

Active Light Modulation to Counter Manipulation of Speech Visual Content

Hadleigh Schwartz¹, Xiaofeng Yan¹, Charles J. Carver^{2*}, and Xia Zhou¹

¹Department of Computer Science, Columbia University ²Lincoln Laboratory, Massachusetts Institute of Technology
 {hadleigh, xia}@cs.columbia.edu, xy2600@columbia.edu, carver@mit.edu

Abstract

High-profile speech videos are prime targets for falsification, owing to their accessibility and influence. This work proposes Spotlight, a low-overhead and unobtrusive system for protecting live speech videos from visual falsification of speaker identity and lip and facial motion. Unlike predominant falsification detection methods operating in the digital domain, Spotlight creates dynamic physical signatures at the event site and embeds them into all video recordings via imperceptible modulated light. These physical signatures encode semantically-meaningful features unique to the speech event, including the speaker’s identity and facial motion, and are cryptographically-secured to prevent spoofing. The signatures can be extracted from any video downstream and validated against the portrayed speech content to check its integrity. Key elements of Spotlight include (1) a framework for generating extremely compact (i.e., 150-bit), pose-invariant speech video features, based on locality-sensitive hashing; and (2) an optical modulation scheme that embeds >200 bps into video while remaining imperceptible both in video and live. Prototype experiments on extensive video datasets show Spotlight achieves AUCs ≥ 0.99 and an overall true positive rate of 100% in detecting falsified videos. Further, Spotlight is highly robust across recording conditions, video post-processing techniques, and white-box adversarial attacks on its video feature extraction methodologies.

1 Introduction

In the early days of video technology, high-profile speeches were some of the first events to be shared over the new communication medium [1]. Influential figures capitalized on its unique persuasive power [2, 3], and ever since, video has been a staple of information exchange. Today, this exchange faces a flood of falsified videos of high-profile speeches spreading disinformation and discord.

This paper focuses on addressing falsification of two salient aspects of a speech event: the speaker’s identity and her lip and face movements, which are directly tied to speech content and speed. These elements are particularly persuasive and semantically-rich [4] and have been targeted in numerous incidents [5–12]. Today, realizing these falsifications is easier than ever. An attacker may use any open-source model or online deepfake tool to generate videos of her victim making fabricated statements [6–9, 11]. Even simple edits achievable on most smartphones, such as changing playback speed or splicing clips, can greatly alter a portrayal (e.g., widely-circulated videos claiming to show Nancy Pelosi and Kamala Harris delivering speeches intoxicated [5, 10, 12]). Once disseminated online, these videos blur the lines between fake and

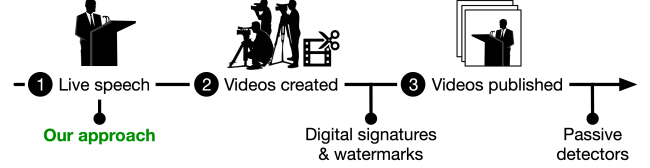


Figure 1: We combat falsification of live speech videos by initiating protection at the earliest possible stage of a video’s lifetime: the physical scene of the speech.

real, posing grave political, financial, and social risks. Current technologies for detecting falsification of speech videos have failed to combat these threats.

Existing technologies and their limitations can be characterized based on the stage of a video’s lifetime they initiate protection (Figure 1). A majority of these methods are *passive*, aiming to detect falsifications *after the video is published* by identifying artifacts introduced by editing or deepfake models [13–15]. These visual imperfections, however, are diminishing with advances in generative AI, and passive detection methods are increasingly bypassed [16, 17].

In response, efforts have increasingly shifted to protecting videos at *their digital creation*. Such methods either incorporate watermarks into generated content [18] or tag videos with digitally signed credentials upon capture [19–22], generation, or editing [23]. While promising, these approaches face several practical barriers stemming from their reliance on cooperation from all recording parties and video creators. (1) Methods adding verification information to fake content are unlikely to be adopted by malicious parties. With deepfake and editing technologies increasingly democratized, attackers can easily create videos lacking watermarks or credentials and then claim they are real. (2) Capture-time tools require use of specialized apps [19] or hardware [20–22], which cannot be guaranteed in the large, unregulated audiences of public speech events. (3) Because digital signatures are bound to a video’s low-level representation (i.e., its pixel values), they must be regenerated when any post-processing techniques are applied. These techniques, such as compression and transcoding, are exceedingly common in today’s video sharing workflows. Digital signature methods thus assume user cooperation at each stage of a video’s lifetime. As we see today, without such uniform compliance, an indistinguishable mix of real and fake unsigned content is produced, sowing public confusion about the underlying events.

This work studies a complementary physical approach to the protection of speech videos, seeking to shift protection agency from recording parties to speakers themselves. We envision speakers deploying a device that creates signatures *physically* at the speech site, such that they are naturally embedded into *all real* recordings. Such physical signatures encode semantically-meaningful features

*Work done while at Columbia University.

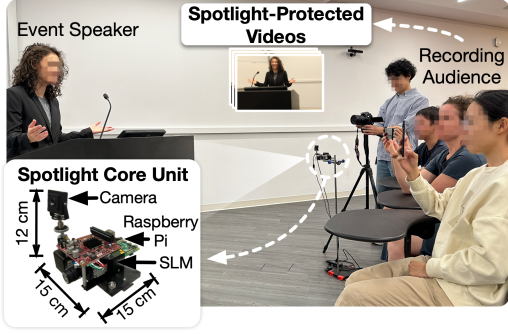


Figure 2: Spotlight protecting a live speech. The low-cost core unit creates signatures encoding the event and embeds them into *all* videos by projecting imperceptible light with a spatial light modulator (SLM).

that are unique to the event and consistent across recording device and position (e.g., representations of speaker lip motion). Legitimate videos inherently pass validation against the signature, while falsified videos possess diverging features and are thus detected. Further, physical signatures are *cryptographically-secured*, preventing their forgery. While digital signature methods require all parties to independently sign their videos, here, cryptographic data is generated only once, on the deployed physical signature creation device.

Physical signatures provide several benefits in the context of live speeches, owing to their uniquely early initiation of protection. (1) They inherently protect all videos at the event without demanding recording device cooperation, thus creating a canonical version of the speech reinforced by each filming attendee. (2) Since physical signatures capture higher-level features rather than low-level pixel values, they remain valid after benign edits that preserve the video’s semantic content (e.g., compression).

This paper demonstrates Spotlight, a physical signature platform that disseminates signatures via modulated light at the speech site (Figure 2). A speaker places a low-cost Spotlight core unit, serving as trustworthy third-party witness, at the site. The core unit continually extracts semantically-meaningful and robust visual features specific to the speaker’s identity and face and lip motion (referred to as a digest) and generates a message authentication code (MAC) for the digest using its private key. The digest and its MAC make up the signature. Spotlight then encodes the signature data as optical modulations that remain *imperceptible* both live and in videos, supporting the platform’s broader adoption. These optical modulations nonetheless manifest in all recordings as decodable pixel-level changes. A published video can be verified at any point in its lifetime by extracting the optical signatures and comparing recovered digests to those computed on the portrayed speech event.

Two main technical challenges arise in realizing Spotlight. (1) The limited frame rates of typical cameras result in a low embedding data capacity (hundreds of bits per second). Thus, digests must be highly compact yet highly descriptive and consistent across camera positions. (2) Optical signature embedding must balance competing objectives of imperceptibility, robustness, and data capacity.

We address these challenges with the following contributions. (1) We propose a framework based on locality-sensitive hashing (LSH) to compress pose-invariant, semantically-meaningful speech video

feature vectors to just 150 bits while preserving their performance. This framework supports diverse feature vectors, of arbitrarily high-dimensionality, and is independent of the signature dissemination modality. (2) We design a spatio-temporal light modulation scheme that boosts bandwidth across *all* RGB cameras while remaining imperceptible and resilient against common video post-processing techniques. To the best of our knowledge, this is the first scheme for embedding invisible information into videos from within the environment, taking an important step towards practical physical signatures. (3) We fabricate a Spotlight prototype and examine its performance on 257 minutes of live speeches captured with our core unit deployed and over 1,300 pairs of real and deepfaked videos, spanning varied falsification granularities. Additionally, we assess its imperceptibility and robustness in diverse recording scenarios. (4) We evaluate Spotlight’s robustness against extensive countermeasures, including two sophisticated white-box adversarial attacks aiming to create falsified content that evades detection.

We summarize our key findings below:

- Spotlight’s LSH framework supports over 100-fold reduction in the representation size of generic speech video features while maintaining their verification performance.
- Spotlight attains Area Under Curves (AUCs) ≥ 0.99 and a true positive rate of 100% in detecting falsifications of speaker identity and face and lip motion. In challenging scenarios where as little as 1.35 s of a video is modified, Spotlight achieves an AUC of ≥ 0.90 , a 40% gain over the best passive detector baseline evaluated.
- Spotlight’s semantically-meaningful digests are robust to varied countermeasures and white-box adversarial attacks.
- Spotlight supports video recording with any RGB camera, at viewing angles up to 60° and distances up to 3 m, even when videos are captured with no optical zoom.
- Spotlight achieves error-free signature data extraction and verification of videos recorded in extensive indoor and outdoor environments, as well as after common video post-processing methods such as compression, transcoding, and filter application.
- User studies and LPIPS scores confirm our optical signatures are imperceptible live and in-video in varied deployment scenarios.

2 Background and Related Work

2.1 Video Falsification

This paper focuses on combating visual falsifications of speaker identity and face and lip motion, which determine delivered content. Such falsifications can be made using traditional techniques such as framerate modification, trimming, and cropping. Increasingly, they are achieved via the following deepfake techniques. (1) *Face reenactment* uses a source video to drive the facial movements of a target image. For example, an attacker may use frames from a legitimate speech video as their targets to create reenactment deepfakes modifying the speaker’s lip movements to match audio of falsified statements [9]. (2) *Identity/face swaps* replace a victim’s face with that of another identity while maintaining the victim’s facial motion and expression; (3) *Complete face synthesis* generates fictitious faces (i.e., non-existent identities). For speech video falsification, this form of deepfake has the same end effect as an identity swap deepfake, as the perceived identity of the speaker is changed.

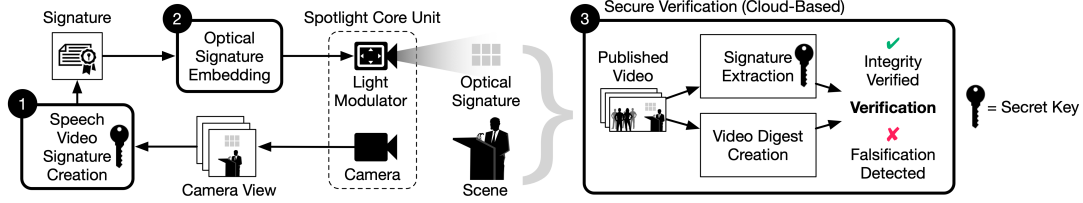


Figure 3: Overview of Spotlight’s modules and workflow, including: (1) the speech video signature creation; (2) optical signature embedding during the live speech; (3) integrity verification of a later published video of the speech by the video verification module.

2.2 Preventing and Detecting Fake Videos

We organize existing techniques for preventing and detecting falsified and fake videos into four categories as below. We discuss mechanisms exclusively targeting AI-generated videos (i.e., deepfakes), as well as broader media authenticity initiatives.

Passive detectors These techniques analyze videos for evidence of tampering and typically are designed to detect deepfakes. They hone in on high-level physical inconsistencies [14, 24–36] (e.g., unnatural lip movements), biometric incongruities [37–41] (e.g., lack of identity-specific head movements), or pixel-level artifacts [42–55] (e.g., anomalous spatial frequencies). Several works train generic classification models on deepfakes [13, 56–71] to learn anomalies.

Unfortunately, passive detectors are increasingly evaded and even hijacked to improve the quality of fake videos. For instance, [16] bypasses remote photoplethysmography (rPPG)-based detection methods [15] by generating faces with realistic rPPG signals. Adversarial attacks have conquered several passive detectors [17].

Digital signing and watermarking These active techniques add information to digital content at its creation to enable immediate or downstream verification. Emerging frameworks append cryptographically-signed provenance metadata to files at recording time [19, 20, 22] or upon editing [21, 23]. Digital watermarks directly embed verification signals into real [72–77] or synthetic [18] media. Overall, these digital methods require the cooperation of all video sources throughout the information ecosystem, since they add information on a per-video level. Unfortunately, adversaries are de incentivized from participating in these frameworks and can find alternatives to create synthetic media lacking such information.

Live QR codes Two prior works display dynamic QR codes by a speaker to disseminate verification information, both addressing only falsification of speech audio. Critch [78] displays QR codes that encode a speech’s transcript but does not prototype or evaluate the idea. In [79], time-frequency features of audio signals are encoded. Because of these features’ lower-level nature, they lack robustness across key recording conditions (e.g., their accuracy is below 90% at distances beyond 2 ft and in the presence of ambient noise) and are not shown to be pose-invariant or adversarially robust. Our work instead focuses on visual falsifications and proposes a generic methodology for compressing pose-invariant, semantically-meaningful features. Our features are robust against diverse ambient conditions and evaluated adversarial attacks.

Independent of the verification features they carry, QR codes impose a strict tradeoff between level of obtrusiveness and supported camera distance and angle. Prior works show a QR code must be over 20 x 20 cm in size to ensure it is decodable by all devices

within 3 m and 45° [80]. While users can use optical zoom to boost recording range, this simply fills a larger portion of the view with the QR code. Consequently, QR code-based systems force users to accept either reduced protection robustness or large flickering QR codes in their videos – a critical barrier to practical adoption.

Liveness detection with light A related line of work employs active illumination from screens to verify that video chat participants are real [81–84]. These works follow a challenge-response model, where facial appearance is analyzed with respect to dynamic screen illumination. They are thus constrained to video chat scenarios.

3 Preliminaries

Spotlight is a physical and proactive approach to protecting live speech videos from visual falsification of speaker identity as well as lip and face movements, which reflect delivered content and speech speed. To match the visual nature of these falsifications, we embed physical signatures with light – the sensing modality of vision. In this section, we describe Spotlight’s design and threat model, and then discuss the technical challenges it overcomes.

3.1 System Overview

Figure 3 overviews the Spotlight design, comprising three modules. The Spotlight core unit, deployed at the site of a live speech, consists of a camera observing the event and light modulator positioned to project light onto any approximately planar surface in the immediate vicinity of the speaker (e.g., a small portion of a podium, wall, backdrop, or curtain). Notably, each of the speech environments portrayed in recent high-profile falsification incidents [5–12] possess such a surface. We believe that, in practice, it is rare to find a high-profile speech event where this is not the case.

For each window of camera video frames, the signature creation module running on the core unit extracts a digest from frames. The digest contains semantically-meaningful features capturing speaker identity and lip and face motion and additional provenance metadata such as a window timestamp. The core unit then generates a message authentication code (MAC) for the digest with its secret key. A digest and its MAC comprise a signature. The optical signature embedding module encodes the signature data as optical modulations projected into the scene in the subsequent window of time. Thus, signatures are naturally embedded into real recordings.

