACID (-2.49) are remarkably close. This corresponds to only a 0.0015 difference in normalized threshold space. For ME-FSD, optimal threshold values are even more similar ($[-2.40, -2.12]$, range 0.28 in log space). This alignment suggests that VFSD's score distributions are broadly compatible across real-video datasets: a threshold chosen on one dataset would land close to the optimal region for the others. However, proximity in a normalized score space does not directly tell us how much detection performance is gained or lost in practice when τ is set on one dataset and applied to another without modification. The next experiment measures this directly.

D.2 Quantifying Threshold Transfer

Setup. We simulate a realistic deployment scenario: if a model is trained on one real-video dataset (the source) and deployed on another (the target), how is the balanced accuracy affected when τ is transferred without modification? Furthermore, if a small set of real videos from the target domain is available for recalibration, how much performance is recovered? For each source-target pair, we compare two conditions:

- **Naive transfer.** The threshold is set to the 95th percentile (P_{95}) of *source* real-video VFSD scores. No data from the target domain is used at all.
- **Target-data calibration.** The source threshold is replaced by $\tau = P_{95}$ of VFSD scores on 100 real videos drawn from the target domain. This requires no AI-generated videos. This simulates a scenario where a practitioner can collect a small held-out set of real videos from the new deployment domain before going live.

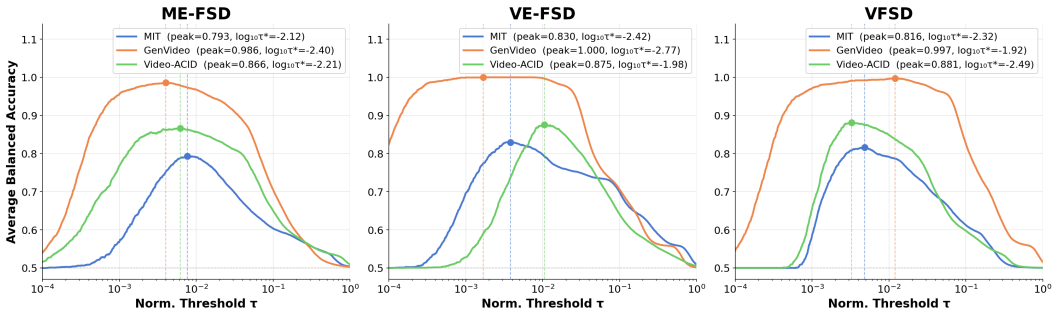


Figure S2: Threshold sensitivity curves. The plots show balanced detection accuracy vs. detection threshold for three real datasets. Scores are min-max normalized jointly across all datasets so that threshold positions are directly comparable. The normalization is applied independently to ME-FSD, VE-FSD, and VFSD. Dots mark the per-dataset optimum threshold; the narrow spread of peaks shows that the optimal operating point is largely stable across datasets. The threshold axis is shown on a $\log_{10}$ scale to make the low-threshold region visible.

Table S6: Balanced accuracy for each (source GMM, target dataset) pair. *Naive*: $\tau = P_{95}$ of source real scores. *Calib*: $\tau = P_{95}$ of 100 target real videos (no synthetic data). Diagonal entries (*) are in-distribution results (source = target).

	Target: MIT		Target: GenVideo		Target: Video-ACID	
GMM source	Naive	Calib	Naive	Calib	Naive	Calib
MIT	0.815*	0.815*	0.992	0.986	0.874	0.827
GenVideo	0.619	0.812	0.975*	0.975*	0.687	0.849
Video-ACID	0.797	0.814	0.994	0.944	0.860*	0.860*

A single GMM is fitted for each source domain and only the scalar threshold τ changes between target domains. This experiment only examines VFSD and balanced accuracy (averaged over all 35 generators) is reported.

Results. The results of this experiment are shown in Table S6. Most naive transfers succeed: three of the six off-diagonal cells reach ≥ 0.87 balanced accuracy, and both MIT and Video-ACID as source datasets transfer well across all targets. The exception is GenVideo as a source: when the threshold set on GenVideo is applied naively to MIT and Video-ACID, the balanced accuracy drops to 0.619 and 0.687, respectively, well below the in-distribution baselines of 0.815 and 0.860.

After target-data calibration, all six off-diagonal entries reach at least 0.812 balanced accuracy. The two entries that failed under naive transfer recover substantially: GenVideo→MIT rises from 0.619 to 0.812 (within 0.003 of the in-distribution value of 0.815), and GenVideo→Video-ACID rises from 0.687 to 0.849 (within 0.011 of 0.860). The remaining four off-diagonal cells, which already performed adequately under naive transfer, stay in the range 0.814–0.986 after calibration.

D.3 Connecting the Two Experiments

Figure S2 shows that, under a common score scale, the optimal operating point τ^* occupies a narrow band across all three datasets – a qualitative indication that VFSD's score distributions are compatible across real-video datasets. Table S6 quantifies this: when source and target distributions overlap well (MIT and Video-ACID), a naively transferred τ already preserves most detection performance. When the optimal threshold difference is larger (*e.g.*, GenVideo→MIT), the performance gap is larger under naive transfer but mostly closed by resetting τ using just 100 real videos from the target domain without any retraining or using synthetic data.

