Light2Lie: Detecting Deepfake Images Using Physical Reflectance Laws

Kavita Kumari[†], Sasha Behrouzi[†], Alessandro Pegoraro[†], Ahmad-Reza Sadeghi[†]
[†]*Technical University of Darmstadt*

***Abstract*—The rapid advancement of generative models such as GANs and diffusion-based architectures has led to the widespread creation of hyperrealistic synthetic images. Although these technologies drive innovation in the media and data generation, they also raise significant ethical, social, and security concerns. In response, numerous detection methods have been developed, including frequency domain analysis and deep learning classifiers. However, these approaches often struggle to generalize across unseen generative models and typically lack physical grounding, leaving them vulnerable to adaptive attacks and limited in interpretability.**

We propose Light2Lie, a physics-augmented deepfake detection framework that leverages principles of specular reflection, specifically the Fresnel reflectance model, to reveal inconsistencies in light–surface interactions that generative models struggle to reproduce effectively. Our method first employs a neural network to estimate the surface base reflectance and then derives a microfacet-inspired specular response map that encodes subtle geometric and optical discrepancies between real and synthetic images. This signal is integrated into a secondary classifier, as feature maps, that learns to distinguish the two classes based on reflectance-driven patterns. To further enhance robustness, we introduce a feedback refinement mechanism that updates the base reflectance model output using classification errors, tightly coupling physical modeling with the learning objective. Extensive experiments on multiple deepfake datasets demonstrate that our approach obtains better generalization performance to unseen generative model samples by getting up to 74% precision on diverse deepfake domains, outperforming state-of-the-art baselines while providing robust, physics-grounded decisions.

I. Introduction

The past decade has witnessed a dramatic rise in the capabilities of AI-generated content, particularly in the domain of synthetic image generation. Techniques such as Generative Adversarial Networks (GANs) [12] and diffusion models [17] have enabled the production of high-fidelity, photorealistic images that are increasingly indistinguishable from real ones. State-of-the-art models like StyleGAN2 [22], DALLE-2 [34], Imagen [45], and Stable Diffusion [41] can synthesize highly detailed human faces, landscapes, and artistic renderings with minimal human input. These advancements have revolutionized entertainment, design, and data augmentation, and continue to push the boundaries of creative and generative AI.

However, the same progress has led to a surge in deepfake content, manipulated or fabricated images intended to mislead viewers or falsify reality. Deepfakes have been implicated in political misinformation campaigns, celebrity impersonations, and fraudulent media incidents [58], [29]. For example, a viral deepfake of President Zelenskyy surrendering to Russian forces [3] sparked global concern over the role of synthetic media in wartime propaganda. Similarly, deepfake pornographic content using non-consenting individuals has been widely reported [50], raising ethical and legal alarms.

Existing detection methods. In response, a wide range of deepfake detection methods have emerged, typically falling into three main categories: 1) Frequency-domain techniques exploit spectral artifacts introduced by generative models, such as analyzing discrepancies in DCT and Fourier spectra respectively [6], [9]; 2) Spatial and statistical methods target inconsistencies in texture, color, or semantics, such as detecting eye-blinking anomalies [24] or leverage CNNs to extract local pixel-level artifacts [10], [62]; 3) More recent transformer-based and ensemble approaches aim to improve generalization using attention mechanisms[14] or self-supervised learning [26].

Additionally, with the rise of multimodal models, researchers have explored using pretrained Vision-Language Models (VLMs) to detect deepfakes. For example, Wang et al. [56] use CLIP embeddings to identify semantic-textural inconsistencies in generated images, while Dong et al. [7] exploit multimodal vision-language transformers for cross-domain deepfake detection. However, these methods are highly sensitive to prompt selection and domain shifts, and incur significant computational overhead.

Although existing methods perform well in controlled settings, they often fail to generalize to unseen generative models or distribution shifts. Many rely on superficial statistical artifacts [55], [52], leaving them vulnerable to adversarial perturbations and post-processing. Furthermore, most are purely data-driven, lacking the physical constraints of real-world image formation, which limits their robustness in open-world scenarios.

Our goal and contributions To address the limitations of existing AI-generated image detectors, we propose a physics-augmented detection framework, called **Light2Lie**, that builds

Network and Distributed System Security (NDSS) Symposium 2026
23-27 February 2026, San Diego, CA, USA
ISBN 979-8-9919276-8-0
https://dx.doi.org/10.14722/ndss.2026.230923
www.ndss-symposium.org

on how light interacts with real-world surfaces. Our core idea is that while AI-generated images can convincingly mimic texture and structure, they often fail to replicate the subtle, physically consistent behavior of light reflection, particularly the way highlights appear on real materials.

In natural images, light reflections behave in complex ways, depending on the surface's shape, roughness, and material properties. These reflective behaviors are governed by physical laws and are very hard to fake/reconstruct convincingly. To capture this and approximate this phenomenon in the image domain, we draw inspiration from classical computer graphics, particularly from the influential work of Blinn [4], who modeled surfaces as being composed of countless tiny mirror-like elements called microfacets (tiny planar surfaces). The way these microfacets reflect light determines the intensity and placement of highlights on a surface. This theory has long been used in physically-based rendering, and we *repurpose* it to analyze how plausible reflections are in digital images.

Our approach **Light2Lie** is inspired by how light interacts with real-world surfaces. We treat each pixel in an image as a tiny reflective patch and analyze how it would naturally reflect light. Rather than relying on external lighting setups or physical measurements, we infer these reflections directly from the image itself, based on the way its local textures and surface shapes would interact with light.

Real images contain small imperfections and irregular textures that scatter light in complex ways, while AI-generated images often appear unnaturally smooth, reflecting light more uniformly. To capture these differences, our framework operates in three stages: first, we extract the surface geometry which interprets the underlying structure of the image to approximate how light would interact with its microfacets; second, we estimate and learn the reflectivity of the input image sample to predict how reflective the image surface would appear under natural conditions by using a base neural network; and third, we combine the visual embedding features with these physical features to learn the subtle surface inconsistencies between the real and the fake sample, by using a secondary or main classifier and finally decide whether an image is real or generated.

Together, these components form a physics-guided detection pipeline that uncovers subtle inconsistencies in light–surface interactions, providing improved robustness and interpretability. Our framework, **Light2Lie**, focuses on reflection patterns that AI-generated images often fail to reproduce, offering a more reliable and generalizable solution to deepfake detection than conventional pixel-driven approaches.

While **Light2Lie** leverages physically inspired artifacts for robust detection, it inherits certain limitations from its reflectance-based assumptions. In scenarios with extreme lighting conditions, highly glossy or metallic materials, or strong image compression and filtering, the surface-level features required to estimate geometry and reflectivity can become less reliable. Nevertheless, the integration of learned visual features with physics-based modeling helps partially compensate for these cases, maintaining competitive performance even under challenging conditions.

Our work makes the following key contributions:

- We present the design and implementation of a novel physics-augmented framework, **Light2Lie**, for detecting deepfake images by leveraging the physical behavior of light reflection.
- We develop a base reflectance prediction network, making it learn the distinction of the base reflectivity value of around 0.96 for synthetic images and a value of around 0.04 for real images. This model is also trained with a margin-based constraint that improves class separability, as detailed in Section V.
- We introduce a physically grounded reflection score derived from microfacet-based light reflection theory. This score quantifies the realism of light behavior in an image and is integrated into a secondary classifier to provide physically interpretable guidance during deepfake detection.
- We introduce a fine-tuning loop into the training of the secondary classifier itself that uses classification errors to improve the base reflectivity estimation over time, increasing model resilience to edge cases and generative variance, as detailed in Section V.
- Lastly, we conduct extensive evaluations across multiple benchmark datasets, including both in-distribution and out-of-distribution deepfake sources to showcase the effectiveness of **Light2Lie**. Our evaluations show that **Light2Lie** outperforms state-of-the-art baselines, achieving up to 74.7% accuracy even on unseen deepfake sources, as detailed in Section VI.

II. Background and Preliminaries

This section introduces the physical components of this model, provides both mathematical and intuitive interpretations, and maps each concept to its use in our deepfake images detection pipeline.

A. Microfacet Theory and Surface Geometry

A realistic representation of materials often begins with modeling a surface as a composition of microfacets, microscopic planar elements (tiny flat surfaces) with varying orientations. While each microfacet behaves as an ideal mirror, the collective distribution of these facets gives rise to the macroscopic appearance of rough or glossy surfaces. The core assumption of the microfacet model is that light reflects off surfaces not as a perfect whole but from these individual facets.