To verify a published video of a speech, the verification module extracts optical signatures from video frames and validates the MACs to confirm digests’ integrity. It then examines whether the semantically-meaningful features recovered from the digests match those of the portrayed speech. To ensure that the secret key used

to validate MACs is secured, we envision the verification module as a secure cloud service queried by users and media platforms. As such, secret keys will never be exposed to third-party devices. We elaborate on Spotlight’s use of secret keys and MACs in §4.3.

3.2 Threat Model

Our threat model focuses on three entities: video producers, video verifiers, and attackers. Our focus is preventing attacks wherein falsified videos purport to be real. We do not address the inverse, wherein an attacker claims real content is fake.

Video producers create and disseminate *legitimate* videos of the speech. This group includes viewers recording at the event and non-malicious parties re-distributing videos. Audience members can record using any RGB camera with a frame rate ≥ 24 FPS and resolution $\geq 1080p$. We assume that the speaker’s face is visible in recordings at a maximum viewing angle and distance of 60° and 3 m. Videos can be saved using common codecs (e.g., H.265, MPEG-4) and post-processed with compression, transcoding, and filter application. We assume cameras remain still throughout the speech but discuss reasonable solutions to avoid this requirement in §12.

Video verifiers are individuals or media platforms (e.g., Facebook, YouTube, X) who seek to confirm the integrity of a video by providing it to the verification module.

Attackers disseminate falsified videos claiming to portray the speech event. We assume they possess white-box knowledge of Spotlight as well as significant computational power, and carry out attacks *after* a speech has taken place in an attempt to spread false media or undermine Spotlight verification. We do not consider environment-level attacks at the speech (including injection of interfering light, as discussed in §12) or tampering of Spotlight’s hardware. Attackers can perform any combination of the following:

- (1) Falsify the speaker’s lip and face motion and/or identity via visual edits (optionally via joint audio manipulation), using traditional techniques or deepfake models.
- (2) Access all Spotlight algorithms and models, including weights.
- (3) Create arbitrary completely synthetic speech videos.
- (4) Incorporate valid signatures from other Spotlight-protected videos into generated fake videos, i.e., replay attack.
- (5) Modify video speed or re-order legitimate video frames.
- (6) Remove or manipulate a video’s embedded signatures.
- (7) Digitally add to any video pixel-level signals that mimic the optical embedding scheme.

We assume that our private key is securely held out of reach of the adversary. Attacks on cryptographic primitives are out of scope.

Real-world limitations Spotlight is but one technical approach to achieving physical signatures and is not without limitations. (1) It requires deployment of the Spotlight core unit at speeches. (2) Because physical signatures are based upon a set of extracted event features, they do not address falsification of features outside this set. In this work, we develop a prototype leveraging visual features of speaker identity and lip and face motion. The prototype thus does not protect against falsifications of facial attributes (e.g., makeup) or non-facial elements (e.g. clothing, surrounding environment). Spotlight’s LSH framework is highly flexible, supporting rich feature sets. Nonetheless, an operator must choose such features based on anticipated attacks. (3) Videos must contain the complete optical

signature projection region to enable verification; as with digital signatures, videos lacking intact optical signatures are viewed as untrustworthy. Signature inclusion can be made highly likely by using a projection region close to the speaker’s face or even configuring a small but visible projection border as a cue for filmers.

While we believe that several of the aforementioned limitations can be addressed via further research (§12), we ultimately view Spotlight not as a panacea to fake speech videos, but rather a complement to passive detectors and digital signature methods. In particular, digital approaches achieve provable security by requiring consistent user and/or recording-device cooperation to bind pixel values to signatures. Spotlight and physical signatures more broadly trade provable security for flexibility, scalability, and a shift of protection agency (and efforts) from audience to speaker.

3.3 Design Challenges

Compact and pose-invariant video digest The video digest must be extracted in real-time and capture the speaker’s identity and facial motion to protect against their falsification. Furthermore, it must be pose-invariant, so that any recording can be verified. Unfortunately, the optical channel offers limited bandwidth, constraining the digest size. This is because a receiver’s sampling rate must be at least double the modulation frequency (i.e., the Nyquist rate) for data to be decodable. The standard frame rate for recording of live events is only 30 frames per second (FPS) [85]. Thus, achieving embedding bandwidths of even hundreds of bits per second (bps) is challenging. Existing speaker analysis methods output large representations (e.g., hundred-dimension embeddings for single images [86] or seconds of audio [87]), well-exceeding this limit.

Signature embedding and extraction Optical signature embedding and extraction contend with a tradeoff between robustness and imperceptibility. While large optical modulations bolster robustness by increasing the signal to noise ratio (SNR) of embedded signals, such fluctuations are highly perceptible at the scene and in recordings. Our modulation and extraction techniques must be both minimally obtrusive and robust to noise that may be introduced during capture and post-processing. Unlike in the digital realm where pixels can be directly modified to covertly and reliably embed information, we must *anticipate* how light *injected* into the scene will induce varying degrees of perceptibility and robustness.

We next present the design of Spotlight’s three modules, which jointly address the above challenges.

4 Speech Video Signature Creation

We propose a feature-agnostic framework for creating compact, pose-invariant speech video signatures in real-time (Figure 4a). We build off existing computer vision tools to extract semantically-meaningful visual features crafted to address the attacks outlined in §2.1. Then, we use a technique known as locality-sensitive hashing (LSH) to compress these high-dimensional features to hundreds of bits (within the embedding data capacity) while preserving their verification functionality. Unlike cryptographic hash functions, which output highly different hashes for inputs with minor differences, LSH maps similar inputs to similar hashes. This enables Spotlight to validate legitimate videos despite minor feature differences inevitably arising from recording condition and feature extraction

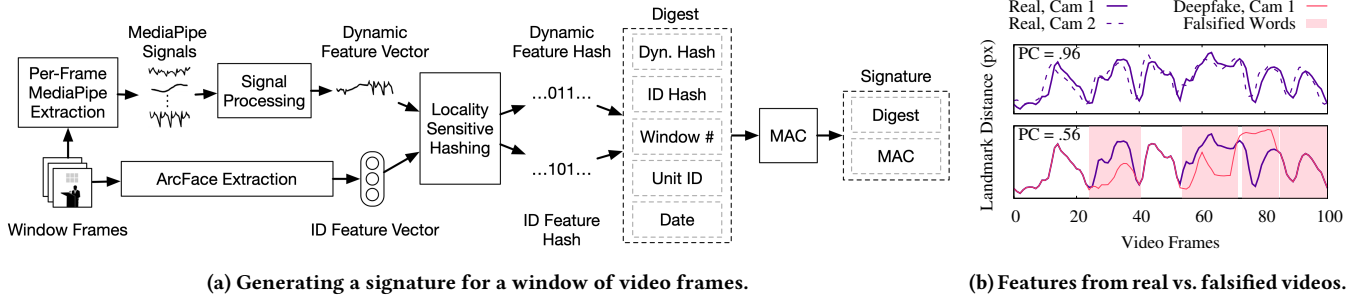


Figure 4: Assembly of Spotlight signatures. (a) Feature vectors hashed via LSH underpin the compact, cryptographically-secured signature. (b) A FaceMesh signal from three speech videos: real videos from two cameras at different yaws, and a reenactment deepfake changing four speaker words. The Pearson correlation (PC) between videos’ signals reflects the similarity of their speech content.

variance. While LSH is used in many domains, to our knowledge we are the first to use LSH-based features for verification.

4.1 LSH-Based Digest Framework

LSH is a technique for reducing data dimensionality while preserving approximate distances between data points. An LSH scheme consists of a function $H : \mathbb{R}^n \mapsto \{0, 1\}^k$ such that $D(H(\vec{u}), H(\vec{v}))$ estimates $\text{sim}(\vec{u}, \vec{v})$. sim is a similarity metric defined for \vec{u}, \vec{v} , D is the Hamming distance, and k is a configurable hash size. An LSH scheme does not preserve the exact similarity of inputs but rather provides a probabilistic guarantee that similar inputs are mapped to similar hashes; using a larger hash size k increases this probability.

We use the cosine similarity LSH scheme [88], denoted H_{\cos} . H_{\cos} outputs k -bit hashes such that $D(H_{\cos}(\vec{u}), H_{\cos}(\vec{v}))$ estimates $\Theta(\vec{u}, \vec{v})$. Here, $\Theta(\vec{u}, \vec{v})$ is the angle between \vec{u} and \vec{v} .

The cosine similarity LSH scheme is appealing for our use case for two reasons. (1) While larger hash sizes always improve the hash’s accuracy in estimating cosine similarity, this relationship is *independent* of the dimensionality of input vectors. This differs from Principle Component Analysis, where information loss is proportional to dimensionality reduction. We derive an equation for the effect of hash size on hashed features’ performance (Theorem 2) to confirm this. In the context of our application, this means that our initial feature vectors can be arbitrarily high-dimensional, so long as they capture the similarity of speech videos via cosine similarity. (2) H_{\cos} is suitable for estimating the Pearson correlation, a popular measure of similarity in time series data. This is because the Pearson correlation of two time series is equivalent to their cosine similarity after zero-meaning. This property is key to computing our dynamic features, which must capture temporal speech characteristics.

Equipped with H_{\cos} , we can separately address the challenges of digest robustness and constraint size. Next, we describe our high-dimensional visual feature vectors, which are hashed to serve as verification data. We formally analyze this hashing method in A.5

4.2 Semantically-Meaningful Video Digests

To address falsifications of speaker identity, lip and face motion, we extract two visual feature vectors: a biometric-based *identity feature vector* and a temporal *dynamic feature vector*. The LSH framework, however, supports varied features, as discussed in §12.

Identity feature vector The identity feature vector is used to verify the speaker identity in a published video to protect against identity swap falsifications. Neural network face embedding models are the gold-standard for extracting visual identity information. They map face images to vectors in a high-dimensional embedding space, where distance corresponds to face similarity. Conveniently, state-of-the-art face embedding models utilize cosine similarity as their distance metric. We employ a pre-trained ArcFace [89] model [90], which outputs a 512-dimensional vector. We pass ArcFace crops of the face obtained from a pre-trained face detector [91].

Dynamic feature vector The dynamic feature vector protects against falsifications of delivered content by ensuring that a speaker’s face and lip motion have not been modified. We use MediaPipe FaceMesh [92], a model for real-time face image analysis, to distill a window of video frames into a signal capturing both coarse and fine-grained spatio-temporal visual characteristics of the speaker. As shown in Figure 4b, the similarity of two speech videos can be quantified as the Pearson correlation of their corresponding signals. We find that these simple signals strongly protect against varied falsifications (§9.1). They also are more compact yet comparable in robustness to other speech features, as discussed in §12.

Thus, given a window of n video frames, we run FaceMesh on each frame to obtain its 52 blendshape scores – pose-invariant coefficients representing facial expressions – and 478 facial landmarks which we align to a canonical view for pose-invariance. We find that FaceMesh produces accurate output for frames captured up to 60° off-axis from the speaker (§9.1). We concatenate the values of 11 blendshapes and 5 distances between landmarks around the lips into separate n -sample signals, which together capture global facial motion and nuanced lip motion. We identify this set of features as optimal via forward sequential feature selection [93] on all blendshape and distance signals, using a comprehensive multi-camera dataset (§9.1). Finally, we smooth and standard normalize all signals and concatenate them into our $16n$ -dimensional feature vector.

4.3 MAC Generation and Key Management

Our digest consists of both feature hashes, a window number, a core unit identifier, and a creation date (Figure 4a). The Spotlight core unit generates a HMAC-SHA1 MAC for each digest using its secret key, ensuring the integrity and authenticity of embedded data. We refer to a digest and its MAC as a speech video signature.

We secure our data via MACs as opposed to public key encryption because public key schemes produce ciphertexts exceeding our embedding bandwidth.¹ We discuss approaches to increasing bandwidth to facilitate public key encryption in §12.

To create and validate MACs, Spotlight must establish a secret key shared by core units and the verification service. This can be done using Diffie-Hellman key exchange [94], a scheme enabling two parties to generate a shared secret key over an insecure channel using their public keys. Spotlight may employ Diffie-Hellman in one of two ways. In the first, it may require each core unit owner to use his own public key to participate in key exchange with the cloud-based verification service (§3.1). During this process, Spotlight can authenticate the owner’s public key via a digital certificate [95]. This creates a unique secret key for each core unit, securely associated with a unit’s identifier and owner. In the second approach, Spotlight may maintain the same key across all core units, refreshing as needed. Our prototype assumes a secret key has already been initialized in one of these manners, since the involved key exchange and certificate technologies are well-established.

5 Optical Signature Embedding

After obtaining the signature for a window, the optical signature embedding module projects light encoding the signature data into the scene. Prior works most relevant to this task explore light-based [96, 97] or screen-camera [98–115] communication. They achieve imperceptibility at the scene while maximizing the visibility of modulated light in captured frames for *real-time* decoding. Such methods are inapplicable in our case, wherein we seek imperceptibility both live and in-video and decoding is performed downstream on videos rather than at capture. Further, several of these works require cameras to operate in rolling-shutter mode [96, 98].

To address these issues, we propose three design elements, illustrated in Figure 5. (1) We modulate light spatially and temporally to boost embedding bandwidth. The temporal modulation operates at low frequencies (e.g., 3–6 Hz) to accommodate commodity camera’s framerates and all shutter modes. We leverage an amplitude-modulating spatial light modulator (SLM) – an optical device that controls the intensity of emitted light in both space and time – to introduce small amounts of carefully-crafted light onto a planar surface in the immediate vicinity of the speaker. Our design maintains imperceptibility both live and in videos by exploiting the human visual system’s low sensitivity to small fluctuations in light intensity occurring in small regions [116] and for short durations [117]. (2) We apply concatenated error correction coding to the signature data to ensure its reliable recovery from videos and enhance its resilience against video post-processing techniques. (3) We design an adaptive embedding mechanism which continually tunes the emitted light to adapt to environmental changes and balance embedding imperceptibility and robustness.

5.1 Concatenated Error Correcting Code

Our concatenated error-correcting code [104] consists of two simpler codes: an outer Reed-Solomon (RS) code and an inner convolutional code. Raw signature data first goes through the RS coder,

¹For equivalent authenticity and integrity guarantees, RSA-based digital signatures are over 10 times larger than HMAC-SHA1 MACs.

which adds $n - k$ parity bytes to the k -byte signature to form an n -byte codeword. RS can correct errors in up to $\lfloor (n - k)/2 \rfloor$ bytes in a codeword. The codeword is then passed through a convolutional coder, yielding our final coded data. To later recover the RS codeword, we perform soft decision Viterbi decoding. The soft decoder takes each bit’s distance to 1 or 0 to compute the corrected sequence. This aids in decoding signals that are consistently noisy. Unlike RS codes, convolutional code correction strength is dependent on error positions and can thus be unstable. Thus the two codes complement each other to greatly improve embedding robustness. The soft decoding corrects a majority of errors. The RS code guarantees to correct all remaining errors up to its correction strength.

5.2 Spatio-Temporal Light Modulation

Coded data is translated into a series of bitmaps, which are projected in sequence by an SLM equipped with red, green, and blue LEDs. The SLM accepts RGB bitmaps, where pixel values in a color channel are proportional to LED intensities. Thus, bitmap values determine the composition of light hitting surface regions, which in turn determines the regions’ pixel values in videos and appearance live.