Together, the two experiments provide evidence that VFSD's detection capability is not strongly tied to the specific real-video dataset used for training: the learned score space generalizes across domains, and the operating threshold can be reliably adapted with a small set of real videos from the target domain.

E Optical Flow Estimator Invariance

While VFSD requires optical flow to capture motion dynamics, it does not rely on any specific optical flow estimator. To demonstrate estimator invariance, we conducted an experiment swapping RAFT [55] with two alternative high-performance architectures: SEA-RAFT [68] and WAFT [69]. Crucially, we kept the visual and motion predictors frozen (using the exact weights trained on RAFT) and only refitted the downstream GMMs to the resulting real-video self-descriptions.

Table S7 presents zero-shot detection performance (AUC) across four representative generative models (IPOW, Midjourney, Mochi, and Luma). In all configurations, VFSD maintains strong discriminative performance, achieving 0.941 average AUC with WAFT and 0.931 in the worst case with SEA-RAFT. The slight performance drop observed when substituting RAFT stems from an input distribution shift relative to the RAFT-trained predictors, which can be readily mitigated by training the linear predictors directly on the target flow estimator.

Table S7: Zero-shot detection AUC across optical flow estimators on representative generative models. Predictors f_I and f_O trained with RAFT are kept frozen; only GMMs are refitted.

	Optical Flow Estimator		
Generator	**RAFT** (Baseline) [55]	**WAFT** [69]	**SEA-RAFT** [68]
IPOW	**1.000**	0.984	0.971
Midjourney	**0.997**	0.945	0.939
Mochi	**1.000**	0.981	0.960
Luma	**0.974**	0.851	0.854
Average	**0.992**	0.941	0.931

F Anomaly Detection Architecture Ablation

In addition to the anomaly detector ablation in the main paper (Table 5), where GMM was evaluated against One-Class SVM and Isolation Forest, here we further benchmark against two modern one-class architectures (Deep SVDD [70] and SimpleNet [71]) applied to the extracted self-descriptions.

As shown in Table S8, GMM achieves the highest overall discrimination (0.992 average AUC), slightly outperforming Deep SVDD (0.986) and substantially outperforming SimpleNet (0.909). Because 3D AR self-descriptions already provide a compact and structured representation space, explicit density estimation with a GMM accurately captures the real-video distribution.

Table S8: Zero-shot detection AUC comparing GMM against anomaly detection architectures trained on real-video self-descriptions.

Generator	**GMM** (Ours)	**Deep SVDD** [70]	**SimpleNet** [71]
IPOW	**1.000**	0.996	0.999
Midjourney	0.997	**1.000**	**1.000**
Mochi	**1.000**	0.999	0.984
Luma	**0.974**	0.952	0.653
Average	**0.992**	0.986	0.909

G Predictor Architecture Ablation

We intentionally chose a linear predictor with few parameters to reduce the chance of overfitting to the training dataset and to improve generalizability. More complex non-linear predictors, such as neural network-based predictors, contain a large number of parameters, which leave them vulnerable to overfitting. As a result, they obtain strong performance during training but poorer performance during testing.

To show this, we ran an experiment where we swapped our linear convolutional predictor with a shallow CNN and an MLP under the same training protocol and measured the zero-shot detection

Table S9: Comparison of zero-shot detection performance (AUC) across different predictor architectures.

Generator	Default Linear	CNN Predictor	MLP Predictor
IPOW	**1.000**	0.561	0.604
Midjourney	**0.997**	0.696	0.827
Mochi	**1.000**	0.702	0.657
Luma	**0.974**	0.706	0.835
Average	**0.992**	0.666	0.731

Table S10: Robustness evaluation under resizing and video speed manipulations (evaluated in AUC).

Generator	Unaltered	scale_1.1x	scale_0.9x	scale_0.75x	speed_1.2x	speed_1.1x	speed_0.9x	speed_0.8x
IPOW	1.000	0.964	0.951	0.989	0.998	0.978	0.993	0.967
Luma	0.974	0.857	0.852	0.852	0.808	0.832	0.864	0.826
Midjourney	0.997	0.963	0.953	0.960	0.958	0.960	0.980	0.973
Mochi	1.000	0.978	0.947	0.996	0.998	0.980	0.984	0.964
Average	0.992	0.940	0.926	0.949	0.940	0.937	0.955	0.932

performance. The results are provided in Table S9 (all numbers are AUC). These results show that our simple linear predictors outperform these non-linear predictors at inference time.

We note that these neural networks performance numbers are likely due to overfitting which may be mitigated with more training data. A thorough investigation of non-linear predictors is beyond the scope of this paper and we will leave it for future work.

H Additional Robustness Experiments

Our method shows strong robustness to compression as demonstrated in Section 6, Table 4. We additionally ran several robustness experiments under resizing and video speed manipulation. The results of these experiments are provided in Table S10 (all numbers are AUC).

These results show that our method is highly robust under these conditions and achieves similar detection performance. The worst-case performance is 0.926 AUC under resizing by a factor of $0.9\times$.