Mathematically, for an incoming light direction E and outgoing (viewing) direction V, we define the halfway vector:

$$H = \frac{E + V}{\|E + V\|} \tag{1}$$

This halfway vector H represents the orientation of a microfacet that would reflect E into V under perfect mirror reflection. Each surface point has a normal vector N, and the relative orientation between N and H determines the orientation of such a microfacet.

For example, consider a surface as a field of microscopic mirrors. The more of them are oriented to reflect light directly to the viewer, the shinier the surface looks. In deepfake or AI-generated images, such micro-level accuracy is often compromised, i.e., the synthetic images look much smoother than their counterpart, human real images. Thus, it results in distinguishable physical inconsistencies. The microfacet model provides a grounded approach to estimate per-pixel geometry and predict reflection behavior. We leverage this to extract the specular reflectance patterns indicative of whether a surface is physically plausible.

B. Specular Reflection Metric

In the context of surface reflectance modeling, the specular reflection metric s is a crucial quantity that captures how light interacts with a surface to produce specular highlights. This metric originates from the microfacet-based Bidirectional Reflectance Distribution Function (BRDF), introduced by Blinn [4], and is used extensively in computer graphics to simulate realistic lighting effects. The metric is given by the expression:

$$s = \frac{DGF}{N \cdot V} \tag{2}$$

Here, D is the microfacet distribution function representing the orientation variance of surface microgeometry, G accounts for the geometric attenuation due to masking and shadowing, F is the Fresnel reflectance term describing how much light is reflected based on the angle of incidence, and $N \cdot V$ captures the cosine of the angle between the surface normal and the eye (or viewing) direction.

This formulation shows that a surface's appearance depends on its material properties, geometry, and the angles of viewing and illumination. In our framework, we infer these light–surface interactions directly from the image itself, i.e, they are image-derived, without requiring any external lighting or physical measurements. This makes the specular reflection metric s a meaningful property for authenticity: real surfaces produce physically consistent reflection patterns, while AI-generated images often fail to replicate them. By integrating s into our main classifier, the model learns to detect subtle, physically grounded inconsistencies between real and synthetic images. Next, we detail the different variables required to determine s.

Fresnel Reflectance Term F: The Fresnel reflection law (Fresnel term F) describes how light is partially reflected and partially transmitted (refracted) at the boundary between two different media (like air and glass), depending on the angle of incidence, and it's based on the base reflectivity F_0. Base reflectivity is the amount of light reflected at normal incidence, that is, when light hits a surface head-on (at a 0° angle). It's a material property that describes how reflective a surface is at the micro level, without considering the viewing angle or light angle.

To simplify computations, we use Schlick's approximation:

$$F = F_0 + (1 - F_0)(1 - (H \cdot V))^5 \tag{3}$$

where F_0 is the reflectance at normal incidence, and $H \cdot V$ is the cosine of the angle between the halfway vector and the viewing direction. In our application, we assume that synthetic images exhibit higher apparent reflectivity than real ones, a value of around 0.96 for synthetic images and a value of around 0.04 for real images. Thus, our first step in **Light2Lie** is to estimate and learn F_0 through a base neural network that distinguishes between realistic and unrealistic surface behaviors.

Microfacet Distribution Function D: The distribution term D models how microfacets are oriented relative to the surface normal. It influences the sharpness or spread of the specular highlight. A surface with low roughness has a peaked distribution (most microfacets point in a similar direction), while a rougher surface has a broader distribution. Specifically, this variable tells us how many microfacets are "tilted just right" to reflect light toward the viewer.

We compute D using the GGX distribution [4], where the surface roughness is estimated from the covariance of local gradients on the height map. This term helps capture the statistical regularity of real surfaces, which is often lacking in synthetic images.

Geometric Attenuation Term G: The geometric term G accounts for self-shadowing and masking effects between microfacets. Specifically, it accounts for how much of the microfacet reflection is visible to the observer or light source. Even if a microfacet is oriented toward the light source, it might be occluded by neighboring facets. That is microfacets may be obscured by others, either from the light source (shadowing) or from the view direction (masking). We use the Smith masking function (mentioned below) [4] to estimate this attenuation based on the angle between the surface normal and the viewing direction. This term reflects the complexity of real-world surfaces, where geometric interactions limit ideal reflectance. The geometric term $G(N, E, v)$ accounts for this by reducing the contribution of such facets.

One approximation is:

$$G(N, E, V) = \min\left(1, \frac{2(N \cdot H)(N \cdot V)}{V \cdot H}, \frac{2(N \cdot H)(N \cdot E)}{V \cdot H}\right) \tag{4}$$

In our framework, we simulate surface masking and shadowing from the image-derived normal map to calculate G, which contributes to estimating how much reflection is blocked.

Normal-View Alignment $N \cdot V$: This term represents the cosine of the angle between the surface normal N and the viewing direction V, serving as a normalization factor. It modulates how much of the reflected light is directed toward the observer. A perfect alignment $(N \cdot V = 1)$ indicates direct reflection, while misalignment reduces the perceived intensity. In deepfake images, where realistic geometry is often ill-defined, this term reveals inconsistencies in surface-view alignment.

Together, these components allow us to construct a physically interpretable score s for any given image, offering a robust mechanism for distinguishing between authentic and AI-generated visual content. By aligning our detection strategy with the fundamental behaviors of light, we gain generalizability across generation techniques.

III. Challenges

This section outlines the challenges of modeling a reflectance-based detection framework, which occur because we map light behavior to 2D images, especially those from generative models.

Representing Images as Microfacets: The microfacet model assumes a 3D surface made of tiny reflectors, while we only have 2D RGB images. We approximate this by creating a height map using local texture entropy and gradient strength, and then computing surface normals. However, pixels do not perfectly correspond to physical microfacets, and generative models may produce patterns with no real-world meaning, which can lead to errors in computing D, G, and F.

Estimating Surface Roughness (α): The distribution term D depends on surface roughness α, which we infer from gradients and edge density. It is hard to separate true surface roughness from the digital image textures, and fake images can mimic real image rough surfaces without a physical basis. This can reduce the reliability of D.

Fresnel Term (F_0) Estimation: The Fresnel term depends on the material's reflectance, which is unknown in images. We train a network to estimate base reflectance F_0, assigning low values to real and high values to fake images. If F_0 is inaccurate, the specular metric s becomes less meaningful.

Geometric Attenuation (G): Computing G needs light, view, and normal directions (E, V, N, H) are hard to estimate from a single 2D image, as we are not using any external lightning source to estimate them. Additionally, synthetic images can have multiple lights or soft shadows, adding uncertainty. Our model makes reasonable approximations and extracts surface statistics to model these light vectors, as described in Section V-B.

Robustness to Domain Shifts: Different generative models may synthesize images with drastically different styles, affecting color distribution, shading, and noise. The proposed pipeline must generalize across synthetic domains without hardcoding features of any particular generative architecture because overfitting to certain generators may result in brittle detection when facing novel synthesis techniques.

In summary, the use of reflectance physics for deepfake detection introduces unique challenges not seen in conventional learning systems.

IV. Threat Model

We consider a powerful adversary $\mathcal{A}$ whose goal is to generate or modify images to convincingly imitate realistic images. The adversary aims to ensure that the deepfake images are indistinguishable from real digital content, thereby evading detection mechanisms deployed by platforms or security systems.

To obtain these deepfake images, $\mathcal{A}$ can leverage any state-of-the-art image synthesis pipeline to either generate new images or enhance images with localized edits, using diffusion-based models (e.g., Stable Diffusion, DreamStudio, DALLE-2) or GAN-based models (e.g., StyleGAN, CIFAKE). We do not place any restrictions on the generation technique, nor do we assume any domain constraints on the type of content produced, allowing $\mathcal{A}$ to generate human faces, objects, or modify existing images to add new context.

The goals of $\mathcal{A}$ are twofold. First, the constructed images must appear semantically and visually coherent, i.e., consistent with human expectations and natural image statistics. Second, they must evade detection by appearing physically plausible, even under forensic scrutiny. This includes producing realistic lighting, shadows, and material textures that blend seamlessly into real-world contexts.

Attacker Knowledge and Capabilities. We adopt a black-box threat model with partial knowledge. It is assumed that $\mathcal{A}$ has knowledge of the overall design of our detection framework, including its physics-augmented pipeline, the use of base reflectance estimation, and the specular consistency metric s. However, the adversary does *not* have access to the trained model parameters or the exact training data.

Unlike a true white-box setting, the $\mathcal{A}$ cannot compute exact gradients through our deployed detector because, in real situations, the trained weights are not made accessible to the public. At best, the $\mathcal{A}$ can build a shadow model with his knowledge of the training algorithm and attempt to train it on a surrogate dataset to craft adaptive attacks. However, because our approach is grounded in physical light behavior, specifically, reflectance properties not explicitly modeled by generative methods, such adversarial images remain vulnerable to our detection method.