We propose a spatially multiplexed modulation scheme. We divide each bitmap into a set of *cells* (blocks of pixels), as shown in Figure 5. Each cell is independently modulated in time to produce optical signals. In a given bitmap, a cell j can be either “on” or “off.” When off, its color is set to black (RGB(0, 0, 0)), corresponding to zero emitted light. When on, its color is set to $c_{SLM}^j = \text{RGB}(R_{SLM}^j, G_{SLM}^j, B_{SLM}^j)$, as determined in §5.3. We employ three types of cells, each with specific modulation behaviors.

(1) Data cells Most cells are data cells, modulated using Binary Phase Shift Keying (BPSK) to carry the raw bits of the signature. Each bit is communicated via the display of a cell in two consecutive bitmaps. A 0 is conveyed by a cell value of c_{SLM}^j followed by RGB(0, 0, 0) (i.e., $\phi = 0^\circ$), and vice versa for a 1. We employ BPSK for its balance of data rate and robustness to noise. Data cells are modulated at f_d Hz by displaying bitmaps at a refresh rate of $2f_d$ Hz. Thus, each carries $m * f_d$ bits for a m s modulation time. To embed b bits, we assign contiguous chunks of $m * f_d$ bits to each data cell.

(2) Synchronization cells Synchronization cells make up the border of the bitmap and facilitate demodulation of data cells. They are consistently modulated at the data frequency f_d with a phase $\phi = 0$. This provides the necessary reference signal for BPSK-demodulation of the data cells. Synchronization cells are also used to determine the start and end of each window, as further described in §6. For an embedding window of n seconds, we modulate all types of cells for $m (< n)$ seconds and leave the remaining $n - m$ as a downtime to facilitate determining the start of each window.

(3) Localization cells The corner cells of a bitmap are localization cells. They are consistently modulated at a f_l Hz as a beacon of the optical signature’s presence and location in recordings, critical for verification. Their larger size and distinctive frequency distinguish them from other cells, supporting downstream localization (§6).

5.3 Adaptive Embedding

The key idea of adaptive embedding is to set each bitmap cell j ’s color to optimize the illumination of its corresponding patch j on the projection surface. While a cell’s SNR is strictly determined by

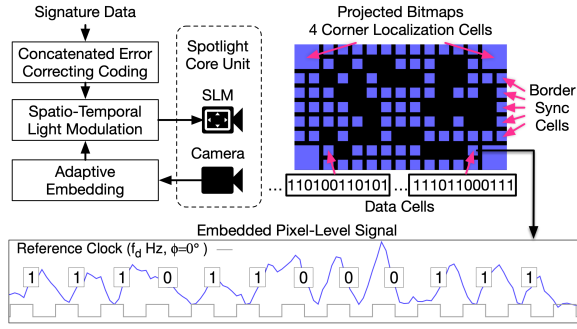


Figure 5: (Top) Overview of optical signature embedding, alongside a bitmap used by our modulation scheme. (Bottom) Resulting signal in a video, belonging to the last data cell. The reference clock, equivalent to our synchronization signal, is used to illustrate our BPSK scheme.

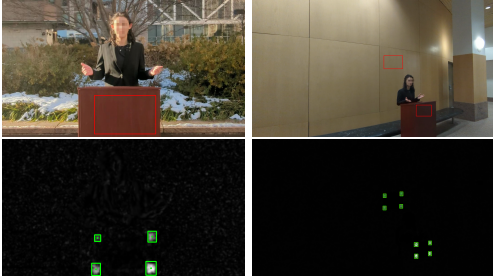


Figure 6: Embedded signatures are localized in any recording – including those where signatures are replicated at the scene (top right) – by condensing the video into a heatmap highlighting localization cells (bottom). Projection regions are outlined in red for visualization.

emitted light intensity (i.e., the sum of RGB channel emissions), its perceptibility is also influenced by its color (i.e., the light’s relative channel values). For a given intensity, light is most imperceptible when its color matches that of a patch in the absence of SLM light.

Thus, we propose an intensity-guided adaptive embedding method, which continually adapts the cell intensities I^j required for sufficient SNR and then optimizes cell colors c_{SLM}^j under this constraint. Prior to Spotlight’s deployment, we perform a short, one-time calibration enabling the core unit to map SLM bitmap pixels to patches viewed in its camera. Upon completion of a window, Spotlight runs the adaptation algorithm (detailed in A.1) in parallel with ongoing modulations. The algorithm assesses robustness and perceptibility based on the completed windows’ core unit video frames, and accordingly increments or decrements I^j . To quantify robustness, the core unit runs data extraction on its past window’s video (§6) and computes the data error rate. The perceptibility of each cell j is measured as the perceived difference between patch j ’s color with and without SLM light, using the popular color difference formula CIEDE2000 [118]. Then, c_{SLM}^j is chosen to minimize perceptibility while satisfying the intensity requirement I_j , via Equation 1.

6 Video Integrity Verification

The last Spotlight module verifies video integrity. It localizes optical signals in videos without prior knowledge of the projection

surface, robustly recovers low-SNR data, and accurately assesses video integrity regardless of recording parameters.

Optical signature localization To locate imperceptible optical signatures in a video, we leverage known properties of localization cells to create a *heatmap* in which they light up in any scene (Figure 6). Since localization cells are modulated at f_l (§5.2), the pixel values at these cells also exhibit oscillations at f_l . Thus, we apply a Fourier transform to each pixel-level signal in the video. We use a subset (e.g., 800) of the frames for this step for efficiency. We record each pixel’s power at f_l and normalize it by the noise at other frequencies. This yields our heatmap, where a pixel’s brightness is proportional to its normalized power at f_l . We then detect the localization cells in the heatmap via contour detection [119]. If fewer than four contours are detected, Spotlight reports a verification failure.

Next Spotlight determines the mapping between pixels in SLM bitmaps and the published video, allowing it to examine embedded cell signals. Any camera’s view of the projected bitmaps is related to the bitmap itself via a homography [120]. We compute the homography using the localization cells as correspondences and apply it to all frames to map video pixels directly to SLM pixels.

Signature data extraction Having obtained the homography mapping cells to video pixels, Spotlight extracts a signal for each cell by taking its average pixel intensity across frames. Building off the scheme described in §5.2, data is recovered from these signals as follows. First, Spotlight determines the start and end of all embedding windows by finding periods of downtime in the synchronization cell signals. Second, it smooths and detrends data cell signals to remove noise and impacts of gradual intensity shifts often induced by camera auto-white-balance and auto-exposure. It then demodulates these signals per-window and passes its predictions to the concatenated error corrector to recover the signature data.

MAC validation Spotlight extracts the digests from all signatures and validates them against their MACs. A MAC mismatch suggests a window’s embedded data has been corrupted by attacker tampering or decoding errors. Spotlight reports windows with corrupt digests as untrustworthy, as their speech content cannot be verified.

Digest comparison A video’s integrity is determined by comparing recovered digests to those computed on portrayed content. For each window i , Spotlight downsamples the video to the core unit framerate and extracts its identity and dynamic feature hashes (§4). Recovered window numbers are ensured to be consecutive, and hashes are compared to their counterparts in the digest recovered from window $i + 1$.² Both the identity feature and dynamic feature hashes are compared via Hamming distance. The identity feature hash is computed for *all frames* to ensure its consistency throughout the video. An integrity decision is made using the maximum identity and dynamic hash distances across windows. If either exceeds a configured decision threshold, the video is deemed falsified.

7 Prototype Implementation

We fabricate a Spotlight prototype using readily-available, affordable components. The core unit (Figure 2) consists of an SLM [121] (\$299), a conventional RGB camera [122] running at 24 FPS, and

²For the final n seconds of a speech video to be verifiable, it must be followed by a window of embedding only. Our implementation configures n to be 4.5 seconds; increasing the embedding bandwidth can allow shorter windows (§12).

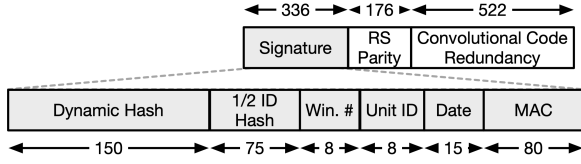


Figure 7: Data embedded per window in our implementation. The signature includes the digest and its MAC.

a Raspberry Pi 4B [123] for controlling the SLM. Code for real-time video signature creation and adaptive embedding runs on a MacBook Pro 14, using CPU only. Our window duration is set to 4.5 s, based on the 232 bps embedding bandwidth achieved under our modulation parameters (A.2). Our code takes 0.5 s to create a signature and send it to the SLM for modulation in a window’s remaining 4 s. Deployment and verification are automated in single scripts, minimizing required operational expertise. Verification is run on the MacBook (offline) for our proof-of-concept. We leave its integration into a cloud service to future work. The average time to verify a 30 s video is 86 s, which can be significantly reduced by using GPU and parallelizing per-frame analysis. We empirically set our digest comparison decision thresholds based on data from §8.

Finally, Figure 7 shows our signature aparameters. We choose hash sizes of 150 bits based on Theorem 2 and confirmed empirically (§9.1). We distribute each identity feature hash over two windows’ digests, such that the recovered identity feature hash is updated every other window during verification. This leaves space for more RS parity while preserving Spotlight functionality, as the speaker’s identity will not change within two windows. We use a 128-bit secret key to compute HMAC-SHA1 MACs, which we truncate to 80 bits to minimize MAC size while maintaining security [124].

8 Protection Performance Evaluation

We evaluate Spotlight’s performance in detecting falsifications of speaker identity, lip or face motion, via deepfakes or basic editing. All studies were approved by our Institutional Review Board.

8.1 Falsification with Deepfake Models

To demonstrate Spotlight’s protection performance across speakers and deepfake models, we first collect a large-scale video dataset of speeches delivered with our core unit present. We then generate extensive identity swap and reenactment deepfakes, and examine our verification module’s ability to differentiate real and fake videos. **Data collection** We collect our own dataset because although there are many general-purpose public deepfake datasets, videos in these datasets were necessarily not collected with Spotlight deployed. Additionally, existing deepfake datasets do not provide *pairs* of real and falsified videos, which would be needed to emulate Spotlight’s comparisons of recovered digests and portrayed content.

To collect authentic videos, we invited 20 participants (11 male and 9 female, ages 18 to 54) to read aloud six paragraphs (roughly 33 s each) while our core unit was deployed. Paragraphs were sourced from the Presidential Deepfakes Dataset (PDD) [126] and displayed on a monitor. Participants were recruited through flyers and emails within our institution and each compensated \$10. The core unit was positioned 1.5 m away from the participant and 2 m away from a white wall. We utilized a 100 x 70 cm portion of the wall

for projection, thoroughly investigating other projection surfaces and recording conditions in §9. We simultaneously recorded on four cameras: a Google Pixel 6A, an iPhone 14 in ProRes mode, a webcam [127], and a DSLR camera [128] positioned around the core unit. We synthesized 257 min of content across 474 videos.

Deepfake generation For each original video, we use FSGAN [129] to generate an identity swap deepfake where the speaker face is supplanted with that of a randomly selected alternative identity. We generate reenactment deepfakes using four state-of-the-art models: DaGAN [130], First Order Motion Model (FOMM) [131], TalkLip [132] and SadTalker [133]. The reenactment deepfakes modify the speaker’s facial movements to reflect their delivery of a different speech from PDD. While TalkLip exclusively modifies the lip region based on driving audio, FOMM, DaGAN, and SadTalker include face images as driving input to also modify expression. These varied reenactment scopes test the coverage of our dynamic features.

For all deepfakes, we only modify the facial region, and leave the rest of the scene (including the projection surface) as-is. This emulates a realistic attack scenario in which an attacker creates a convincing falsified video by maintaining the video’s context while changing speech content. We generate 1,883 reenactment deepfakes (753 min) and 473 identity swap deepfakes (261 min) total.

Metrics We input all videos to the verification module and record the Hamming distances between computed and recovered dynamic and identity hashes, as well as the module’s final decision on video integrity. We quantify performance using recall (i.e., true positive rate) and Area Under Curve (AUC), standard metrics for evaluating binary classifiers. An AUC of 1 indicates perfect separation of positive and negative class scores. We use feature hash distances as our scores. Since reenactment deepfakes maintain identity but change content, we compute the AUC for reenactment detection using only dynamic feature hash distances. Similarly, we use identity feature hash distances for identity swaps. The recall is based on Spotlight’s final binary decisions and thus considers both distances.

Passive detector comparison We compare Spotlight performance to that of 11 state-of-the-art passive detection models (Table 1), spanning a range of whole-video and frame-level methods. The goal of these comparisons is to ensure our created dataset is sufficiently challenging to fairly evaluate Spotlight. We choose these models as they are top performers in the comprehensive DeepfakeBench benchmark [134]. Specifically, each ranks within the top-3 methods for at least three datasets assessed, indicating effective cross-domain performance. We use implementations provided via DeepfakeBench for all models. Each was trained on the FaceForensics++ dataset [60], which contains both identity swap and reenactment deepfakes. We do not fine-tune the models on our data, since Spotlight requires no deepfake-specific fine-tuning. Further, in practice, passive detectors do not have *a priori* knowledge of the origins of their input.

Overall results As shown in Table 1, Spotlight achieves AUCs above 0.99 for all deepfake models and outperforms passive detectors by 37% on average. Spotlight has a recall of 100%, indicating it detects every one of the over 2,000 fake videos in our dataset. It additionally exhibits generalizability and explainability.

Among all reenactment deepfakes, only one possessed an identity hash distance above the decision threshold. Among all identity swap deepfakes, only 23 possessed an anomalous dynamic hash

Deepfake Model	Passive Detector										Ours
	Meso4 [57]	Xception [60]	Capsule [49]	Efficient [125]	SRM [50]	SPSL [51]	Recce [52]	UCF [53]	TALL [54]	AltFreeze. [55]	
DaGAN (R)	0.52	0.73	0.59	0.67	0.70	0.68	0.68	0.72	0.61	0.50	0.99
SadTalker (R)	0.62	0.72	0.71	0.68	0.70	0.70	0.66	0.71	0.62	0.45	0.99
FOMM (R)	0.64	0.83	0.73	0.80	0.79	0.74	0.77	0.80	0.69	0.43	0.99
TalkLip (R)	0.83	0.95	0.86	0.95	0.96	0.93	0.94	0.92	0.80	0.34	0.99
FSGAN (F)	0.66	0.88	0.69	0.79	0.84	0.88	0.83	0.83	0.79	0.57	1.00

Table 1: AUC scores achieved by Spotlight and eleven state-of-the-art passive detectors on our end-to-end video dataset. Our dataset includes both reenactment (R) and faceswap (F) deepfakes. Best performing method is bolded.

distance. Thus, our identity and dynamic features effectively isolate identity and motion-specific video elements, respectively. As a result, Spotlight can report the *type* of video falsification detected.

We also see that Spotlight’s performance generalizes across deepfake models. We attribute this to our digest’s focus on higher-level, semantically-meaningful visual features (e.g., temporal lip movement patterns, facial characteristics). These features are guaranteed to differ across content (deepfake-generated or not), unlike the low-level deepfake model-specific artifacts sought by passive detectors.