I Time Complexity Analysis

The computational cost of our zero-shot detection framework during inference is primarily bottlenecked by the initial optical flow extraction and the constrained least-squares optimization required to fit the autoregressive self-descriptions. Because our decoupled visual and motion predictors (f_I and f_O) are implemented as strictly linear convolutions, their forward-pass overhead is negligible. Furthermore, the final generative likelihood scoring is performed in a highly compact, low-dimensional parameter space. Consequently, the total inference complexity scales linearly with the video length T and spatial resolution $H \times W$. The time complexities of the primary pipeline components are summarized in Table S11.

J GenVideo Results

We evaluated our approach on each of the AI video generators in the GenVideo dataset by using a GMM trained on the real subset of the dataset. The results of this experiment are provided in Table S12. The performance of our method on this dataset remains very high (all numbers are AUC).

Table S11: Summary of inference time complexity. T denotes the number of frames, $H \times W$ the spatial resolution, d the maximum length of the extracted self-description vectors ($d = \max(d_I, d_O)$), and M the number of GMM components.

Pipeline Component	**Time Complexity**
Residual Extraction (Optical Flow & Linear Predictors)	$\mathcal{O}(T \cdot HW)$
AR Self-Description Fitting	$\mathcal{O}(T \cdot HW \cdot d^2)$
GMM Likelihood Scoring	$\mathcal{O}(M \cdot d^2)$

Table S12: Zero-shot detection performance (AUC) evaluated across individual generators in the GenVideo dataset.

Generator	Crafter	Gen2	HotShot	Lavie	ModelScope	MoonValley	MorphStudio	Show_1	Sora	WildScrape
AUC	1.000	1.000	0.999	0.997	0.999	1.000	1.000	0.998	1.000	0.997

K AR Frequency Response Computation for Figure 3

To visualize the spectral microstructures shown in Figure 3 of the main paper, we compute the 2D spatial frequency response of the learned FSD parameters (Φ for VE-FSD and Z for ME-FSD). We describe the procedure below for Φ; the computation for Z is identical.

From the 3D spatio-temporal AR kernel, we extract the spatial slice corresponding to temporal lag zero ($\tau = 0$), denoted Φ_0. Φ_0 is then zero-padded onto an $M \times M$ grid ($M = 256$), yielding $\tilde{\Phi}_0$. The centered log-magnitude frequency response $S \in \mathbb{R}^{M \times M}$ is then computed as:

$$S = \log_{10}\left(\left|\mathrm{fftshift}\left(\mathcal{F}_{2\mathrm{D}}(\tilde{\Phi}_0)\right)\right| + \epsilon\right), \tag{S1}$$

where $\mathcal{F}_{2\mathrm{D}}$ denotes the 2D Discrete Fourier Transform, $\mathrm{fftshift}$ shifts the zero-frequency (DC) component to the center of the spectrum, and $\epsilon = 10^{-8}$ ensures numerical stability. For each generator (and real videos), the displayed fingerprint is obtained by averaging across videos:

$$\bar{S}^{(g)} = \frac{1}{N} \sum_{i=1}^{N} S_i^{(g)}. \tag{S2}$$

L Broader Societal Impacts

The development of robust, zero-shot detectors like Video Forensic Self-Descriptions (VFSD) offers significant positive societal impacts by providing scalable tools to combat visual misinformation. Integrating such frameworks into social media platforms can reliably inform users about the provenance of digital media, fostering a more trustworthy information ecosystem. Conversely, a potential negative impact stems from the dual-use nature of forensic technologies. Malicious actors could employ our detection framework as an adversarial objective during the training of new generative models, explicitly optimizing them to suppress temporally distributed forensic microstructures and synthesize highly evasive videos that circumvent current detection pipelines.

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 claims made in the abstract and the introduction are supported by Sections 3-7.

   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: The limitations are discussed in Section 6. Computational efficiency is discussed in Appendix I.

   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: All experimental details are discussed in Section 5 and Appendix A enabling reproducibility of the main results.

   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: [Yes]

Justification: Code and reproducibility instructions will be made publicly available upon publication at `https://arefazizpour.github.io/VFSD-NeurIPS2026/`.

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: All details are discussed in Sections 3-5, Appendix A, and Appendix B.

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: [No]

Justification: Due to computational constraints, we were unable to include statistical significance analysis. However, the results reported in the paper are averaged over 35 generators for detection and 7 sources for attribution. Additionally we used multiple real datasets to ensure our method produces consistent performance.

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: The related information is reported in Section 5 and Appendix I.

   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 conducted in this paper conforms with the NeurIPS Code of Ethics in every respect.

   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: A Broader Societal Impacts section is included in Appendix L.

    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: Our paper poses no critical risks for misuse.

    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: [Yes]

    Justification: The creators and original owners of all assets are cited and properly credited in the main paper and appendix, and all terms of use are respected.

    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 introduce new datasets or benchmarks. All code and documentation will be released at `https://arefazizpour.github.io/VFSD-NeurIPS2026/`.

    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 nor research with human subjects.

    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 crowdsourcing nor research with human subjects.

    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 in this research does not involve LLMs as any important, original, or non-standard components.

    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.