V. Design Details

In this section, we present **Light2Lie**, which detects deepfake images by approximating the physics of light reflection. The system has three modules: (1) learning base reflectance F_0, (2) training the main classifier $\mathcal{F}_s$, which computes the specular reflection score s and fine-tunes $\mathcal{F}_0^r$, and (3) inference. Figure 1 illustrates the overall workflow.

A. High-level Idea

Learning Base Reflectance F_0: We begin with analyzing the base reflectivity F_0 of the image, which in physical optics, refers to the proportion of light reflected at normal incidence on a surface. Inspired by this concept, we design a neural network $\mathcal{F}_0^r$ trained to predict F_0 based on an input image. This predicted value is essential to calculate the Fresnel term F, which plays a pivotal role in light reflection behavior. Since real-world surfaces and AI-generated ones differ significantly in their intrinsic material reflectance, we enforce a margin-aware loss that pushes F_0 values of AI-generated images toward the high-reflectance range (near 0.96) and real images toward the low-reflectance range (near 0.04). This helps in

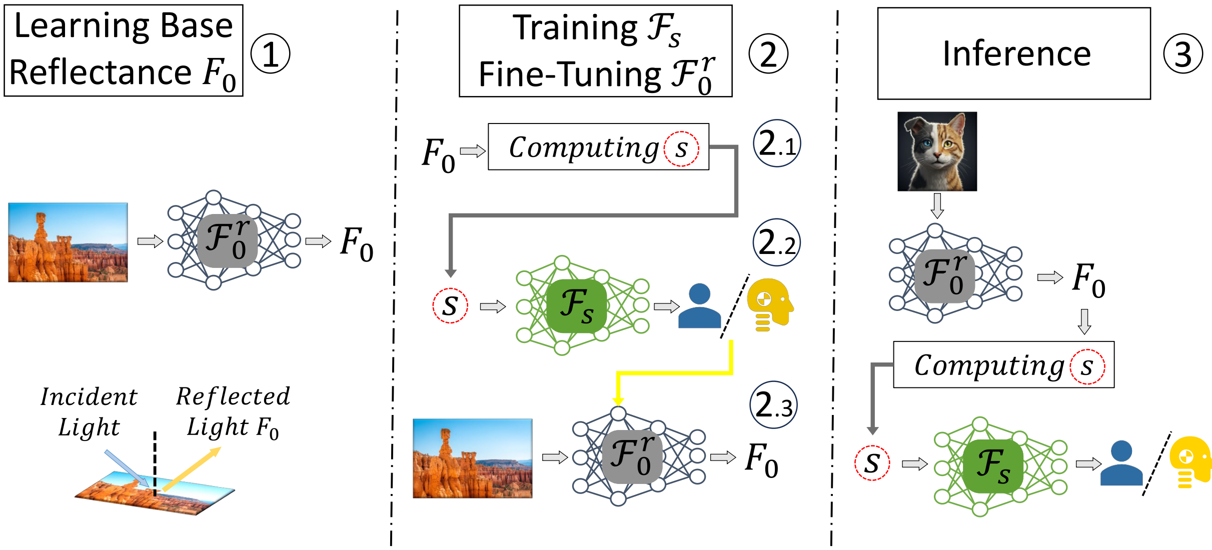


Fig. 1: Overview of **Light2Lie**. In step 1, we train a base reflectance model $\mathcal{F}_0^r$, in step 2, we compute s trough $\mathcal{F}_0$ (2.1), then, we use the Specular Reflection to train the main classifier $\mathcal{F}_s$ (2.2), and fine-tune $\mathcal{F}_0^r$ for (any) misclassifications (2.3). Lastly, in step 3, we utilize the full pipeline for inference to detect the class of any unknown input sample.

amplifying the separability between the two classes from a physical standpoint.

Training Main Classifier $\mathcal{F}_s$: After obtaining F_0, we model how light would interact with the image as if its pixels were tiny planar microfacets. Rather than using any external lighting setup, we derive the incident and reflected light directions (for each pixel) directly from the image itself, based on local surface geometry inferred from the height map and its gradients (defined below). This produces per-pixel light vectors and surface normals that capture how light would naturally reflect, be attenuated, or be shadowed across the surface. From these geometry-dependent interactions, we then compute the specular reflection score $s = \frac{DGF}{N \cdot V}$ ((Section II-B)), where D encodes local roughness, G accounts for shadowing and masking, F uses the Schlick approximation from F_0, and $N \cdot V$ captures the alignment between the surface and the viewing direction.

Once we compute the s score for each image, we integrate it into the training of a secondary or a main ML model $\mathcal{F}_s$ as an additional input feature. The training of this model is guided by the s metric to facilitate the learning process in physically grounded artifacts. This allows the model to focus on physical reflectance discrepancies and on the visual texture features. We also introduce a fine-tuning mechanism where, if the predicted F_0 appears inconsistent with the expected behavior (i.e., a real image predicted with a very high F_0 or vice versa), the F_0 network $\mathcal{F}_0^r$ is selectively retrained on those edge cases. This joint optimization improves robustness and adaptiveness over time.

Inference Phase: During inference, the trained base reflectance model computes F_0, which is then used to calculate the specular score s. This score, together with the image features, is fed into the trained ML model to classify the image as real or deepfake.

B. Design Structure

This section elaborates on the internal workings of our system by providing mathematical and algorithmic details of each component introduced in the high-level overview.

Embedding Generation: To obtain a compact and semantically rich representation of each image, we use the pre-trained EfficientNet-B7 model [49]. EfficientNet-B7 is a CNN that applies compound scaling to balance depth, width, and resolution for high accuracy and efficiency. It employs inverted residual blocks and Squeeze-and-Excitation (SE) [18] layers to enhance representational power by emphasizing informative features. Importantly, the final convolutional layers of EfficientNet-B7 produce semantically rich feature maps that capture both global and local patterns from the input image.

Learning Base Reflectance F_0: To learn F_0, we train a CNN model $\mathcal{F}_0^r$ that estimates this base reflectivity from input image embeddings E. The model $\mathcal{F}_0^r$ takes an input image embedding $z \in \mathbb{R}^{C \times H \times W}$ and predicts a scalar value $F_0 = \mathcal{F}_0^r(z)$. $\mathcal{F}_0^r$ is a CNN designed for binary classification on high-dimensional image embeddings z. The model has three convolutional layers with batch normalization and ReLU, two max-pooling layers, and two fully connected layers, ending with a sigmoid for probability output. It is trained with the Adam optimizer ($lr = 0.0001$) for 20 epochs using binary cross-entropy loss.

We map the true labels $y \in \{0, 1\}$ to target reflectance values: 0 is mapped to 0.04 (real) and 1 to 0.96 (deepfake), aligning with the physical intuition that deepfake images often appear smoother and more reflective. The network is trained with binary cross-entropy (BCE) to match predicted F_0 to these targets. Additionally, we introduce a margin-based regularization to enhance separation between the classes by penalizing predictions that lie near the decision boundary:

$$\mathcal{L}_{\text{margin}} = \mathbb{E}\left[y \cdot \max(0, 1 - F_0) + (1 - y) \cdot \max(0, F_0)\right]$$

The final loss combines both terms: $\mathcal{L}_{F_0} = \mathcal{L}_{\text{BCE}} + \lambda \mathcal{L}_{\text{margin}}$. This ensures $\mathcal{F}_0^r$ produces discriminative, physically plausible outputs for specular metric computation, with $\lambda = 0.1$ chosen empirically.

Computing Specular Reflection s**:** After obtaining F_0, we compute the specular reflection score s (using Equation 2) for each image embedding z. To compute this, we first need to simulate different light vectors, i.e., normal ($\mathbf{N}$) light ($\mathbf{E}$), view ($\mathbf{V}$), and halfway ($\mathbf{H}$). Steps are detailed below to compute them.

Light Vectors: First, we approximate z as a 3D surface by converting it to a grayscale height map $h(x, y)$ that incorporates three attributes: (1) local texture complexity via an entropy map, (2) edge-derived depth information via gradient magnitude, and (3) a geodesic-inspired local energy term that emphasizes regions with high neighborhood variance. We first form a base height map from normalized entropy and depth maps using a convex combination:

$$h_{\text{base}} = \gamma \cdot h_{\text{depth}} + (1 - \gamma) \cdot h_{\text{entropy}},$$

where γ balances depth and entropy contributions (set to 0.5 in our experiments). Next, we modulate this base height map with an energy term $E(x, y)$ derived from the local patch variance to highlight geometrically discriminative regions:

$$h(x, y) = h_{\text{base}}(x, y) \cdot \big(1 + \beta\, E(x, y)\big),$$

where β controls the influence of the energy modulation (0.3 in our implementation). Finally, $h(x, y)$ is normalized to $[0, 1]$ to produce the final height map used for gradient and normal computation.