While the passive detector failures cannot be diagnosed, as they stem from black-box neural networks, Spotlight’s results are quite explainable. Its inaccuracies are overwhelmingly false positives (real videos labeled fake) triggered by high dynamic hash distances. These cases are caused by sporadic FaceMesh inaccuracies, which degrade the Pearson correlation between dynamic feature vectors. We consider alternative features not relying on FaceMesh in §12.

Signature extraction failures Out of 2,400 inputted videos, Spotlight could not localize the signature in 36 (6 original, and their 30 deepfake counterparts), all corresponding to one participant’s session. While a fraction of videos that passed localization had corrupted extracted signatures, each had sufficient intact signatures to enable a final verification decision. All such failures can be resolved by configuring the tradeoff between SNR and imperceptibility (§5.3).

8.2 Other Falsification Attacks

Beyond the above deepfake falsifications, Spotlight inherently addresses other common falsification techniques and attacks.

Speech speed modification Attacks modifying only the playback speed of a video [5, 10] are achieved by either changing a video’s framerate or duplicating/dropping frames to change the effective content speed. Both approaches alter the structure and frequency of embedded signals (e.g., halving their frequency to achieve a 0.5x slow-down), triggering localization and demodulation failures. A knowledgeable attacker may preserve the playback speed of only the optical signature region; however, this will desynchronize signatures and speech content, causing a conflict of dynamic features.

Video clipping and splicing Removing portions of video or splicing together clips (e.g., to re-order the speaker’s words) changes the progression of the speakers’ face movements, captured by dynamic features. Spotlight thus prevents such edits. Non-consecutive window numbers would expose clever re-ordering of intact windows.

Signature injection or manipulation An attacker may try to embed a signature that complements her modified or synthetic content by digitally injecting or manipulating pixel signals. However,

without Spotlight’s secret key, she fundamentally cannot. Copying other videos’ signatures fails as they are highly event-specific.

9 Protection Robustness Evaluation

The previous section evaluated Spotlight’s protection performance under a single recording and attack configuration. We now delve into its performance across a broad range of practical attack and recording conditions. We separately evaluate each condition’s impact on digest and optical signature embedding performance.

Summary of results Spotlight’s digest extraction and optical embedding modules both support recording at up to 60° off axis and 3 m away from the speaker and projection surface, with supported range further extended when optical zoom is employed. Digests enable detection of content falsifications as fine-grained as ≤ 1.35 s of a window and generalize across hundreds of evaluated identities. Finally, digests and optical signatures are resilient to varied post-processing techniques, including compression and filter application.

9.1 Digest Robustness

We evaluate our digest identity and dynamic features in terms of their pose-invariance and generalization across speakers. We explore our dynamic features’ sensitivity to fine-grained reenactment deepfakes, wherein an attacker modifies only a portion of a window. **Multi-pose and fine-grained deepfake datasets** To evaluate our identity features across poses and subjects at a large scale, we turn to the Labeled Faces in the Wild dataset (LFW) [135]. LFW images are captured at extensive angles and distances in unconstrained environments. Our experiments span all 1,680 LFW individuals.

To evaluate our dynamic features’ pose invariance and sensitivity to fine-grained modifications, we build a dataset of speech videos simultaneously captured from extensive angles and distances. We then generate pinpointed reenactment deepfakes of these videos.

Specifically, we construct a multi-camera rig of six synchronized 1080p webcams positioned across two distances (1.5 m and 3 m) and three angles (0°, 45°, and 60° from the speaker³). We record nine participants as they read aloud four paragraphs (roughly 15-30 s each) sourced from the popular acoustic-phonetic corpus TIMIT [136].

For each of these 864 real videos, we create a suite of deepfakes by falsifying individual words in the paragraphs. We replace each targeted word’s *exact* portion of video with a reenactment deepfake of the same duration portraying the speaker uttering a different word from TIMIT. A SadTalker, FOMM, DaGAN, and TalkLip version is

³We assume performance is symmetrical about the speaker face and thus do not mirror the configuration at angles $< 0^\circ$.

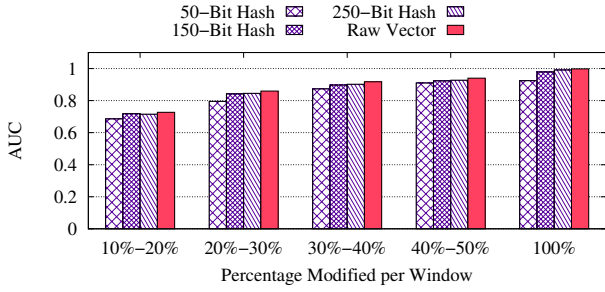


Figure 8: Window-level AUC scores achieved by Spotlight’s dynamic features across modification granularities and hash sizes.

created for each case. We arrive at 11,250 videos (10,368 deepfakes) of diverse camera positions and falsification granularities.

Metrics We report AUC scores for verification of all LFW subjects using identity features. For dynamic features, we report per-window AUC scores under various percentages of modified content, by duration. In computing these AUCs, negative class scores are distances between dynamic features extracted from windows of the same scene, shot from different camera positions. Positive scores are distances between real video windows and their fake counterparts.

Sensitivity to fine-grained modifications Figure 8 shows our dynamic feature AUCs across modification granularities, while Table 2 summarizes the AUCs of passive detectors. Spotlight’s 150-bit dynamic hashes score an AUC of .98 for fully-falsified windows and AUCs ≥ 0.90 for modification percentages ≥ 30 . This is a 40% gain over the best performing passive detector on the 30-40% bin.

We observe that AUC drops with decreasing modification percentage. Windows with minor modifications may exhibit dynamic feature signals dominated by the similarities between remaining clean content, causing false negatives. For the 10-20% bin, Spotlight AUC drops to 0.72. Notably, this modification rate corresponds to a highly specific attack, in which as little as 0.45 s of words within one 4.5 s window are *precisely* supplanted. If any introduced reenactments are even a few frames longer or shorter than content they are replacing, all subsequent frames in the video are shifted; this has the effect of modifying 100% of content in subsequent windows.

Ultimately, detection capability is inevitably dependent on the degree of modification. Indeed, existing passive detectors struggle to temporally localize finer-grained falsifications [137]. Spotlight maintains reasonable performance and inherently localizes falsifications on the resolution of windows. We explore the potential of other dynamic features to counter subtle falsifications in §12.

Generalization across identities Our identity hashes score an AUC of 0.99 in differentiating all 1,680 LFW subjects. Dynamic feature hashes are similarly robust across speakers, with errors in detecting reenactment deepfakes distributed evenly across subjects.

Hash size The AUC drop-off between 150- and 50-bit dynamic feature hash sizes (Figure 8) validates our choice of 150-bit hashes. Hashing ArcFace vectors to 150 bits lowers their AUC by just 0.0016.

Recording distance and angle Digest performance is consistent across recording positions up to 3 m and 60° off-axis from the speaker. Amongst videos captured at the harsh 3 m, 60° position, dynamic feature hashes have an AUC of 0.96 in detecting fully



Figure 9: Environments and respective surfaces (S1-7) tested in §9.2. Insets show close-ups of the projection surfaces, varying in their textures and coloration. Surfaces 5, 6, and 7 are outdoors.

falsified windows. Note that we record all dataset video with no zoom for consistency; using zoom naturally boosts supported range.

Video post-processing We do not observe significant changes in digest performance upon compression, transcoding, or filtering, likely due to the diversity of ArcFace and FaceMesh training data.

9.2 Signature Embedding Robustness

We evaluate our optical signature embedding scheme’s robustness with respect to Spotlight’s ability to ultimately extract embedded data from videos. We consider extensive practical factors, from ambient lighting and projection surfaces to video post-processing.

Experimental setup To assess the effects of projection surface and ambient lighting, we project onto seven surfaces (Figure 9), including three outdoors under dynamic cloud coverage. We evaluate two lighting conditions for S1-3, for a total of 10 environments. To assess the remaining factors, we project onto S1. For each scenario, we embed a random bitstream for 100 s and extract the data from the recording. By default, we record with the core unit setup detailed in §8 and a Google Pixel at 2 m from the projection surface. We quantify robustness in terms of the bit error rate (BER) at each stage of decoding: raw, post-Viterbi, and final, upon full error correction.

Recording distance and angle Spotlight’s embedding supports recording up to 3.5 m and 60° from the projection surface (Table 3a). We find embedding robustness is primarily constrained not by camera distance, but rather the resolution of cells in recordings, as this determines their SNR. The BER increase at 5 m is fundamentally due to inadequate cell resolution. Table 3f shows final BER is zero so long as cells occupy $\geq 35 \times 35$ pixels. The full projection region corresponding to this cell resolution is just 16% of a 1080p frame.

Recording environment and projection surface Spotlight achieves error-free embedding in all evaluated scenes (Table 3b), including dynamic outdoor environments and surfaces ranging from red brick (S6) to irregularly patterned, glossy paper (S3). We attribute this to Spotlight’s adaptive embedding procedure, which continually ensures sufficient SNR. We find Spotlight achieves zero final BER in any environment where ambient light intensity does not dominate SLM-projected light. We measure this threshold value to be 4 klx (roughly 20x brighter than a typical indoor setting [138]).

Recording device Because Spotlight encodes data as simple light intensity changes, it is naturally compatible with any modern RGB camera. Signatures are reliably extracted from videos captured on all five tested $\geq 1080p$ devices, from webcams to DSLRs (Table 3c).

Video post-processing Signatures remain decodable after varied forms of video post-processing, including compression (done by reducing bitrate from the original 19k kbps), transcoding from the













Target Motion (Attacker)	Reenactment Model Outputs				Real Video (Victim)
	$\alpha = 0$	$\alpha = 10$	$\alpha = 20$	$\alpha = 40$	
					
					
Verification Success Rate	0%	7%	13%	35%	

Figure 10: Example DaGAN outputs after training with various weights α on the adversarial Spotlight-spoofing objective. Higher weights increase generated videos’ verification success rates, but cause the model to reproduce the semantic content of the real video.

original H.265 codec (Table 3d), contrast and exposure changes, and use of monochrome and auto-enhancing [139] filters (Table 3e).

10 Countermeasures

We explore countermeasures an attacker may employ in an attempt to create falsified content that nonetheless passes verification, i.e., spoofs, or otherwise disrupt Spotlight operation. We assume the attacker has white-box access to all Spotlight algorithms and models.

10.1 Spoofing via Adversarial Examples

Extensive prior works show that deep neural networks (DNNs) are vulnerable to adversarial examples – carefully crafted inputs that look normal to the naked eye but cause models to make incorrect predictions [140]. An attacker may try to craft adversarial examples against the FaceMesh and ArcFace DNNs that Spotlight uses to extract feature vectors (§4.2). Concretely, her goal is to take a Spotlight-protected video and create a falsified version that, although to the naked eye portrays a different identity or facial motion, elicits identity and dynamic features highly similar to those of the real video. If she achieves this, her fake video’s feature vectors will possess locality-sensitive hashes similar to those in the original video signature. She can thus simply retain this signature in her fake video, and it will pass verification, *spoofing* Spotlight.

We demonstrate two approaches to creating such adversarial examples – one in which the attacker specially trains her deepfake model and the other in which she perturbs her videos post-factum – and find neither succeeds. While we are under no illusion this means creating adversarial examples against Spotlight is fundamentally impossible, we show it is in practice highly challenging. This significantly raises the bar for attack execution.

Below, we use the VoxCeleb video dataset [141] for training and tests. Note that the LFW dataset, while favorable for our evaluations in §9.1 due to its greater identity and pose diversity, cannot be used in the following video-focused studies because it is image-based.

Adversarial deepfake generation In the first method, with white-box access to the FaceMesh and ArcFace models, the attacker directly incorporates the above signature spoofing objective into the loss function of her deepfake model. Specifically, during training, she can extract the Spotlight feature vectors from generated videos and apply a penalty if those feature vectors are dissimilar from the original video feature vectors. If our digests are vulnerable to

adversarial examples, the deepfake model should learn to satisfy the attack objective while still achieving the intended falsification. Note that adversarial examples must meet *both* of these criteria. If those generated videos that pass verification simply resemble the real videos, Spotlight is providing the expected protections.

We test this method using the FSGAN identity swap model and DaGAN reenactment model. We choose DaGAN as our representative reenactment model because its outputs were the hardest for passive detectors to detect (Table 1). We modify the DaGAN and FSGAN loss functions to include a *Spotlight-spoofing term*, which applies the aforementioned penalty based on generated videos’ dynamic (DaGAN) or identity (FSGAN) feature vectors. We train both models with three different weights on this term (empirically set to optimize attack success), and then use each version to generate 50 fake videos. We report the rate at which each model’s generated videos pass Spotlight verification. Details can be found in A.3.

We find that none of our attack models can produce adversarial examples. When trained with sufficiently high weight on the Spotlight-spoofing term, DaGAN learns to output content that passes Spotlight verification by simply retaining the face and lip motion of the real video. Figure 10 illustrates this effect. We can see that while the original DaGAN model (spoofing weight $\alpha = 0$) modifies the victim’s facial movement according to the attacker-provided target, the outputs of the adversarial model with the highest success rate ($\alpha = 40$) largely portray the same expressions as the real video frames, with some perceptual degradations. The FSGAN spoofing rate remained at 0% across all tested weights on our term.

These behaviors arise from a clear contradiction between the Spotlight-spoofing and original deepfake loss function components: while the former enforces similarity between the real and fake videos’ semantic visual aspects, the latter explicitly rewards real videos’ modification. During training, we observe one component strictly dominates the other, depending on their relative weights. Even after extensive testing, we cannot find a weight at which both components simultaneously converge. Thus Spotlight’s digests are adversarially robust; even when deepfake models can directly back-propagate through the feature extractors in training, they cannot find loopholes enabling generation of adversarial examples. **Adversarial perturbation of frames** The attacker may also add adversarial perturbations to video frames as a post-processing step, inspired by other perturbation-based attacks on vision DNNs [140].

We first apply this method to our identity features by adapting Fawkes [142], a white-box attack on face recognition models. Given a structural dis-similarity (DSSIM) [143] budget configuring permitted perturbation visibility, Fawkes perturbs a *source* face image to shift its feature space representation towards that of a desired *target* identity. We replace the Fawkes feature extractor with our identity feature extractor, consisting of the face detector and ArcFace model.

We then randomly choose 22 source-target pairs of identities from VoxCeleb and use Fawkes to perturb all source video frames toward their targets. Because frames must be perturbed independently, this attack is highly computationally expensive (over 1 min per frame even on GPU). We report the rate at which perturbed source videos are successfully verified as the target identity. We perform this experiment under four DSSIM budgets: 0.003, 0.005, 0.007, and 0.009. A larger budget enables larger feature space shifts – increasingly the chances of verification success – but introduces

visible artifacts exposing the attack. Prior studies perturbing face images use DSSIM budgets from 0.003 to 0.007 [142, 144] to maintain imperceptibility. Note, however, recent work [145] suggests perturbations at these budgets may become visible in *videos* due to temporal incoherence across independently-perturbed frames.

We observe a verification success rate of zero for all budgets ≤ 0.007 . Though Fawkes successfully perturbs a larger portion of frames per video with larger budgets, it fails to succeed on *all* frames. This is necessary for the video to pass verification, as Spotlight validates all frames’ identity features (§6). At the highest budget of 0.009, the success rate rises to 4.5%; however, the frames exhibit noticeable artifacts, causing a distinct flickering effect when played. Thus, even when allowed a budget exceeding prior perceptibility thresholds, the perturbation-based attack on identity features fails.

These results can be attributed to Spotlight’s particularly stringent identity feature verification threshold (configured in §7), which forces perturbations to shift features by larger amounts to produce a spoof. Spotlight’s identity feature verification threshold is particularly strict because videos recorded at a speech site necessarily vary only in their viewpoint of the speaker, with other appearance variations that ArcFace is trained to accommodate (e.g., makeup, lighting) naturally constant. As a result, legitimate video identity features are generally highly similar to those disseminated by the core unit. This phenomenon is also reflected in Spotlight’s perfect AUC in detecting identity swap deepfakes in §8 (Table 1).

Beyond this empirical validation, several works show that with sufficient perturbed and unperturbed images of a subject, a defender can train a highly accurate adversarial perturbation detector (AUC $> .997$) that generalizes across perturbation methods [146, 147]. For high-profile speakers, images for such training are abundant. Thus, Spotlight can incorporate a detector to preemptively detect and reject perturbed videos. Given these findings, we leave exploration of perturbation-based attacks on dynamic features to future work.

10.2 Other Countermeasures

Screen recording An adversary may launch the following attack based on screen recording: she places the core unit in front of a screen displaying a fake video, records the outputted optical modulations, and then digitally overlays them on the fake video. This attack can be simply addressed by equipping the core unit with an existing depth-sensing tool [148] to differentiate a 2D screen from a speaker’s physical presence, which we leave to future work.

Environment-level interference An attacker present at a speech could interfere with Spotlight by injecting light onto its projection surface. For this to be effective, interfering light must dominate SLM illumination (i.e., measure roughly 4 klx, based on §9.2). Light of this intensity is quite visible; thus the attack can be detected and stopped at the scene. As further defenses, Spotlight can project onto multiple surfaces and periodically randomize projection surfaces.

11 Perceptibility Evaluation

We evaluate the perceptibility of Spotlight’s introduced optical modulations both live and in video via a user study and perceptual metrics. We summarize findings below, with further details in A.4.

For our user study, we invited participants to each scene in Figure 9. At each, they were asked to assess the projection surface

during trials in which the core unit either performed embedding or was powered off as a control case. For each trial, they reported whether they believed optical modulations were present, and rated the obtrusiveness of any perceived modulations. We repeated the study with *videos* of the surfaces. As shown in the bottom panels of Figure 11, participants overwhelmingly performed no better than random at detecting Spotlight operation, indicating the effective imperceptibility of its projected light. In the few cases users accurately detected operation, they uniformly reported low obtrusiveness.

We also analyze videos using the learned perceptual loss (LPIPS) [149] metric. All videos possess LPIPS scores over ten times lower than the established LPIPS perceptibility threshold (Figure 11).

12 Discussion and Future Work

Alternative embedding mechanisms Spotlight is compatible with varied embedding methods. Its modulation scheme supports any cell shapes and layouts and can be realized via existing projectors or screens at events. Future work will explore acoustic methods as well as optical embedding on non-planar and non-stationary surfaces, with the ultimate goal of projecting onto the speaker face.

Alternative dynamic features Spotlight is compatible with diverse dynamic features. One alternative is the cryptographic hash of a window’s script, extracted via real-time speech-to-text. This would provide key semantic information and aid detection of fine-grained content falsifications (e.g., changing “do” to “do not”), though failing to capture speech speed or visual cues. Future work will pursue LSH-compatible features capturing additional audio characteristics (e.g., tone) and visual attributes. Ultimately, dynamic features should be chosen based on anticipated attacks, and LSH aids in including multiple, complementary features in a signature.

Verifying the last speech window To ensure all speech content is protected, a video must have a window of downtime at its end for embedding of the final signature. Minimizing window durations can mitigate this overhead. This can be achieved by increasing embedding bandwidth, in turn reducing the time needed to embed each signature. The current scheme’s bandwidth can be boosted by increasing projection region size or adjusting the imperceptibility-SNR trade-off. Developing acoustic embedding methods for joint use with optical modulation is a further promising direction.

Camera movement The current Spotlight implementation assumes recordings are taken on a still camera, a constraint imposed by the verification module’s assumption that embedded signals are carried by the same pixels throughout a video. Future efforts will integrate established video stabilization [150, 151] and inter-frame alignment [152] methods into verification to allow camera motion.

References

- [1] Phoebe Sengers. The History of Television. <https://www.cs.cornell.edu/~pjs54/Teaching/AutomaticLifestyle-S02/Projects/Vlku/history.html>.
- [2] Robert J. Donovan and Ray Scherer. *Unsilent Revolution: Television News and American Public Life, 1948-1991*. Woodrow Wilson International Center for Scholars; Cambridge University Press, 2005.
- [3] University of Virginia Miller Center. The Presidency in the Television Era. <https://millercenter.org/the-presidency/teacher-resources/recasting-presidential-history/presidency-television-era>.
- [4] S Shyam Sundar, Maria D Molina, and Eugene Cho. Seeing Is Believing: Is Video Modality More Powerful in Spreading Fake News via Online Messaging Apps? *Journal of Computer-Mediated Communication*, 26(6):301–319, 08 2021.
- [5] Doctored Nancy Pelosi video highlights threat of “deepfake” tech. CBS News, 2019.

- [6] AFP USA Natalie Wade. Deepfake of Bella Hadid misrepresents her statements on Israel. *AFP Fact Check*, 2023.
- [7] Deepfake John Swinney addresses Scotland's parliament. *AFP Fact Check*, 2024.
- [8] Fact Check: Video of Joe Biden calling for a military draft was created with AI. <https://www.reuters.com/fact-check/video-joe-biden-calling-military-draft-was-created-with-ai-2023-10-19/k>, 2023.
- [9] AFP USA Bill McCarthy. Video of Biden botching Ukraine history is a deepfake. <https://factcheck.afp.com/doc.afp.com.34LY8TK>, 2024.
- [10] Old Kamala Harris footage manipulated to slow her speech. <https://factcheck.afp.com/doc.afp.com.36KM82F>, 2024.
- [11] Deepfake Kamala Harris slurs her lines. <https://www.aiaaic.org/aiaaic-repository/ai-algorithmic-and-automation-incidents/deepfake-kamala-harris-slurs-her-lines>, 2024.
- [12] Trump shares heavily edited video that highlights verbal stumbles by Pelosi and questions her mental acuity. https://www.washingtonpost.com/politics/trump-shares-video-that-highlights-verbal-stumbles-by-pelosi-and-questions-her-mental-acuity/2019/05/24/a2e83ed8-7e0d-11e9-8ede-f4abf521ef17_story.html, 2019.
- [13] Sheng-Yu Wang, Oliver Wang, Richard Zhang, Andrew Owens, and Alexei A. Efros. CNN-Generated Images Are Surprisingly Easy to Spot... for Now. In *Proc. of CVPR*, pages 8692–8701, 2020.
- [14] Shruti Agarwal, Hany Farid, Ohad Fried, and Maneesh Agrawala. Detecting deep-fake videos from phoneme-viseme mismatches. In *Proc. of CVPR Workshops*, pages 2814–2822, 2020.
- [15] 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.
- [16] Mingliang Chen, Xin Liao, and Min Wu. PulseEdit: Editing Physiological Signals in Facial Videos for Privacy Protection. *IEEE Transactions on Information Forensics and Security*, 17:457–471, 2022.
- [17] Nicholas Carlini and Hany Farid. Evading Deepfake-Image Detectors with White- and Black-Box Attacks. *arXiv preprint arXiv:2004.00622*, 2020.
- [18] SynthID. <https://deepmind.google/technologies/synthid/>.
- [19] Truepic. <https://truepicvision.com/>.
- [20] Yuxin (Myles) Liu, Zhihao Yao, Mingyi Chen, Ardan Amiri Sani, Sharad Agarwal, and Gene Tsudik. ProCam: A Camera Module with Self-Contained TCB for Producing Verifiable Videos. In *Proc. of MobiCom*, 2024.
- [21] Yuxin Liu, Yoshimichi Nakatsuka, Ardan Amiri Sani, Sharad Agarwal, and Gene Tsudik. Vronicle: verifiable provenance for videos from mobile devices. In *Proc. of MobiSys*, pages 196–208, 2022.
- [22] Partnership for greater trust in digital photography: Leica and Content Authenticity Initiative. <https://leica-camera.com/en-US/news/partnership-greater-trust-digital-photography-leica-and-content-authenticity-initiative>.
- [23] Content Authenticity Initiative. <https://contentauthenticity.org/>.
- [24] Yuezun Li, Ming-Ching Chang, and Siwei Lyu. In ictu oculi: Exposing ai created fake videos by detecting eye blinking. In *IEEE International Workshop on Information Forensics and Security*, pages 1–7, 2018.
- [25] Shu Hu, Yuezun Li, and Siwei Lyu. Exposing gan-generated faces using inconsistent corneal specular highlights. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 2500–2504, 2021.
- [26] Falko Matern, Christian Riess, and Marc Stamminger. Exploiting Visual Artifacts to Expose Deepfakes and Face Manipulations. In *IEEE Winter Applications of Computer Vision Workshops*, pages 83–92, 2019.
- [27] 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*, pages 1–1, 2020.
- [28] Hua Qi, Qing Guo, Felix Juefei-Xu, Xiaofei Xie, Lei Ma, Wei Feng, Yang Liu, and Jianjun Zhao. DeepRhythm: Exposing deepfakes with attentional visual heartbeat rhythms. In *ACM International Conference on Multimedia*, pages 4318–4327, 2020.
- [29] Steven Fernandes, Sunny Raj, Eddy Ortiz, Iustina Vintila, Margaret Salter, Gordana Urosevic, and Sumit Jha. Predicting heart rate variations of deepfake videos using neural ode. In *Proc. of CVPR Workshops*, pages 0–0, 2019.
- [30] Shruti Agarwal and Hany Farid. Detecting Deep-Fake Videos from Aural and Oral Dynamics. In *Proc. of CVPR Workshops*, pages 981–989, 2021.
- [31] Trisha Mittal, Uttaran Bhattacharya, Rohan Chandra, Aniket Bera, and Dinesh Manocha. Emotions don't lie: An audio-visual deepfake detection method using affective cues. In *ACM International Conference on Multimedia*.
- [32] Komal Chugh, Parul Gupta, Abhinav Dhall, and Ramanathan Subramanian. Not made for each other-audio-visual dissonance-based deepfake detection and localization. In *ACM International Conference on Multimedia*, pages 439–447, 2020.
- [33] Alexandros Haliassos, Konstantinos Vougioukas, Stavros Petridis, and Maja Pantic. Lips don't lie: A generalisable and robust approach to face forgery detection. In *Proc. of CVPR*, pages 5039–5049, 2021.
- [34] Yipin Zhou and Ser-Nam Lim. Joint audio-visual deepfake detection. In *Proc. of CVPR*, pages 14800–14809, 2021.
- [35] Pavel Korshunov and Sébastien Marcel. Speaker inconsistency detection in tampered video. In *European Signal Processing Conference*, pages 2375–2379, 2018.
- [36] Xin Yang, Yuezun Li, and Siwei Lyu. Exposing deep fakes using inconsistent head poses. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 8261–8265. IEEE, 2019.
- [37] Chen-Zhao Yang, Jun Ma, Shilin Wang, and Alan Wee-Chung Liew. Preventing DeepFake Attacks on Speaker Authentication by Dynamic Lip Movement Analysis. *IEEE Transactions on Information Forensics and Security*, 16:1841–1854, 2021.
- [38] Shruti Agarwal, Liwen Hu, Evonne Ng, Trevor Darrell, Hao Li, and Anna Rohrbach. Watch those words: Video falsification detection using word-conditioned facial motion. In *IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 4710–4719, 2023.
- [39] Shruti Agarwal, Hany Farid, Tarek El-Gaaly, and Ser-Nam Lim. Detecting Deep-Fake Videos from Appearance and Behavior. In *IEEE International Workshop on Information Forensics and Security*, pages 1–6, 2020.
- [40] Shruti Agarwal, Hany Farid, Yuming Gu, Mingming He, Koki Nagano, and Hao Li. Protecting World Leaders Against Deep Fakes. In *Proc. of CVPR Workshops*, page 38, 2019.
- [41] Davide Cozzolino, Andreas Rössler, Justus Thies, Matthias Nießner, and Luisa Verdoliva. Id-reveal: Identity-aware deepfake video detection. In *International Conference on Computer Vision*, pages 15108–15117, 2021.
- [42] Joel Frank, Thorsten Eisenhofer, Lea Schönherr, Asja Fischer, Dorothea Kolossa, and Thorsten Holz. Leveraging Frequency Analysis for Deep Fake Image Recognition. In *International Conference on Machine Learning*, 2020.
- [43] Xu Zhang, Svebor Karaman, and Shih-Fu Chang. Detecting and Simulating Artifacts in GAN Fake Images. In *IEEE International Workshop on Information Forensics and Security*, pages 1–6, 2019.
- [44] Yuezun Li and Siwei Lyu. Exposing DeepFake Videos By Detecting Face Warping Artifacts. In *Proc. of CVPR Workshops*, 2019.
- [45] Gaojian Wang, Qian Jiang, Xin Jin, and Xiaohui Cui. Ffr_fd: Effective and fast detection of deepfakes via feature point defects. *Information Sciences*, 596:472–488, 2022.
- [46] Lingzhi Li, Jianmin Bao, Ting Zhang, Hao Yang, Dong Chen, Fang Wen, and Baining Guo. Face X-Ray for More General Face Forgery Detection. *Proc. of CVPR*, pages 5000–5009, 2019.
- [47] Florian Lugstein, Simon Baier, Gregor Bachinger, and Andreas Uhl. PRNU-based deepfake detection. In *Proceedings of the 2021 ACM workshop on information hiding and multimedia security*, pages 7–12, 2021.
- [48] Marissa Koopman, Andrea Macarulla Rodríguez, and Zeno Geradts. Detection of Deepfake Video Manipulation. 08 2018.
- [49] Huy H Nguyen, Junichi Yamagishi, and Isao Echizen. Capsule-forensics: Using capsule networks to detect forged images and videos. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 2307–2311, 2019.
- [50] Yuchen Luo, Yong Zhang, Junchi Yan, and Wei Liu. Generalizing face forgery detection with high-frequency features. In *Proc. of CVPR*, pages 16317–16326, 2021.
- [51] Honggu Liu, Xiaodan Li, Wenbo Zhou, Yuefeng Chen, Yuan He, Hui Xue, Weiming Zhang, and Nenghai Yu. Spatial-phase shallow learning: rethinking face forgery detection in frequency domain. In *Proc. of CVPR*, pages 772–781, 2021.
- [52] Junyi Cao, Chao Ma, Taiping Yao, Shen Chen, Shouhong Ding, and Xiaokang Yang. End-to-end reconstruction-classification learning for face forgery detection. In *Proc. of CVPR*, pages 4113–4122, 2022.
- [53] Zhiyuan Yan, Yong Zhang, Yanbo Fan, and Baoyuan Wu. Ucf: Uncovering common features for generalizable deepfake detection. *arXiv preprint arXiv:2304.13949*, 2023.
- [54] Yuting Xu, Jian Liang, Gengyun Jia, Ziming Yang, Yanhao Zhang, and Ran He. Tall: Thumbnail layout for deepfake video detection. In *International Conference on Computer Vision*, pages 22658–22668, 2023.
- [55] Zhendong Wang, Jianmin Bao, Wengang Zhou, Weilun Wang, and Houqiang Li. Altfreezing for more general video face forgery detection. In *Proc. of CVPR*, pages 4129–4138, 2023.
- [56] Ning Yu, Larry S. Davis, and Mario Fritz. Attributing Fake Images to GANs: Learning and Analyzing GAN Fingerprints. *International Conference on Computer Vision*, pages 7555–7565, 2018.
- [57] Darius Afchar, Vincent Nozick, Junichi Yamagishi, and Isao Echizen. Mesonet: a compact facial video forgery detection network. In *IEEE International Workshop on Information Forensics and Security*, pages 1–7, 2018.
- [58] Peng Zhou, Xintong Han, Vlad I. Morariu, and Larry S. Davis. Two-Stream Neural Networks for Tampered Face Detection. In *Proc. of CVPR Workshops*, pages 1831–1839, 2017.
- [59] David Güera and Edward J. Delp. Deepfake Video Detection Using Recurrent Neural Networks. In *IEEE International Conference on Advanced Video and Signal Based Surveillance*, pages 1–6, 2018.
- [60] Andreas Rössler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Nießner. Faceforensics++: Learning to detect manipulated facial images. In *International Conference on Computer Vision*, pages 1–11, 2019.