Next, we compute horizontal and vertical gradients $\partial h/\partial x$ and $\partial h/\partial y$ using Sobel filters[20], which are edge detection operators used to approximate the first derivative in the image. These filters are particularly suitable for computing gradient maps due to their smoothing and differentiation properties.

Using these gradients, we construct surface normals as:

$$\mathbf{N}(x, y) = \frac{1}{\sqrt{(\partial h/\partial x)^2 + (\partial h/\partial y)^2 + 1}} \cdot [-\partial h/\partial x, -\partial h/\partial y, 1]$$

Next, we estimate the surface roughness parameter α (which will be used to determine D and G) by computing the eccentricity c_3 derived from the covariance matrix of the gradients. Thus, we assume $c_3 = \alpha$ in our framework. Specifically,

$$\mathbf{C} = \begin{bmatrix} \mathbb{E}[\partial_x^2] & \mathbb{E}[\partial_x \partial_y] \\ \mathbb{E}[\partial_x \partial_y] & \mathbb{E}[\partial_y^2] \end{bmatrix}$$

Let λ_1 and λ_2 be the ordered eigenvalues of $\mathbf{C}$ such that $\lambda_1 \geq \lambda_2$. The eccentricity is then defined as:

$$c_3 = \frac{\sqrt{\lambda_1}}{\sqrt{\lambda_2 + \epsilon}} \qquad \text{where } \epsilon \text{ ensures numerical stability}$$

This formulation quantifies the directional anisotropy in surface variation. A high c_3 indicates significant surface tilt along one direction, suggesting a rougher microfacet structure.

We synthesize light ($\mathbf{E}$), view ($\mathbf{V}$), and halfway ($\mathbf{H}$) vectors based on c_3 and the normal map. $\mathbf{E}$ and $\mathbf{V}$ are constructed with directional tilt and vertical components adjusted using c_3, which encodes the roughness, and then, the halfway vector is computed as the midway vector between the two using Equation 1.

These approximated vectors simulate how a viewer would observe reflected light under varying surface configurations. Next, each term in the specular score is computed as follows:

Microfacet Distribution D**:** The GGX (Trowbridge-Reitz) distribution (as mentioned in Section II-B) is employed to model the probability that a microfacet is aligned in a direction to reflect incoming light toward the observer. It is defined as:

$$D = \frac{\alpha^2}{\pi\left[(\cos\theta)^2(\alpha^2 - 1) + 1\right]^2}$$

where $\cos\theta = \mathbf{N} \cdot \mathbf{H}$ represents the alignment between the surface normal and halfway vector, and $\alpha = c_3$ is the roughness. This formulation ensures that sharper, smoother surfaces yield peaked distributions, while rough surfaces produce flatter ones.

Geometric Attenuation G**:** This term accounts for how microfacets may block each other's reflections due to surface occlusion and self-shadowing. We use the Smith approximation (detailed in Section II-B) for simplicity and efficiency:

$$G = \frac{2(\mathbf{N} \cdot \mathbf{V})}{(\mathbf{N} \cdot \mathbf{V}) + \sqrt{\alpha^2 + (1 - \alpha^2)(\mathbf{N} \cdot \mathbf{V})^2}}$$

This formulation penalizes light that gets blocked due to rougher or steeper facets, reflecting the true shadowing behavior of natural surfaces.

Fresnel Reflectance F**:** The Fresnel term describes how reflectance changes with the viewing angle. We use Schlick's approximation mentioned in Equation 4. This efficiently models the increase in reflectance at grazing angles, an effect widely observed in real-world materials. It relies on the predicted F_0 as base reflectance and provides angular sensitivity to the model.

Angular Alignment $N \cdot V$ This final term accounts for how aligned the viewer is with the surface normal: $N \cdot V = \max(\mathbf{N} \cdot \mathbf{V}, \epsilon)$. This prevents numerical instability and encodes viewing geometry dependence, which is essential for computing physically accurate reflection intensity.

Finally, we compute s as the mean reflection intensity across all pixels using all the relevant variables discussed above.

Physics-Augmented ML Training: The computed score s is passed into a secondary classifier $\mathcal{F}_s$, which combines learned visual features with s to predict the final class label. Formally, $\hat{y} = \mathcal{F}_s(Z, s)$.

$\mathcal{F}_s$ is a dual-branch CNN that combines visual embeddings with physics-based features. The image branch uses five 1×1 convolutions to reduce 2560 input channels to 32, followed by flattening and four fully connected layers to produce a 128-D vector. The physics branch processes the specular map s with two convolutions, batch normalization, and adaptive average pooling to yield a 16-D vector.

The two branches are concatenated and passed through two additional fully connected layers for final classification. The model is trained using binary cross-entropy loss, optimized with Adam ($lr = 1e-4$), and trained over 15 epochs. Finally, the model is trained using: $\mathcal{L}_s = \text{BCE}(\hat{y}, y)$.

This design leverages both deep features and physics-based cues. Lastly, for any misclassified sample, i.e., if $\hat{y}$ is incorrect, we fine-tune $\mathcal{F}_0^r$, reinforcing physical alignment and improving generalization. Algorithm 1 summarizes the working of **Light2Lie**.

Inference Phase: During inference, given an image embedding z, we first compute $F_0 = \mathcal{F}_0^r$, then, derive the reflection score s, and use $\mathcal{F}_s(z, s)$ to predict its label. Because each step mimics light reflection in physical space, our system outputs accurate and generalizable decisions resilient to distribution shifts.

VI. Evaluation

In this section, we evaluate **Light2Lie** across multiple settings. We first analyze why models struggle with out-of-distribution (OOD) data (Section VI-D) and examine the predictive power of **Light2Lie** against various generation frameworks (Section VI-E). We then test generalization to unseen models (Section VI-F), analyze failure cases of **Light2Lie** (Section VI-G), and assess robustness to adaptive adversarial attacks (Section VI-H). Finally, we present an ablation study of each component of **Light2Lie** (Section VI-I).

A. Experimental Setup

All experiments were conducted using the PyTorch [19] framework on a server equipped with 4 NVIDIA RTX 8000 GPUs (each with 48GB memory), an AMD EPYC 7742 CPU, and 1TB of system memory.

We curated a benchmark covering a diverse collection of AI-generated image sources to ensure a comprehensive and realistic evaluation. Our evaluation includes images produced by both diffusion-based models (DALL·E 2 [35], Stable Diffusion [40], DreamStudio [48]) and GAN-based architectures (StyleGAN [21], CIFAKE [60]), reflecting the diversity of modern image generation pipelines. The ground truth for real samples was extracted from the ARIA and LAION Datasets [25], [46]. This setup allows us to test both in-domain and out-of-distribution generalization, capturing the performance of **Light2Lie** on sources it was not explicitly trained on.

Algorithm 1 Physics-Informed Deepfake Detection

```
Initialize trained reflectance predictor $\mathcal{F}_0^r$ and classifier $\mathcal{F}_s$
Set margin threshold $\tau$, learning rates $\eta_1$, $\eta_2$, and regularization weight $\lambda$
for each epoch do
    for each batch $(z, y)$ do
        // Step 1: Predict base reflectance
        $F_0 \leftarrow \mathcal{F}_0^r(z)$
        // Step 2: Compute specular reflection score $s$
        Compute height map $h(x, y)$
        Compute image gradients $\partial_x h$, $\partial_y h$ using Sobel filters
        Estimate surface normals $\mathbf{n}$ and covariance matrix $\mathbf{C}$
        Compute roughness $\alpha \leftarrow c_3 = \sqrt{\lambda_1}/\sqrt{\lambda_2 + \epsilon}$
        Simulate light $\mathbf{E}$, view $\mathbf{V}$, and halfway vector $\mathbf{H}$
        Compute components $D$, $G$, $F$, and angular alignment $N \cdot V$
        $s \leftarrow \mathbb{E}_{x,y}\left[\frac{DGF}{N \cdot V}\right]$
        // Step 3: Physics-Augmented ML Training
        $\hat{y} \leftarrow \mathcal{F}_s(z, s)$
        Compute loss $\mathcal{L}_s \leftarrow \text{BCE}(\hat{y}, y)$
        Update classifier: $\mathcal{F}_s \leftarrow \mathcal{F}_s - \eta_1 \cdot \nabla \mathcal{L}_s$
        // Step 4: Fine-tune $\mathcal{F}_0^r$ on difficult samples
        Identify hard examples: $M \leftarrow \{i \mid (y_i = 1 \wedge F_{0,i} < \tau) \vee (y_i = 0 \wedge F_{0,i} > 1 - \tau)\}$
        if $M \neq \emptyset$ then
            $I_{\text{hard}} \leftarrow z[M]$, $y_{\text{hard}} \leftarrow y[M]$
            $F_0^{\text{hard}} \leftarrow \mathcal{F}_0^r(I_{\text{hard}})$
            Define ground-truth $F_0$: $y_{\text{mapped}} \leftarrow 0.96$ if $y = 1$, else $0.04$
            Compute loss: $\mathcal{L}_{F_0} \leftarrow \text{BCE}(F_0^{\text{hard}}, y_{\text{mapped}}) + \lambda \cdot \mathcal{L}_{\text{margin}}$
            Update: $\mathcal{F}_0^r \leftarrow \mathcal{F}_0^r - \eta_2 \cdot \nabla \mathcal{L}_{F_0}$
        end if
    end for
end for
```

Model	Params (M)	Size (MB)	Infer. (ms/img)	Peak Mem (MB)
$\mathcal{F}_0$	0.35	1.34	0.54	718.44
$\mathcal{F}_s$	29.75	113.49	2.21	722.70

TABLE I: Computational overhead metrics of $\mathcal{F}_0$ and $\mathcal{F}_s$.

Additionally, we also evaluate the computational overhead of **Light2Lie** to demonstrate its suitability for practical deployment. Table I reports the number of parameters, model size, per-image inference time, and peak GPU memory usage for both $\mathcal{F}_0$ and $\mathcal{F}_s$. Despite incorporating physics-based modeling, the framework remains lightweight, with $\mathcal{F}_0$ requiring less than 1 MB and $\mathcal{F}_s$ under 120 MB, and total inference time below 3 ms per image on a single GPU.

Next, we evaluate core detection methods, comparing accuracy and generalization on datasets ranging from photorealistic to stylized images, testing our method's ability to detect reflectance-based inconsistencies beyond simple statistical features.

B. Dataset Description

Table II summarizes the dataset with samples from multiple deepfake generators, each using a different synthesis approach. Below, we briefly describe each system.

a) ARIA Dataset [25]: The AdversaRIal AI-Art (ARIA) dataset is a large-scale benchmark for AI-generated media. We use 17,040 real samples across five categories: artworks, social media, news, disaster, and anime. Each sample includes a caption and a textual description usable as a generation prompt, and most evaluated deepfake datasets follow the same five-category distribution.

b) LAION [46]: We utilized a subset of the LAION-5B dataset consisting solely of images depicting humans. LAION is a large-scale, open dataset of image-text pairs collected from the web. For our study, we filtered the dataset using semantic similarity scores and metadata of the captions to isolate human-centric images.

c) DALL·E 2 [35]: DALL·E 2 is a diffusion-based model that generates images from natural language prompts. It first maps text to a latent representation using a pretrained CLIP model [33], and then iteratively refines a noise vector through a diffusion process to produce a high-fidelity image.

d) Stable Diffusion [40]: Stable Diffusion relies on a latent diffusion model. It encodes images into a lower-dimensional latent space using a variational autoencoder (VAE), performs the diffusion process in this compact space, and then decodes the latent result back into the pixel space. Prompts guide the generation via cross-attention with a text encoder.

e) DreamStudio [48]: DreamStudio is a commercial interface for Stable Diffusion developed by Stability AI. It allows users to generate images by customizing various parameters such as prompt strength, number of inference steps, and image resolution. The underlying generation process is similar to Stable Diffusion.

f) StyleGAN [21]: StyleGAN is a GAN-based architecture that generates images by mapping a latent code through a style-based synthesis network. The network introduces styles at each convolutional layer, enabling control over visual features at different scales.

g) CIFAKE [60]: CIFAKE is a synthetic dataset created using GANs, designed specifically to evaluate fake image detection. It employs a modified GAN pipeline to generate images that resemble those in the CIFAR-10 dataset. The goal is to produce realistic fakes that challenge detection models while preserving class consistency.

TABLE II: Dataset statistics for GAN-based, Diffusion-based approaches, and Genuine images.

	Generation Approach	Number of Samples
Diffusion	DALL·E 2 [35]	27,072
	Stable Diffusion [40]	50,048
	DreamStudio [48]	32,768
GAN	StyleGAN [21]	7,040
	CIFAKE [60]	60,096
Real	ARIA [25]	17,040
	LAION [46]	6,358

C. Evaluation Metrics

Following prior work in deepfake image detection [47], [31], we use a comprehensive set of evaluation metrics to assess the effectiveness of our proposed framework. These metrics reflect both detection performance and robustness under varying conditions:

- **True Positive Rate (TPR):** measures the model's ability to correctly identify deepfake (AI-generated) images. It is defined as the proportion of correctly detected fake samples (True Positives, TP) among all actual fake samples: $\text{TPR} = \frac{\text{TP}}{\text{TP+FN}}$.
- **True Negative Rate (TNR)**, also referred to as specificity, quantifies the model's ability to correctly recognize authentic (human-generated) images. It is calculated as the ratio of correctly identified real images (True Negatives, TN) to all actual real images: $\text{TNR} = \frac{\text{TN}}{\text{TN+FP}}$.
- **F1-Score** is the harmonic mean of precision and recall, offering a balanced measure of detection performance when both false positives and false negatives are of concern. Here, *precision* is defined as the proportion of predicted fake images that are fake: $\text{Precision} = \frac{\text{TP}}{\text{TP+FP}}$, and *recall* is the same as TPR: $\text{Recall} = \frac{\text{TP}}{\text{TP+FN}}$. The F1-score combines both as: $\text{F1-Score} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision+Recall}}$. It ranges from 0 (worst) to 1 (best), with higher values indicating better detection quality.

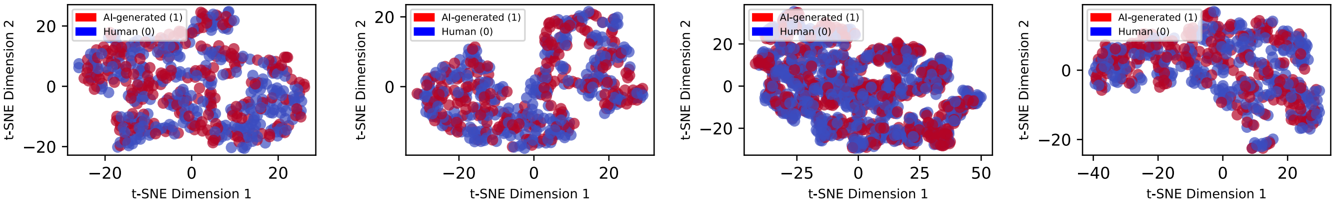
These metrics enable a thorough evaluation of both in-distribution detection accuracy and generalization to out-of-distribution deepfake sources.

D. Data Analysis

This section evaluates why models struggle to generalize to out-of-distribution (OOD) data, especially when low-level features of real and fake images are visually similar.

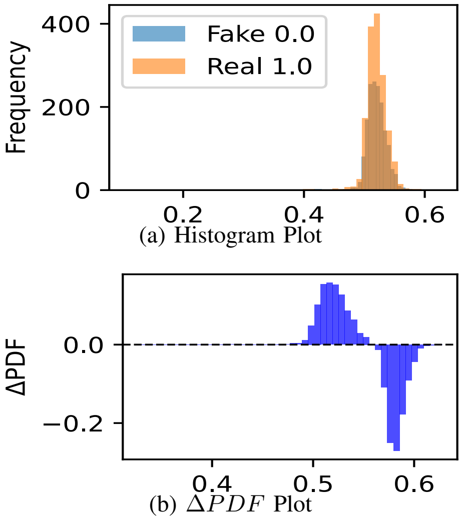
We conduct a t-SNE [51] feature projection across four generative models: DALL·E 2 [35], StyleGAN [21], DreamStudio [48], and Stable Diffusion [40]. We analyze three scalar descriptors of each embedding: *spectral entropy*, which measures frequency-domain randomness via the normalized Fourier power spectrum (low values indicate smooth structure, high values indicate rich or noisy content); *kurtosis*, which reflects the "tailedness" of the pixel distribution, where high values suggest sparse or peaky activations and low values indicate flatter, Gaussian-like behavior; and *sparsity*, the proportion of near-zero elements, capturing the efficiency of feature activations, as generative models often produce more uniformly filled embeddings than real images.

The three features are then concatenated into a $\mathbf{R}^3$ vector per image and visualized with t-SNE, which projects them



(a) DALLE (b) DreamStudio (c) Stable Diffusion (d) StyleGAN

Fig. 2: t-SNE projection of handcrafted features (spectral entropy, kurtosis, sparsity) from image embeddings. Points represent images: red = AI-generated, blue = human. The two classes largely overlap in feature space.