- [61] Hanqing Zhao, Wenbo Zhou, Dongdong Chen, Tianyi Wei, Weiming Zhang, and Nenghai Yu. Multi-attentional Deepfake Detection. *Proc. of CVPR*, pages 2185–2194, 2021.
- [62] Yuyang Qian, Guojun Yin, Lu Sheng, Zixuan Chen, and Jing Shao. Thinking in Frequency: Face Forgery Detection by Mining Frequency-Aware Clues. In *European Conference on Computer Vision*, page 86–103, 2020.
- [63] Junke Wang, Zuxuan Wu, Wenhao Ouyang, Xintong Han, Jingjing Chen, Yungang Jiang, and Ser-Nam Li. M2TR: Multi-Modal Multi-Scale Transformers for Deepfake Detection. In *International Conference on Multimedia Retrieval*, page 615–623, 2022.
- [64] Hong-Shuo Chen, Mozdeh Rouhsedaghat, Hamza Ghani, Shuowen Hu, Suyu You, and C-C Jay Kuo. Defakehop: A light-weight high-performance deepfake detector. In *IEEE International conference on Multimedia and Expo*, pages 1–6, 2021.
- [65] Loc Trinh, Michael Tsang, Sirisha Rambhatla, and Yan Liu. Interpretable and trustworthy deepfake detection via dynamic prototypes. In *IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1973–1983, 2021.
- [66] Shivangi Aneja and Matthias Nießner. Generalized zero and few-shot transfer for facial forgery detection. *arXiv preprint arXiv:2006.11863*, 2020.
- [67] Nicolo Bonettini, Edoardo Daniele Cannas, Sara Mandelli, Luca Bondi, Paolo Bestagini, and Stefano Tubaro. Video face manipulation detection through ensemble of cnns. In *International Conference on Pattern Recognition*, pages 5012–5019, 2021.
- [68] Hao Dang, Feng Liu, Joel Stehouwer, Xiaoming Liu, and Anil K Jain. On the detection of digital face manipulation. In *Proc. of CVPR*, pages 5781–5790, 2020.
- [69] Mengnan Du, Shiva Pentiyala, Yuening Li, and Xia Hu. Towards generalizable deepfake detection with locality-aware autoencoder. In *ACM International Conference on Information & Knowledge Management*, pages 325–334, 2020.
- [70] Akash Kumar, Arnav Bhavsar, and Rajesh Verma. Detecting deepfakes with metric learning. In *International Workshop on Biometrics and Forensics*, pages 1–6, 2020.
- [71] Iacopo Masi, Aditya Killekar, Royston Marian Mascarenhas, Shenoy Pratik Gurudatt, and Wael AbdAlmageed. Two-branch recurrent network for isolating deepfakes in videos. In *European Conference on Computer Vision*, pages 667–684, 2020.
- [72] Paarth Neekhar, Shehzeen Hussain, Xinqiao Zhang, Ke Huang, Julian McAuley, and Farinaz Koushanfar. FaceSigns: Semi-Fragile Neural Watermarks for Media Authentication and Countering Deepfakes, 2022.
- [73] Adnan Alattar, Ravi Sharma, and John Scriven. A system for mitigating the problem of deepfake news videos using watermarking. *Electronic Imaging*, 32:1–10, 2020.
- [74] Amna Qureshi, David Megías, and Minoru Kuribayashi. Detecting deepfake videos using digital watermarking. In *2021 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pages 1786–1793, 2021.
- [75] Run Wang, Felix Juefei-Xu, Meng Luo, Yang Liu, and Lina Wang. Faketagger: Robust safeguards against deepfake dissemination via provenance tracking. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 3546–3555, 2021.
- [76] Yuankun Yang, Chenyue Liang, Hongyu He, Xiaoyu Cao, and Neil Zhenqiang Gong. Faceguard: Proactive deepfake detection. *arXiv preprint arXiv:2109.05673*, 2021.
- [77] Salma Masmoudi, Maha Charfeddine, Sameer Alsharif, and Chokri Ben Amar. A New Blind IoT-Based MP3 Audio Watermarking Scheme for Content Integrity Checking and Copyright Protection. *Wireless Communications and Mobile Computing*, 2022, 2022.
- [78] Andrew Critch. WordSig: QR streams enabling platform-independent self-identification that’s impossible to deepfake. *arXiv preprint arXiv:2207.10806*, 2022.
- [79] Irtaza Shahid and Nirupam Roy. "Is this my president speaking?" Tamper-proofing Speech in Live Recordings. In *Proceedings of the 21st Annual International Conference on Mobile Systems, Applications and Services*, pages 219–232, 2023.
- [80] Maria Caria, Gabriele Sara, Giuseppe Todde, Marco Polese, and Antonio Pazzona. Exploring smart glasses for augmented reality: A valuable and integrative tool in precision livestock farming. *Animals*, 9(11):903, 2019.
- [81] Candice R. Gerstner and Hany Farid. Detecting real-time deep-fake videos using active illumination. In *Proc. of CVPR*, pages 53–60, 2022.
- [82] Jiacheng Shang and Jie Wu. Protecting Real-time Video Chat against Fake Facial Videos Generated by Face Reenactment. In *International Conference on Distributed Computing Systems*, pages 689–699, 2020.
- [83] Hongbo Liu, Zhihua Li, Yucheng Xie, Ruizhe Jiang, Yan Wang, Xiaonan Guo, and Yingying Chen. LiveScreen: Video Chat Liveness Detection Leveraging Skin Reflection. In *Proc. of INFOCOM*, pages 1083–1092, 2020.
- [84] Habiba Farrukh, Reham Mohamed Aburas, Siyuan Cao, and He Wang. FaceRevlio: A Face Liveness Detection System for Smartphones with a Single Front Camera. In *Proc. of MobiCom*, 2020.
- [85] What is a frame rate? <https://www.adobe.com/creativecloud/video/discover/frame-rate.html>.
- [86] O. Wiles, A.S. Koepke, and A. Zisserman. Self-supervised learning of a facial attribute embedding from video. In *British Machine Vision Conference*, 2018.
- [87] Qiongqiong Wang, Koji Okabe, Kong Aik Lee, Hitoshi Yamamoto, and Takafumi Koshinaka. Attention mechanism in speaker recognition: What does it learn in deep speaker embedding? In *2018 IEEE Spoken Language Technology Workshop*, pages 1052–1059, 2018.
- [88] Moses S Charikar. Similarity estimation techniques from rounding algorithms. In *Proceedings of the thirty-fourth annual ACM symposium on Theory of computing*, pages 380–388, 2002.
- [89] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4690–4699, 2019.
- [90] InsightFace: 2D and 3D Face Analysis Project. <https://github.com/deepinsight/insightface/tree/master>.
- [91] Ultra-light-fast-generic-face-detector-1mb. GitHub repository.
- [92] Face landmark detection guide. https://developers.google.com/mediapipe/solutions/vision/face_landmarker.
- [93] F.J. Ferri, P. Pudil, M. Hatef, and J. Kittler. Comparative study of techniques for large-scale feature selection* this work was supported by a serc grant gr/e 97549. the first author was also supported by a fpi grant from the spanish mec, pf92 73546684. In *Pattern Recognition in Practice IV*, volume 16, pages 403–413. North-Holland, 1994.
- [94] Whitfield Diffie and Martin E Hellman. New directions in cryptography. In *Democratizing Cryptography: The Work of Whitfield Diffie and Martin Hellman*, pages 365–390, 1976.
- [95] S. Farrell S. Boeyen R. Housley W. Polk D. Cooper, S. Santesson. RFC 5208 - Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. <https://datatracker.ietf.org/doc/html/rfc5280>, 2008.
- [96] Hui-Yu Lee, Hao-Min Lin, Yu-Lin Wei, Hsin-I Wu, Hsin-Mu Tsai, and Kate Ching-Ju Lin. RollingLight: Enabling Line-of-Sight Light-to-Camera Communications. In *Proc. of MobiSys, MobiSys ’15*, 2015.
- [97] Daniel Cotting, Martin Naef, Markus Gross, and Henry Fuchs. Embedding imperceptible patterns into projected images for simultaneous acquisition and display. In *Third IEEE and ACM International Symposium on Mixed and Augmented Reality*, pages 100–109. IEEE, 2004.
- [98] Kensei Jo, Mohit Gupta, and Shree K. Nayar. DisCo: Display-Camera Communication Using Rolling Shutter Sensors. *ACM Trans. Graph.*, 35(5), 2016.
- [99] Tsutomu Kusanagi, Shingo Kagami, and Koichi Hashimoto. [POSTER] Lighting Markers: Synchronization-free Single-shot Detection of Imperceptible AR Markers Embedded in a High-Speed Video Display. In *2017 IEEE International Symposium on Mixed and Augmented Reality (ISMAR-Adjunct)*, pages 229–234, 2017.
- [100] Hao Cui, Huanyu Bian, Weiming Zhang, and Nenghai Yu. UnseenCode: Invisible On-screen Barcode with Image-based Extraction. In *Proc. of INFOCOM*, pages 1315–1323, 2019.
- [101] Akira Matsumoto, Satoshi Abe, Takefumi Hiraki, Shogo Fukushima, and Takeshi Naemura. Imperceptible AR Markers for Near-Screen Mobile Interaction. *IEEE Access*, 7:79927–79933, 2019.
- [102] Han Fang, Dongdong Chen, Feng Wang, Zehua Ma, Honggu Liu, Wenbo Zhou, Weiming Zhang, and Nenghai Yu. TERA: Screen-to-Camera Image Code With Transparency, Efficiency, Robustness and Adaptability. *IEEE Transactions on Multimedia*, 24:955–967, 2022.
- [103] Grace Woo, Andy Lippman, and Ramesh Raskar. VR Codes: Unobtrusive and active visual codes for interaction by exploiting rolling shutter. In *IEEE International Symposium on Mixed and Augmented Reality (ISMAR-Adjunct)*, pages 59–64, 2012.
- [104] Kai Zhang, Chenshu Wu, Chaofan Yang, Yi Zhao, Kehong Huang, Chunyi Peng, Yunhao Liu, and Zheng Yang. ChromaCode: A Fully Imperceptible Screen-Camera Communication System. In *Proc. of MobiCom*, page 575–590, 2018.
- [105] Kaihua Song, Ning Liu, Zhongpai Gao, Jiahe Zhang, Guangtao Zhai, and Xiaoping Zhang. Deep Restoration of Invisible QR Code from TPVM Display. In *IEEE International Conference on Multimedia and Expo Workshops*, pages 1–6, 2020.
- [106] Anran Wang, Chunyi Peng, Ouyang Zhang, Guobin Shen, and Bing Zeng. In-Frame: Multiflexing Full-Frame Visible Communication Channel for Humans and Devices. 2014.
- [107] Anran Wang, Zhuoran Li, Chunyi Peng, Guobin Shen, Gan Fang, and Bing Zeng. InFrame++: Achieve Simultaneous Screen-Human Viewing and Hidden Screen-Camera Communication. In *Proc. of MobiSys*, page 181–195, 2015.
- [108] Zhongpai Gao, Guangtao Zhai, and Chunjia Hu. The Invisible QR Code. In *ACM International Conference on Multimedia*, 2015.
- [109] Xiao Zhang, Jiqiang Liu, Zhongjie Ba, Yaodong Tao, and Xiaochun Cheng. MobiScan: An enhanced invisible screen-camera communication system for IoT applications. *Transactions on Emerging Telecommunications Technologies*, 33, 2022.