(a) Histogram Plot

(b) ΔPDF Plot

Fig. 3: Comparison of real vs. AI-generated feature distributions. (a) Histogram of mean height map values; (b) ΔPDF highlighting separable regions between real and synthetic samples.

to 2D while preserving local similarity. As shown in Fig. 2, red points (AI-generated) and blue points (human) overlap heavily across all datasets, with no clear separation. This illustrates why simple statistical descriptors cannot reliably distinguish human from AI images, motivating the physics-aware, structure-based s metric used in **Light2Lie**.

E. Dataset Evaluations

First, we illustrate the distribution of the observed height maps, normal maps, s maps, and F_0 values obtained by **Light2Lie** after training the classifiers $\mathcal{F}_0^r$ and $\mathcal{F}_s$. Since the plots are similar for all the datasets, for this analysis, we opted StyleGAN dataset to showcase the trend in the aforementioned values. Then, we detail the efficacy of **Light2Lie** in detecting the fake samples vs the real samples in terms of TPR, TNR, and F1-Score metrics. Table III shows its detection accuracy and robustness across different datasets.

TABLE III: Effectiveness of **Light2Lie** for each individual dataset. All values are in percentages.

Dataset	TPR	TNR	F1-Score
CIFAKE	96.58	98.57	98.78
DALLE-2	95.63	96.48	96.91
StyleGAN (SG)	97.68	97.92	97.43
Stable Diffusion (SD)	98.61	97.68	98.11
DreamStudio	92.49	91.15	92.89
Combined (SD & DALLE)	97.91	97.06	98.03
Combined (SD & DALLE $ SG)	97.87	96.61	90.37
Combined (SD & DALLE & SG & LAION)	98.00	97.49	97.39

Illustration of Height map values: Figure 3a shows that real images occupy a narrow range of mean height values, while deepfakes span a broader distribution that includes the real range. This indicates that generative models often produce exaggerated or irregular local surfaces, yielding wider height map variations. The $\Delta \mathrm{PDF}$ in Figure 3b confirms this pattern: positive peaks mark regions dominated by real images, negative valleys indicate fake-dominated regions, and real samples remain concentrated within a narrow interval.

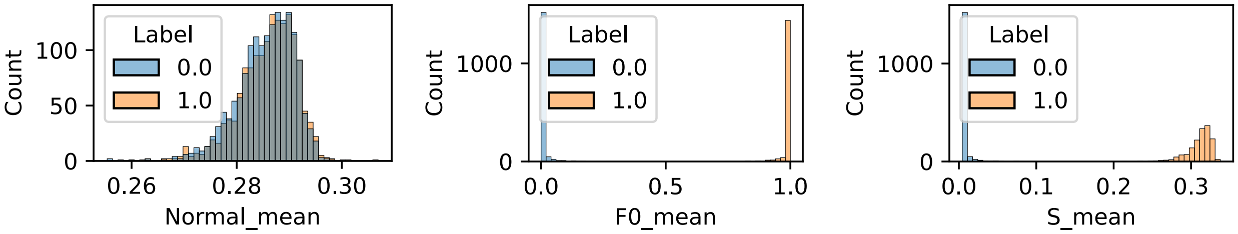
Illustration of Normal values: Figure 4a shows that real and fake images have largely overlapping mean normal magnitudes, with fakes showing a slightly broader spread from local surface irregularities. While this captures some smoothness differences, normal magnitude alone is less discriminative than reflectance-based features.

Illustration of F_0 values: Figure 4b shows clear separation in estimated F_0 values: real images (label 0) cluster at lower reflectance, while fakes (label 1) concentrate at higher values. This confirms that the F_0 module in **Light2Lie** captures meaningful surface reflectance differences between real and generated samples.

Illustration of S-map values: Figure 4c shows the distribution of s values from **Light2Lie**. Fake samples cluster around 0.3, while real images lie mostly below 0.04, with a few misclassified exceptions. This clear separation demonstrates the effectiveness of the reflectance model $\mathcal{F}_s$ in capturing discriminative physical properties.

Next, we illustrate the performance of **Light2Lie** on the datasets mentioned in Table II.

CIFAKE Results: The CIFAKE dataset serves as a benchmark for foundational evaluation, comprising simple, low-resolution AI-generated and real images. Our method achieves a high True Positive Rate (TPR) of 96.58%, and a True Negative Rate (TNR) of 98.57%, resulting in an F1-score of 0.98. These metrics affirm that even at low



(a) (b) (c)

Fig. 4: Distributions of physics-based features for real and AI-generated images: a) normal map (local surface orientation), b) base reflectance F_0, and c) mean s-map. Together, they reveal clear separable patterns between the two classes.

resolution, **Light2Lie** is highly effective at differentiating between authentic and synthetic content.

DALLE-2 Results: Images generated by DALLE-2 are semantically coherent and visually appealing, often closely resembling human art. Despite this complexity, **Light2Lie** achieves a TPR of 95.63% and a TNR of 96.48%, demonstrating robustness to high-quality generative outputs. The F1-score of 0.96 proves strong classification performance, even when image content becomes semantically rich and stylized.

Stable Diffusion Results: Stable Diffusion synthesizes complex images with an emphasis on coherence and fidelity. Yet, **Light2Lie** maintains a TPR of 98.61% and a TNR of 97.68%, yielding a strong F1-score of 0.98. This demonstrates that **Light2Lie** can reliably detect diffusion-based generation artifacts, even when image quality is visually indistinguishable from real content.

DreamStudio Results: DreamStudio consists of a diverse range of AI-generated images across various themes and styles. In this evaluation, **Light2Lie** achieves a TPR of 92.49% and a TNR of 91.15%, resulting in an F1-Score of 0.92. These results indicate that **Light2Lie** performs robustly despite the variability in DreamStudio's content.

Combined Dataset Results: We also test **Light2Lie** on combined datasets to evaluate its ability to discern beyond single-source data. The Stable Diffusion and DALLE-2 combination yields a robust TPR of 97.91% and an F1-Score of 0.98, confirming cross-model generalizability. As more generators are included (e.g., SD, DALLE, SG, and Human), the performance remains consistently high with an F1-score of 0.99, validating the method's scalability and consistency across broader, more diverse distributions.

F. Generalized Evaluation

This section compares **Light2Lie** with state-of-the-art deepfake detectors (listed below), using their open-sourced pretrained models. Below, we provide some brief details about the inner workings of these methods:

TABLE IV: Comparison of **Light2Lie** with DE-FAKE [47] and Universal Fake Image Detector [31] on the OOD DreamStudio's test partition [48]. Using DE-FAKE and UFID pretrained model checkpoints. All values in percentage.

Method	Metric	Art	Disaster	Instagram	News	Pixiv
DE-FAKE [47]	TPR	30.14	59.08	26.00	57.85	61.88
	TNR	45.40	28.27	56.73	29.14	37.71
	F1	32.91	50.31	30.30	49.56	54.65
UFID [31]	TPR	15.30	49.90	11.10	15.80	7.50
	TNR	96.10	83.50	86.20	97.60	87.30
	F1	25.70	59.64	17.67	26.70	12.43
AIDE [59]	TPR	12.21	18.73	10.97	30.39	11.21
	TNR	99.96	97.87	99.66	97.79	97.84
	F1	21.04	30.97	19.71	45.78	19.76
ObjFormer [53]	TPR	24.36	34.86	45.30	52.21	58.30
	TNR	62.81	47.33	43.07	42.25	38.72
	F1	30.36	36.60	44.14	48.83	52.56
GANDCT [10]	TPR	27.31	36.55	0.36	23.47	62.48
	TNR	66.37	68.0	99.93	95.86	40.81
	F1	34.14	43.02	0.71	36.70	55.83
Qwen [54]	TPR	47.39	12.04	17.46	24.66	98.88
	TNR	94.82	98.67	95.00	97.53	2.09
	F1	62.18	21.23	28.46	38.74	65.85
Llama [15]	TPR	1.98	3.33	4.88	1.3	67.65
	TNR	100.00	81.25	89.61	100.00	67.67
	F1	3.88	5.88	8.51	2.56	58.60
Light2Lie	TPR	**74.77**	**67.16**	**44.91**	**19.86**	**46.66**
	TNR	**49.09**	**57.78**	**81.40**	**95.28**	**65.42**
	F1	**56.40**	**60.60**	**68.80**	**69.00**	**59.80**

- **DE-FAKE [47]:** DE-FAKE was developed to tackle text-to-image deepfake generation, using three key components: a plain machine learning classifier, a fingerprint detection approach for generative models, and an image-text encoder (CLIP [33]). The ensemble approach was evaluated by its authors on the generators: Stable Diffusion, Latent Diffusion, GLIDE, and DALLE-2.
- **Universal Fake Image Detector (UFID) [31]:** UFID detects fake images by performing real-vs-fake classification in the feature space of a large pretrained vision-language model (CLIP-ViT), rather than training a dedicated classifier.
- **AIDE [59]:** Detects AI-generated images by selecting extreme-frequency patches via DCT, extracting noise with

SRM filters, and combining them with semantic features from OpenCLIP.

- **Objectformer [53]:** A multimodal transformer for image manipulation detection that fuses RGB and high-frequency features into patch embeddings and uses learnable object prototypes with cross-attention to capture object-level inconsistencies.
- **GANDCT [10]:** GANDCT uses the discrete cosine transform (DCT) to expose up-sampling artifacts in GAN images, which appear consistently across architectures and datasets. Compared to pixel-domain classifiers, it achieves higher accuracy with fewer parameters and less training data.
- **Llama-3.2-11B-Vision-Instruct [15]:** This vision language model is fine-tuned for image reasoning, captioning, and visual QA. Its vision encoder uses a two-stage design (32-layer local + 8-layer global) with gated attention and concatenated features (7680-D), improving reasoning at the cost of higher computation. Raza et al. [37] reported 74.82% accuracy with the base Llama-3.2-11B-Vision; here we test whether the instruction-tuned version improves deepfake detection.
- **Qwen2.5-VL-72B-Instruct [54]:** This model achieves performance comparable to GPT-4o and Claude 3.5 Sonnet on multimodal benchmarks and outperforms many general vision-language models. It uses a streamlined Vision Transformer with window attention, SwiGLU, and RMSNorm for efficient, high-fidelity processing. Ren et al. [39] showed the 7B version struggles on deepfake benchmarks; thus, we evaluate the flagship Qwen2.5-VL-72B model.

Inference on Pre-trained Models: To evaluate existing detectors, we conducted an inference-only assessment of DE-FAKE [47], UFID [31], AIDE [59], ObjectFormer [53], GANDCT [10], and two vision-language models (VLMs): Qwen2.5-VL [54] and Llama-3.2 [15], using their pretrained model checkpoints on the DreamStudio dataset. This dataset was selected because it was not included in the training of any of those checkpoints, providing a fair basis to assess generalization.

As summarized in Table IV, all prior detectors exhibit significant performance drops in out-of-distribution (OOD) scenarios. DE-FAKE achieves moderate TPRs but suffers from low TNRs on *Disaster* and *News*, indicating frequent misclassification of real images. UFID achieves high TNRs but very low TPRs, often labeling fake images as real unless they resemble training data. AIDE and the VLMs show extremely unbalanced performance, with either near-zero TPRs or very low F1 scores. ObjectFormer and GANDCT achieve only partial improvements, with inconsistent trade-offs between TPR and TNR across categories.

Compared to these approaches, **Light2Lie** achieves a more balanced performance across all categories, achieving higher TPRs on the *Art* and *Disaster* partitions, along with competitive F1 scores across all categories. While **Light2Lie** 's TNRs are moderate, this balance results in consistently higher F1 scores, reflecting an improved trade-off between precision and recall compared to existing detectors.

To provide a more complete evaluation, we also fine-tuned these detection models on a train partition of the DreamStudio dataset. These results, reported in App. A, show similar trends and confirm **Light2Lie** 's robustness.

G. Error Analysis

To evaluate **Light2Lie**'s brittleness and generalization error on extreme OOD cases, we evaluate three DreamStudio categories: *art*, *disaster*, and *news*. Figure 5 shows the s-score distributions, explaining its lower accuracy on *news* compared to the other two.

For both the *art* (Figure 5a) and the *disaster* (Figure 5b) category, real images (label 0) and fake images (label 1) occupy either the low s region or high s region, thus, maintaining a clear separation. However, the *news* (Figure 5c) category displays the smallest separation between real and fake distributions, with both concentrated at lower s-scores. This behavior is primarily due to strong compression, downsampling, and professional post-processing of news imagery, which reduces the local microfacet variations that our reflectance-based model exploits.

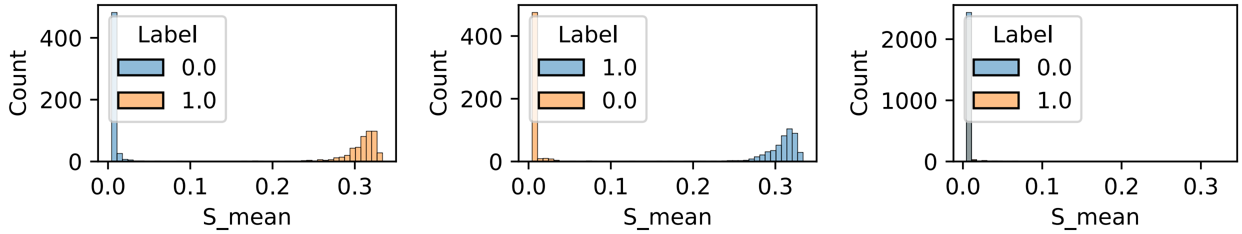
Thus, these observations show that while **Light2Lie** generalizes well, separation decreases in low-reflectance domains, motivating adaptive thresholds or lightweight feature fusion for more robust performance.

H. Adversarial Robustness

In addition to generalization under distribution shifts, deepfake detectors must remain robust to adaptive attacks, where adversaries craft images that fool models while appearing realistic. We evaluate **Light2Lie** against three adaptive attacks employing different perturbation strategies: (1) surface-feature attacks (FGSM [13], PGD [27]), (2) light-consistent prompts, and (3) localized edits. We assume an adversary with full architectural knowledge but no access to trained weights (Sec. IV).

Surface Feature Adversarial Perturbation. We evaluated FGSM and PGD attacks with $\epsilon = 0.01$ on StyleGAN and DALLE-2 samples using our model trained on the combined samples from StyleGAN, DALLE-2, and Stable Diffusion. The model showed strong robustness, maintaining 100% TPR on both datasets. Under FGSM, F1 Scores were 0.6869 (StyleGAN) and 0.7256 (DALLE-2), and under PGD, 0.6850 and 0.7181, respectively.

Light-Consistent Prompts. To conduct adaptive attacks with the goal of generating fake samples using the light consistent prompts, we generated 1,000 synthetic images from 1,000 real StyleGAN face images using Stable Diffusion v1.5 [42]. We employed light-consistent prompts, e.g., “A hyper-realistic image with soft studio lighting, natural shadows”, to maintain relevant illumination. Evaluation with our CIFAKE-pretrained model, we achieved F1 = 0.9820 and TPR = 96.47%, highlighting strong resilience.



(a)

(b)

(c)

Fig. 5: Distribution of s for DreamStudio datasets: a) Art, b) Disaster, and c) News.

Localised Edits. Localized perturbations or adversarial images were generated with FLUX.1-Redux-dev by prompting FLUX.1-Redux-dev [23] with real images, using YOLO [38] detected object regions as masks, and replacing them with objects to simulate realistic localized changes while preserving overall scene consistency. Under this adaptive attack, our CIFAKE-pretrained model achieved F1 = 0.8941 and TPR = 80.84%, revealing some susceptibility to highly localized manipulations while maintaining strong overall robustness.

In summary, embedding real-world optical constraints enables our model to achieve strong generalization and robust adversarial resistance, supporting real-world deployment of **Light2Lie**.

I. Ablation Study of Light2Lie Components

To evaluate how each component contributes to **Light2Lie**, we perform an ablation study isolating physics-based features (F_0 and s), semantic embeddings from EfficientNet-B7, and their combination for the final model in **Light2Lie**.

First, we modify $\mathcal{F}_s$ to include two parallel branches: (i) a visual branch using EfficientNet-B7 embeddings and (ii) a physics branch computing the s-score from predicted F_0. The model operates in three modes by selectively enabling or zeroing branches to measure standalone performance.

In the **Embedding-only mode**, $\mathcal{F}_s$ only uses the visual embedding branch to process image embeddings, with the physics branch fully disabled. No microfacet-based features (height maps, normal maps, F_0, or s) are used. This setup reflects performance using only high-level semantic and texture features from EfficientNet-B7, without any physical priors.

In the **Physics-only mode**, the embedding branch is disabled, and $\mathcal{F}_s$ uses the physics-derived reflectance features. We consider two variants: the scalar base reflectance F_0 and the full s-score from the microfacet model using D, G, and Fresnel terms. This mode evaluates how well physical features alone distinguish real from fake images without semantic context.