- [110] Chunjia Hu, Guangtao Zhai, and Zhongpai Gao. Visible Light Communication via Temporal Psycho-Visual Modulation. In *ACM International Conference on Multimedia*, page 785–788, 2015.
- [111] Yiyan Yang, Zhongpai Gao, and Guangtao Zhai. LRS-Net: invisible QR Code embedding, detection, and restoration. In *2021 International Conference on Visual Communications and Image Processing (VCIP)*, pages 1–5, 2021.
- [112] Lan Zhang, Cheng Bo, Jiahui Hou, Xiang-Yang Li, Yu Wang, Kebin Liu, and LYunhao Liu. Kaleido: You Can Watch It But Cannot Record It. In *Proc. of MobiCom*, page 372–385, 2015.
- [113] Feng Wang, Hang Zhou, Han Fang, Weiming Zhang, and Nenghai Yu. Noise Simulation-Based Deep Optical Watermarking. In *Artificial Intelligence and Security*, pages 283–298, 2022.
- [114] Hiroshi Unno and Kazutake Uehira. Lighting Technique for Attaching Invisible Information Onto Real Objects Using Temporally and Spatially Color-Intensity Modulated Light. *IEEE Transactions on Industry Applications*, 56(6):7202–7207, 2020.
- [115] Tianxing Li, Chuankai An, Xinran Xiao, Andrew T. Campbell, and Xia Zhou. Real-Time Screen-Camera Communication Behind Any Scene. In *Proc. of MobiSys*, page 197–211, 2015.
- [116] Danielle Szafr. Color discrimination as a function of exposure time*. *Journal of Vision*, 17(10):1189, August 2017.
- [117] Michael H. Siegel. Color discrimination as a function of exposure time*. *Journal of the Optical Society of America*, 55(5):566–568, May 1965.
- [118] Gaurav Sharma, Wencheng Wu, and Edul N Dalal. The ciede2000 color-difference formula: Implementation notes, supplementary test data, and mathematical observations. *Color Research & Application*, 30(1):21–30, 2005.
- [119] Satoshi Suzuki. Topological structural analysis of digitized binary images by border following. *Computer Vision, Graphics, and Image Processing*, 30(1):32–46, 1985.
- [120] Anubhav Agarwal, CV Jawahar, and PJ Narayanan. A survey of planar homography estimation techniques. *Centre for Visual Information Technology, Tech. Rep. IITR/TR/2005/12*, 2005.
- [121] TI. DLPDLCR230NPEVM. <https://www.ti.com/tool/DLPDLCR230NPEVM>.
- [122] Arducum. 8MP IMX179 Arducum. <https://www.arducum.com/product/8mp-imx179-autofocus-usb-camera-module-with-waterproof-protection-case-for-windows-linux-android-and-mac-os/>.
- [123] Raspberry Pi. Raspberry Pi 4. <https://www.raspberrypi.com/products/raspberrypi-4-model-b/>.
- [124] R. Canetti H. Krawczyk, M. Bellare. HMAC: Keyed-Hashing for Message Authentication. <https://datatracker.ietf.org/doc/html/rfc2104#section-5>, 1997.
- [125] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. pages 6105–6114, 2019.
- [126] Aruna Sankaranarayanan, Matthew Groh, Rosalind Picard, and Andrew Lippman. The presidential deepfakes dataset. In *AloFAL: 1st workshop on adverse impacts and collateral effects of artificial intelligence technologies*, volume 2942, 2021.
- [127] Mokose. MOKOSE 4K@30fps USB Camera Webcam. <https://www.mokose.com/products/mokose-4k-30fps-usb-camera-webcam-uv-c-free-drive-compatible-windows-mac-os-x-linux>.
- [128] Canon. Canon EOS 60D. <https://www.usa.canon.com/support/p/eos-60d/srslid=AfmBOptKQWLY9tzKSu2nzaw8TFlviCAXdC3U0luRFYsPuDXqIJ9EnO9>.
- [129] Yuval Nirkin, Yosi Keller, and Tal Hassner. FSGAN: Subject agnostic face swapping and reenactment. In *Proc. of CVPR*, pages 7184–7193, 2019.
- [130] Fa-Ting Hong, Longhao Zhang, Li Shen, and Dan Xu. Depth-aware generative adversarial network for talking head video generation. 2022.
- [131] Aliaksandr Siarohin, Stéphane Lathuilière, Sergey Tulyakov, Elisa Ricci, and Nicu Sebe. First order motion model for image animation. *Advances in neural information processing systems*, 32, 2019.
- [132] Jiadong Wang, Xinyuan Qian, Malu Zhang, Robby T Tan, and Haizhou Li. Seeing What You Said: Talking Face Generation Guided by a Lip Reading Expert. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14653–14662, 2023.
- [133] Wenxuan Zhang, Xiaodong Cun, Xuan Wang, Yong Zhang, Xi Shen, Yu Guo, Ying Shan, and Fei Wang. SadTalker: Learning Realistic 3D Motion Coefficients for Stylized Audio-Driven Single Image Talking Face Animation. In *Proc. of CVPR*, pages 8652–8661, 2023.
- [134] Zhiyuan Yan, Yong Zhang, Xinhang Yuan, Siwei Lyu, and Baoyuan Wu. Deepfakebench: A comprehensive benchmark of deepfake detection. pages 4534–4565, 2023.
- [135] Gary B Huang, Marwan Mattar, Tamara Berg, and Eric Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. In *Workshop on faces in 'Real-Life' Images: detection, alignment, and recognition*, 2008.
- [136] John S Garofolo. Timit acoustic phonetic continuous speech corpus. *Linguistic Data Consortium*, 1993, 1993.
- [137] Zhixi Cai, Kalin Stefanov, Abhinav Dhall, and Munawar Hayat. Do you really mean that? content driven audio-bisual deepfake dataset and multimodal method for temporal forgery localization. In *2022 International Conference on Digital Image Computing: Techniques and Applications*, pages 1–10, 2022.
- [138] Prana Air. Illuminance Levels Indoors: Your Standard Lux Level Char. <https://www.pranaair.com/blog/illuminance-levels-indoors-the-standard-lux-levels/#:-:text=For%20general%20office%20tasks%2C%20a,between%20750%20to%201000%20lux>.
- [139] Apple. Adjust a photo's light, exposure, and more in Photos on Mac. <https://support.apple.com/guide/photos/adjust-a-photos-light-exposure-and-more-pht806aea6a6/9.0/mac/14.0>, 2024.
- [140] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy*, pages 39–57, 2017.
- [141] Arsha Nagrani, Joon Son Chung, Weidi Xie, and Andrew Zisserman. Voxceleb: Large-scale speaker verification in the wild. *Computer Science and Language*, 2019.
- [142] Shawn Shan, Emily Wenger, Jiayun Zhang, Huiying Li, Haitao Zheng, and Ben Y Zhao. Fawkes: Protecting privacy against unauthorized deep learning models. In *29th USENIX Security Symposium*, pages 1589–1604, 2020.
- [143] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.
- [144] Bolun Wang, Yuanshun Yao, Bimal Viswanath, Haitao Zheng, and Ben Y Zhao. With great training comes great vulnerability: Practical attacks against transfer learning. In *27th USENIX Security Symposium*, pages 1281–1297, 2018.
- [145] Josephine Passananti, Stanley Wu, Shawn Shan, Haitao Zheng, and Ben Y Zhao. Disrupting style mimicry attacks on video imagery. *arXiv preprint arXiv:2405.06865*, 2024.
- [146] Evani Raddya-Dixit, Sanghyun Hong, Nicholas Carlini, and Florian Tramèr. Data poisoning won't save you from facial recognition. 2021.
- [147] Jan Hendrik Metzen, Tim Genewein, Volker Fischer, and Bastian Bischoff. On detecting adversarial perturbations. In *International Conference on Learning Representations*, 2017.
- [148] Apple 3DFaceScan. <https://apps.apple.com/us/app/3dfacescan-structure-sdk/id6473282888>.
- [149] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proc. of CVPR*, pages 586–595, 2018.
- [150] Yasuyuki Matsushita, Eyal Ofek, Weina Ge, Xiaoou Tang, and Heung-Yeung Shum. Full-frame video stabilization with motion inpainting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(7):1150–1163, 2006.
- [151] Feng Liu, Michael Gleicher, Jue Wang, Hailin Jin, and Aseem Agarwala. Subspace video stabilization. *ACM Transactions on Graphics*, 30(1):1–10, 2011.
- [152] Kenji Okuma, James J Little, and David G Lowe. Automatic rectification of long image sequences. In *Asian Conference on Computer Vision*, volume 9, 2004.
- [153] Open Neural Network Exchange. <https://onnx.ai/>.
- [154] Learned Perceptual Image Patch Similarity (LPIPS). https://lightning.ai/docs/to-rchmetrics/stable/image/learned_perceptual_image_patch_similarity.html.
- [155] Cassidy Laidlaw, Sahil Singla, and Soheil Feizi. Perceptual adversarial robustness: Defense against unseen threat models. *arXiv preprint arXiv:2006.12655*, 2020.

Appendix

A.1 Adaptive Optical Embedding

We formalize our adaptive optical embedding method (§5.3) in the following algorithm.

Input: V : Window's core unit video frames
 I : required intensity for each SLM bitmap cell j
Output: c_{SLM} : RGB value for each SLM bitmap cell j
 I : updated required intensity for SLM cells.

```

1 Function Adapt( $V, I$ ):
2   Let  $\Phi_{max}$  be the perceptibility threshold
3   Let  $\beta_{max}$  be the BER threshold
4   Let  $\delta$  be intensity increment/decrement value
5    $d \leftarrow$  Extract data embedded in  $V$ 
6    $\beta \leftarrow \text{BER}(d)$ 
7   if  $\beta \geq \beta_{max}$ 
8      $I \leftarrow$  Increment all  $I^j$  by  $\delta$ 
9   foreach cell  $j$  do
10     $c_{pon}^j, c_{poff}^j \leftarrow$  Patch  $j$  color w/ and w/o SLM light
11    if  $\beta < \beta_{max}$  and  $\text{CIEDE2K}(c_{pon}^j, c_{poff}^j) \geq \Phi_{max}$ 
12       $I^j \leftarrow I^j - \delta$ 
13     $c_{SLM}^j \leftarrow \text{Equation1}(c_{poff}^j, I^j)$ 
14 return  $c_{SLM}, I$ 

```

Algorithm 1: Adaptation performed after completion of a window to set upcoming SLM bitmap cell colors and intensities.

The below equation, employed on Line 13 configures bitmap colors to match the patch of projection surface they are incident upon.

$$c_{SLM}^j = (\alpha * R_{poff}^j, \alpha * G_{poff}^j, \alpha * B_{poff}^j), \quad (1)$$

subj. to $\alpha * R_{poff}^j + \alpha * G_{poff}^j + \alpha * B_{poff}^j = I_j$,

where $R_{poff}^j, G_{poff}^j, B_{poff}^j$ are the RGB values of c_{poff}^j .

The user-specified values of ϕ_{max} and β_{max} configure the trade-off between data error rate, in the form of BER and perceptibility. δ tunes the system's adaptation speed, with a higher value enabling faster adaptation but risking "overshooting" in terms of perceptibility. Furthermore, we note that if BER is above β_{max} , all I^j are incremented by δ . The rationale for the global increase in Line 8 is that, while cells' perceptibilities are independent, robustness depends on all cells since data is distributed across them. We also note I^j may be incremented even when $\text{CIEDE2000}(c_{pon}^j, c_{poff}^j)$ exceeds Φ_{max} , consciously prioritizing robustness. This is a conscious design decision to prioritize robustness, as we believe the impact of an unrecoverable signature to be worse than that of embedding visibility.

A.2 Optical Modulation Configuration

For all prototype experiments, we utilize 640 x 360 pixel SLM bitmaps, configured to fit 16 x 9 cells. This corresponds to 87 data cells and 32 synchronization cells, in addition to our four larger localization cells. We set f_d to be 3 Hz, as we found that BER increased significantly at larger frequencies. Finally we set f_l to be 6 Hz, and our window duration to be 4.5 s, consisting of .5 s of downtime followed by 4 s of modulation. Thus we can embed 12

bits per data cell, i.e., $12 * 87 = 1044$ bits per window. We configure the adaptive embedding algorithm (Algorithm 1) with $\beta_{max} = 0$, $\phi_{max} = 5$, and $\delta = 5$.

A.3 Adversarial Deepfake Model Training

Here we provide details on the training, implementation, and evaluation of our DaGAN and FSGAN models.

DaGAN The DaGAN reenactment model forward pass takes as input two face images: a source image of the victim and an attacker-provided "driving" image. It synthesizes a fake image, where the victim possesses a new facial expression or pose, to match that in the driving image. The model can be called for each pair of frames in an original video and driving video, with model outputs concatenated to form a video.

During training of our modified DaGAN model, we extract the 16 FaceMesh features considered by Spotlight's dynamic feature vectors (§4.2) from both *source* image and the generated *fake* image. Our loss function L_{adv} is then defined as follows:

$$L_{adv} = L + \alpha \Theta(Dyn_{src}, Dyn_{fake})$$

where L is the original DaGAN loss function, $Dyn \in \mathbb{R}^{16}$ are vectors containing the FaceMesh results, and Θ is the cosine similarity function. The coefficient α weights the important of preserving dynamic features in generated content.

To implement this modified version of DaGAN, we first develop a fully differentiable PyTorch implementation of FaceMesh. This is needed because Google only provides FaceMech as a LiteRT module, which is designed for inference only and does not have the necessary mechanisms for calculating gradients during back-propagation. We use ONNX-converted [153] ports of each of the three neural networks underlying FaceMesh and integrate them to the best of our abilities by referencing MediaPipe's public model cards. Our implementation outputs facial landmarks with an average difference of 2 pixels and blendshape scores with an average difference of 0.11 (arbitrary units, ranging from 0 to 1) from their official implementation counterparts when run on DaGAN's training dataset.

We fine-tune the model from the checkpoint released by the authors, applying early stopping based on validation loss. We use the same learning rate and parameters employed in their original implementation, as well as their same training data, sourced from VoxCeleb. We create evaluation videos by randomly choosing 55 pairs of videos from the VoxCeleb test split and using one to drive the other.

FSGAN The FSGAN faceswapping model similarly takes as input a source image of the victim and an attacker-provided target face image. The model synthesizes a fake image in which the victim's face is supplanted with the target's face, effectively modifying the portrayed identity.

During training of our modified FSGAN models, we extract the ArcFace embedding from both the source and fake images. Similar to above, our loss function L_{adv} is then defined as follows:

$$L_{adv} = L + \alpha \Theta(Arc_{src}, Arc_{fake})$$

% Window Modified	Passive Detector										Ours
	Meso4 [57]	Xception [60]	Capsule [49]	Efficient [125]	SRM [50]	SPSL [51]	Recce [52]	UCF [53]	TALL [54]	AltFreeze. [55]	
10-20	0.53	0.59	0.57	0.57	0.55	0.55	0.55	0.60	0.55	0.47	0.72
20-30	0.58	0.62	0.55	0.58	0.55	0.58	0.55	0.62	0.57	0.47	0.84
30-40	0.60	0.64	0.58	0.60	0.58	0.60	0.56	0.63	0.58	0.47	0.90
40-50	0.59	0.67	0.59	0.63	0.58	0.61	0.59	0.66	0.61	0.48	0.92
100	0.75	0.90	0.80	0.81	0.76	0.76	0.83	0.84	0.84	0.26	0.98

Table 2: Comparison of AUC scores achieved by passive detectors and Spotlight’s 150-bit dynamic feature hashes on our multi-posed, fine-grained reenactment dataset (§9.1), in which modifications of various granularities (by percentage duration modified) are applied to each 4.5 s window. Best performing method is bolded.

(a) Distance and viewing angle.									
	2m			3.5m			5m		
	0°	45°	60°	0°	45°	60°	0°	45°	60°
R	0.01	0.02	0.02	0.05	0.04	0.05	0.13	0.13	0.12
V	0.00	0.00	0.00	0.00	0.00	0.00	0.04	0.04	0.03
F	0.00	0.00	0.00	0.00	0.00	0.00	0.03	0.03	0.01

(b) Ambient light intensity (lx) and projection surface.										
	320			750			530	3k	3.5k	2.6k
	S1	S2	S3	S1	S2	S3	S4	S5	S6	S7
R	0.01	0.07	0.01	0.01	0.08	0.01	0.70	0.01	0.03	0.01
V	0.00	0.04	0.00	0.00	0.02	0.00	0.05	0.00	0.00	0.00
F	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

(c) Device type.					
	Google Pixel	iPhone (HD)	iPhone (ProRes)	Webcam	DSLR
R	0.01	0.01	0.01	0.01	0.00
V	0.00	0.00	0.00	0.00	0.00
F	0.00	0.00	0.00	0.00	0.00

(d) Video transcoding and compression.								
	Transcoding			Bitrate Decrease				
	None	H.264	MPEG4	10%	30%	50%	70%	90%
R	0.01	0.02	0.04	0.03	0.02	0.03	0.03	0.02
V	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
F	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

(e) Exposure (E), Contrast (C), and Auto/Mono adjustment.						
	C-50	C+50	E-50	E+50	Auto	Mono
R	0.04	0.01	0.01	0.02	0.01	0.01
V	0.00	0.00	0.00	0.00	0.00	0.00
F	0.00	0.00	0.00	0.00	0.00	0.00

(f) In-video cell resolution.				
	30 x 30 px	35 x 35 px	40 x 40 px	50 x 50 px
R	0.12	0.05	0.04	0.02
V	0.03	0.00	0.00	0.00
F	0.01	0.00	0.00	0.00

Table 3: Signature embedding robustness in terms of bit error rate (BER) across recording factors. We report the BER at each stage of error correction (§5.1): raw (R), before error correction; after the concatenated error corrector Viterbi decoding (V); and final (F), after RS error correction.

where L is the original FSGAN loss function, $Arc \in \mathbb{R}^{512}$ are the extracted ArcFace embeddings, and Θ is the cosine similarity function. The coefficient α weights the important of preserving identity features in generated content.

We train the FSGAN model from scratch using data from the VoxCeleb dataset, because the authors do not release all weights necessary for fine-tuning. We use the same learning parameters employed in the original version and applying early stopping based on validation loss. We create evaluation videos by randomly choosing 55 pairs of videos from the VoxCeleb dataset (ensuring they portray different identities) and generating a faceswap deepfake for each.

A.4 Perceptibility Evaluation Details

Here, we detail the setup of our perceptibility user study and LPIPS evaluation and further discuss the studies’ results.

User study We invite groups of 9 participants (recruited via email, with ages ranging from 18 to 34) to each of the test environments (Figure 9). At each environment participants were asked to assess the projection surface during four 45 s trials in which the core unit either performed embedding of a random bitstream or was

powered off as a control scenario (done for two randomly selected trials). Participants were informed that light may be projected and shown the projection region boundaries. During each trial they were allowed to walk freely up to 1.5 m of the surface, and asked to answer two questions once ready:

- Q1 Do you believe the light pattern is present in the video?(Y/N)*
Q2 How obtrusive do you find the pattern? (Low/Medium/High Obtrusiveness).

Finally, we invite 20 participants (recruited via email, ages 18-65) to assess videos of the projection surface. For each video, the boundaries of the projection region were marked to enable participants to accurately examine it. Participants were permitted to freely zoom into any portion of frames, pause, and rewind.

As shown in the bottom two figures of Figure 11, for each environment we plot (1) $\Delta = TPR - FPR$, where TPR is the rate at which participants responded "Yes" to Q1 when embedding was occurring, and FPR is the rate at which they incorrectly responded "Yes"; and (2) the average response to Q2, excluding those given with a Q1 false positive. A Δ value ≤ 0 indicates that embedding is effectively imperceptible, as participants perform no better than random at

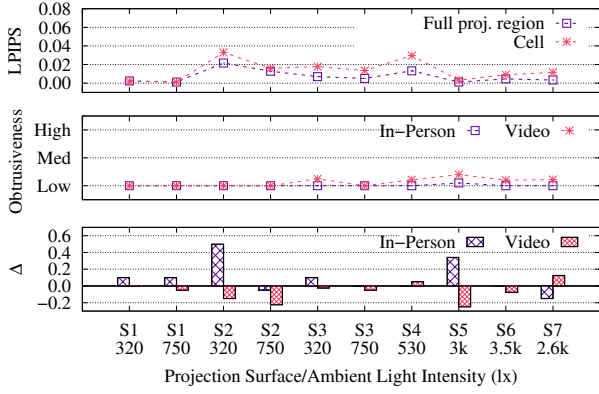


Figure 11: Embedding perceptibility results. Top axis: Average LPIPS across pairs of video frames captured w/ and w/o Spotlight operation. All scores are significantly below the established LPIPS perceptibility threshold of 0.5 [155]. Middle: average obtrusiveness reported by users study respondents. Bottom: user study $\Delta = \text{TPR} - \text{FPR}$, normalizing true positive identifications of embedding occurrence to false positive identifications during control trials. A nonpositive value indicates imperceptibility.

detecting it. We observe this in all but two video cases. In person, $\Delta \leq 0.2$ in all but two environments. Respondents uniformly report low obtrusiveness in-video and live. The primary factors influencing perceptibility are the surface’s texture and color. Darker, homogeneous surfaces (i.e., S2, S4, S5) fundamentally contrast with impinging light, whereas brighter-colored, more complex ones (e.g., S1, S3, S5) provide a camouflage. Increasing ambient light intensity can counteract this (as with S4 and S2 at 750 lx) by increasing the baseline brightness of a surface’s appearance.

Perceptual metrics We additionally evaluate optical signature perceptibility in video using the learned perceptual loss (LPIPS) [149, 154] metric. This metric has been shown to have superior correlation with human perception, especially in the context of fine-grained changes such as those introduced by Spotlight’s optical modulations. LPIPS takes as input two images and outputs a score ranging from 0 to 1, where a lower value indicates the inputs are more perceptually similar. In particular, prior works show humans cannot sense differences when LPIPS is below 0.5 [155]. To apply these image-level metrics to our videos, we compute the average score between crops of the full projection region and individual cells in 5,000 pairs of frames captured with and without Spotlight operating. We compute at both the full projection region and cell level to understand perceptibility at both scales differences occur.

The top panel of Figure 11 shows that full region and cell-level scores are highly correlated. Further, all LPIPS scores are over ten times lower than then established LPIPS perceptibility threshold.

A.5 Proofs and Definitions: Locality-Sensitive Hashing

Here we detail our methodology for utilizing cosine similarity-based LSH for verification. We include both a definition and discussion of the methodology at a high level (Definition 2) and derive an

equation for the relationship between hash size and verification performance (Theorem 2), which illustrates that verification performance has no dependence on input vector dimensionality.

DEFINITION 1 (COSINE SIMILARITY LSH SCHEME).

We utilize the locality sensitive hashing scheme for the cosine similarity as proposed in [88]. Let \vec{r} be a vector in \mathbb{R}^n chosen randomly from the n -dimensional Gaussian distribution (i.e., each coordinate is drawn from a Gaussian distribution). Let the hash function $h_{\vec{r}} : \vec{u} \in \mathbb{R}^n \mapsto \{0, 1\}$ be defined as follows:

$$h_{\vec{r}}(\vec{u}) = \begin{cases} 1 & \text{if } \vec{r} \cdot \vec{u} \geq 0 \\ 0 & \text{if } \vec{r} \cdot \vec{u} < 0 \end{cases}$$

The locality sensitive hashing scheme H is defined as $H(\vec{u}) = \{h_1(\vec{u}), h_2(\vec{u}), \dots, h_k(\vec{u})\}$, where h_i are independently and randomly chosen hash functions from \mathbb{F} . Thus, given an input $\vec{u} \in \mathbb{R}^n$, H outputs a k -bit vector, formed by concatenating the single bit outputs of each of its hash functions h_i .

The key idea of this scheme is that the sign of a vector \vec{x} ’s projection onto \vec{r}_i is fundamentally related to the angle between \vec{vecx} and \vec{r}_i . Thus if \vec{u} and \vec{v} have a high cosine similarity, their projections onto \vec{r}_i are more likely to have the same sign. Each bit in the hash serves as an additional “sample” to aid in approximating Θ ; including more bits increases the probability that D correctly reflects Θ . This captures the cosine similarity with extreme space efficiency.

DEFINITION 2 (VERIFICATION USING LSH).

We depart from the formal definition of a traditional verification problem: given two feature vectors, we would like to confirm that they represent the same source (e.g., face embeddings corresponding to the same identity). Two feature vectors u, v are said to correspond to the source if $\Theta(\vec{u}, \vec{v}) < \theta_{th}$, where θ_{th} is a decision threshold. Otherwise, the vectors are said to correspond to different sources.

We can similarly formalize the verification problem on hashed feature vectors: the hashes of two feature vectors, $H_{cos}(u), H_{cos}(v)$ are said to correspond to the same source if $D(H(\vec{v}), H(\vec{u})) > d_{th}$, where d_{th} is a decision boundary for verification on hashed vectors. If θ_{th} is the optimal decision boundary for verification on the raw feature vectors, intuitively, the optimal value of d_{th} should be the expected value of $D(H(\vec{v}), H(\vec{u}))$ for two vectors u, v s.t. $\Theta(\vec{u}, \vec{v}) = \theta_{th}$. From Equation 1, this is $\frac{k\theta_{th}}{\pi}$.

THEOREM 1 (EXPECTED VALUE OF HAMMING DISTANCE). Let H be a locality sensitive hashing scheme defined according to Definition 1 and D be the Hamming distance function. The expected value of $D(H(\vec{v}), H(\vec{u}))$ is $\frac{k\Theta(\vec{u}, \vec{v})}{\pi}$.

PROOF. Recall that D , the Hamming distance function, gives the number of positions at which the values of two bitstrings differ. Since the value of each bit of H ’s output is determined by a hash function h_i , the expected value of $D(H(\vec{v}), H(\vec{u}))$ is the expected number of H ’s k hash functions h_i for which $h_i(\vec{u}) \neq h_i(\vec{v})$. From [88],

$$\Pr[h(\vec{u}) \neq h(\vec{v})] = \frac{\Theta(\vec{u}, \vec{v})}{\pi}.$$

Therefore, the probability that $D(H(\vec{v}), H(\vec{u})) = n$, (i.e., the outputs of exactly n of the hash functions h_i differ) is

$$\begin{aligned} & \binom{k}{n} \Pr[h(\vec{u}) \neq h(\vec{v})]^n \Pr[h(\vec{u}) = h(\vec{v})]^{k-n} \\ &= \binom{k}{n} \left(\frac{\Theta(\vec{u}, \vec{v})}{\pi} \right)^n \left(1 - \frac{\Theta(\vec{u}, \vec{v})}{\pi} \right)^{k-n} \end{aligned} \quad (2)$$

The expected value of $D(H(\vec{v}), H(\vec{u}))$ is thus

$$\sum_{n=0}^k n \binom{k}{n} \left(\frac{\Theta(\vec{u}, \vec{v})}{\pi} \right)^n \left(1 - \frac{\Theta(\vec{u}, \vec{v})}{\pi} \right)^{k-n} = \frac{k\Theta(\vec{u}, \vec{v})}{\pi} \quad (3)$$

□

THEOREM 2 (IMPACT OF k ON VERIFICATION PERFORMANCE). *To assess the impact of the hash size k on verification performance, we seek the probability $P_{\theta_{th}}(k)$ that all decisions obtained from verification on k -bit hashed feature vectors are the same as those obtained from performing verification on the raw vectors with a decision boundary of θ_{th} . This event indicates that the hashed feature vectors perfectly preserve verification performance. Thus we can view the probability of its occurrence as a measure of the hashed vectors' performance relative to that of the raw vectors. $P_{\theta_{th}}(k)$ is given by the following equation*

$$\begin{aligned} P_{\theta_{th}}(k) &= \exp \left(\int_0^{\theta_{th}} \ln \left(\sum_{n=0}^{\frac{k\theta_{th}}{\pi}} \binom{k}{n} \left(\frac{\theta}{\pi} \right)^n \left(1 - \frac{\theta}{\pi} \right)^{k-n} \right) d\theta \right) * \\ &\quad \exp \left(\int_{\theta_{th}}^{\pi} \ln \left(\sum_{n=\frac{k\theta_{th}}{\pi}}^k \binom{k}{n} \left(\frac{\theta}{\pi} \right)^n \left(1 - \frac{\theta}{\pi} \right)^{k-n} \right) d\theta \right) \end{aligned}$$

PROOF. Based on Definition 2, verification decisions are obtained from raw or hashed feature vectors using the criteria $\Theta(\vec{u}, \vec{v}) < \theta_{th}$ or $D(H(\vec{v}), H(\vec{u})) < \frac{k\theta_{th}}{\pi}$, respectively. Thus we have

$$\begin{aligned} P_{\theta_{th}}(k) &= \Pr[\\ &\quad \forall(\vec{u}, \vec{v}) \in \{(\vec{u}, \vec{v}) : \Theta(\vec{u}, \vec{v}) \leq \theta_{th}\}, \quad D(H(\vec{v}), H(\vec{u})) \leq \frac{k\theta_{th}}{\pi} \\ &\quad \cap \\ &\quad \forall(\vec{u}, \vec{v}) \in \{(\vec{u}, \vec{v}) : \Theta(\vec{u}, \vec{v}) > \theta_{th}\} \quad D(H(\vec{v}), H(\vec{u})) > \frac{k\theta_{th}}{\pi} \\ &\quad] \end{aligned}$$

Intuitively,

$$\Pr[\forall(\vec{u}, \vec{v}) \in \{(\vec{u}, \vec{v}) : \Theta(\vec{u}, \vec{v}) \leq \theta_{th}\}, \quad D(H(\vec{v}), H(\vec{u})) \leq \frac{k\theta_{th}}{\pi}]$$

is the probability that for all \vec{u}, \vec{v} satisfying $\Theta(\vec{u}, \vec{v}) \leq \theta_{th}$, at most $\frac{k\theta_{th}}{\pi}$ bits of $H(\vec{u})$ and $H(\vec{v})$ differ.

Using Equation 2 and the independence of each comparison of $(\vec{u}, \vec{v}) \in \{(\vec{u}, \vec{v}) : \Theta(\vec{u}, \vec{v}) \leq \theta_{th}\}$,

$$\begin{aligned} & \Pr[\forall(\vec{u}, \vec{v}) \in \{(\vec{u}, \vec{v}) : \Theta(\vec{u}, \vec{v}) \leq \theta_{th}\}, \quad D(H(\vec{v}), H(\vec{u})) \leq \frac{k\theta_{th}}{\pi}] \\ &= \exp \left(\int_0^{\theta_{th}} \ln \left(\sum_{n=0}^{\frac{k\theta_{th}}{\pi}} \binom{k}{n} \left(\frac{\theta}{\pi} \right)^n \left(1 - \frac{\theta}{\pi} \right)^{k-n} \right) d\theta \right) \end{aligned}$$

By the same logic,

$$\begin{aligned} & \Pr[\forall(\vec{u}, \vec{v}) \in \{(\vec{u}, \vec{v}) : \Theta(\vec{u}, \vec{v}) > \theta_{th}\}, \quad D(H(\vec{v}), H(\vec{u})) > \frac{k\theta_{th}}{\pi}] \\ &= \exp \left(\int_{\theta_{th}}^{\pi} \ln \left(\sum_{n=\frac{k\theta_{th}}{\pi}}^k \binom{k}{n} \left(\frac{\theta}{\pi} \right)^n \left(1 - \frac{\theta}{\pi} \right)^{k-n} \right) d\theta \right) \end{aligned}$$

We thus have:

$$\begin{aligned} P_{\theta_{th}}(k) &= \exp \left(\int_0^{\theta_{th}} \ln \left(\sum_{n=0}^{\frac{k\theta_{th}}{\pi}} \binom{k}{n} \left(\frac{\theta}{\pi} \right)^n \left(1 - \frac{\theta}{\pi} \right)^{k-n} \right) d\theta \right) * \\ &\quad \exp \left(\int_{\theta_{th}}^{\pi} \ln \left(\sum_{n=\frac{k\theta_{th}}{\pi}}^k \binom{k}{n} \left(\frac{\theta}{\pi} \right)^n \left(1 - \frac{\theta}{\pi} \right)^{k-n} \right) d\theta \right) \end{aligned}$$

□