The **Full Light2Lie mode** uses both embedding and physics branches jointly. Their feature vectors—visual embeddings and s-score-based physics features—are concatenated and passed to the classifier, leveraging complementary data-driven and physics-informed cues for robust detection.

TABLE V: Ablation study on StyleGAN dataset using pretrained F_0 and $\mathcal{F}_s$ models. The results report Accuracy and F1-score for visual-only (EfficientNet-B7 features), physics-only (F_0 and s maps), and the full combined **Light2Lie** model.

Model Variant	**Accuracy**	**F1-score**
Visual-only	0.8993	0.8961
Physics-F_0 only	0.9807	0.9813
Physics-s-score only	0.9807	0.9813
Full Light2Lie	0.9727	0.9734

We use the StyleGAN dataset with pretrained F_0 and $\mathcal{F}_s$ models. Table V reports Accuracy and F1-score for each ablation mode, from which we draw the following observations:

Physics-only models (F_0 and s-score) achieve the strongest standalone performance, showing that **Light2Lie** effectively leverages specular reflectance and surface geometry to distinguish real from fake images. The embedding-only model performs moderately, confirming that semantic or texture cues alone are insufficient. The full **Light2Lie** model is robust but slightly lower than physics-only, likely due to minor feature redundancy. Overall, these results support that **Light2Lie** relies on physical characteristics rather than dataset-specific biases.

VII. Security Discussions

We evaluate the robustness of **Light2Lie** against a strong white-box adversary $\mathcal{A}$ who understands the detection pipeline, including its physics-based components F_0 and s, but lacks access to both model's weights or training data. Given this, $\mathcal{A}$ can mount adaptive attacks using modern generation tools and simulate detection using a locally trained shadow model.

To make the system fail, $\mathcal{A}$ can deploy multiple evasion strategies. First, the adversary may generate samples using a diverse suite of generative models (e.g., Stable Diffusion, StyleGAN, DALLE-2) and tune them to maximize semantic realism and visual coherence. However, these generative models are not physically grounded and do not model reflectance or light behavior. As a result, their outputs, even if perceptually convincing, fail to conform to the optical consistency patterns exploited by our detection framework.

Second, $\mathcal{A}$ may launch direct adversarial perturbations using gradient-based methods such as FGSM and PGD. These attacks attempt to subtly alter the pixel space to minimize the likelihood of detection, all while staying imperceptible to the human eye. However, because **Light2Lie** relies on physically interpretable features such as F_0 and s, these small pixel-level perturbations do not fundamentally alter the global light interaction properties. Our empirical findings confirm that **Light2Lie** remains robust under such attacks, consistently achieving high detection rates.

Finally, $\mathcal{A}$ may attempt to circumvent the detector through domain transfer-training on synthetic styles or categories that fall outside the distribution used to train **Light2Lie**. However, **Light2Lie** generalizes well across unseen generation styles and content types, owing to its grounding in the physical laws rather than specific dataset artifacts. As a result, even in open-world settings where attackers use previously unseen styles or prompt variations, the model continues to detect inconsistencies arising from non-physical generation.

In summary, **Light2Lie** remains robust against a range of adaptive strategies an adversary may realistically deploy.

VIII. Related Works

A. *Frequency-Domain Analysis*

Early forensic methods and several modern detectors have explored the frequency domain to reveal manipulation artifacts. Frank et al. [10] proposed GANDCT, a statistical learning method which uses handcrafted DCT-based frequency features to identify GAN-generated images. Guo et al. [16] presented two statistical fake colorization detectors, one of which (FCID-FE) uses frequency-domain features encoded via Fisher vectors. Sabitha et al. [44] proposed EMFID, a hybrid model that integrates discrete wavelet transform (DWT) on CNNs. Martín-Rodríguez et al. [28] fused traditional forensic signals such as Photo Response Non-Uniformity (PRNU) and Error Level Analysis (ELA) with deep CNNs. Ojha et al. [31] analyzed the frequency spectra of various fake image distributions, showing that GAN-generated images exhibit distinct periodic patterns, while diffusion models do not. Similarly, Zhao et al. [61] introduced a deepfake detection framework with a textural feature enhancement module focused on shallow-layer features, which are sensitive to frequency-level discrepancies.

More recently, Yan et al. [59] proposed AIDE, which selects image patches with extreme frequency content using a DCT-based scoring module, extracts noise patterns with SRM filters, and fuses these with semantic embeddings from OpenCLIP. This hybrid design leverages both fine-grained artifacts and high-level contextual cues to improve detection robustness.

B. *Spatial and Statistical Features*

Before the deep learning era, traditional forensic methods relied on pixel-level inconsistencies, residual statistics, and hand-crafted features. Techniques like resampling detection [32], JPEG quantization and double compression analysis [2], [11], and scene lighting inconsistency detection [30] formed the backbone of early digital image forensics. More recent approaches include JPEG dimple detection [1], co-occurrence-based tampering localization [5], and histogram-based color statistics [16]. With the emergence of deep learning, particularly Convolutional Neural Networks (CNNs), image forensics has shifted from handcrafted feature extraction and statistical methods to data-driven representation learning. Rao and Ni [36] introduced one of the earliest CNN-based models where the first layer was initialized with spatial-rich model (SRM) high-pass filters to emphasize manipulation traces while suppressing content. Cozzolino et al. [6] proposed ForensicTransfer, an autoencoder that operates on high-pass residuals and separates latent subspaces for real and fake distributions, improving generalization under weak supervision. Rössler et al. [43] released the FaceForensics++ dataset and baseline CNNs based on XceptionNet, which detect subtle pixel-space manipulation patterns across multiple face-editing techniques. Martín-Rodríguez et al. [28] also rely heavily on spatial forensic cues.

These methods, while effective under specific settings, often show limited robustness to unseen generative models or sophisticated forgeries due to overfitting to dataset-specific statistics.

C. *Transformer-Based and Ensemble Classifiers*

To improve generalization and modeling capacity, recent works have adopted transformer-based architectures or hybrid ensembles. Weir et al. [57] proposed a CNN/ViT+Attention ensemble model, integrating local CNN features with ViT [8] representations and attention modules. Trained on CIFAKE, their model achieved 99.77% accuracy on Stable Diffusion-generated images, significantly outperforming individual CNN or ViT variants. Sha et al. [47] proposed DE-FAKE, a hybrid CLIP-based detector and attributor for fake images generated by text-to-image models. Their approach fuses image and prompt embeddings, enabling both detection and source attribution with generalization across models like Stable Diffusion, DALLE-2, and GLIDE. Wang et al. [53] introduced ObjectFormer, a multimodal transformer that combines RGB and high-frequency features with learnable object prototypes. Using cross-attention, it models object-level consistency and refines patch embeddings, enabling effective detection and localization of manipulated regions.

IX. Conclusion

In this work, we presented **Light2Lie**, a physics-augmented deepfake detection framework that leverages the fundamental principles of light reflection to distinguish between real and AI-generated images. By estimating a physically meaningful specular reflectance score, we introduced a novel way to incorporate real-world optical artifacts into the training and inference of a deep learning model. Our experiments demonstrate that **Light2Lie** achieves strong performance, particularly in generalizing to unseen generative models and resisting adversarial perturbations.

Acknowledgment

Our research work was partially funded by DFG-SFB 1119-236615297, the Horizon program of the European Union under the grant agreements: 101070537-CrossCon and

101093126-ACES, NSF-DFG-Grant 538883423, the European Research Council under the ERC Programme-Grant 101055025-HYDRANOS, as well as the Federal Ministry of Education and Research of Germany (BMBF) within the IoTGuard project.

References

[1] Shruti Agarwal and Hany Farid. Photo forensics from jpeg dimples. In *2017 IEEE workshop on information forensics and security (WIFS)*, pages 1–6. IEEE, 2017.

[2] Sebastiano Battiato and Giuseppe Messina. Digital forgery estimation into dct domain: a critical analysis. In *Proceedings of the First ACM workshop on Multimedia in forensics*, pages 37–42, 2009.

[3] BBC News. Ukraine war: Deepfake video of zelenskyy could be 'tip of the iceberg'. https://www.bbc.com/news/technology-60780142, 2022.

[4] James F Blinn. Models of light reflection for computer synthesized pictures. In *Proceedings of the 4th annual conference on Computer graphics and interactive techniques*, pages 192–198, 1977.

[5] Davide Cozzolino, Giovanni Poggi, and Luisa Verdoliva. Splicebuster: A new blind image splicing detector. In *2015 IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 1–6. IEEE, 2015.

[6] Davide Cozzolino, Justus Thies, Andreas Rössler, Christian Riess, Matthias Nießner, and Luisa Verdoliva. Forensictransfer: Weakly-supervised domain adaptation for forgery detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[7] Haoran Dong, Xiaofeng Liu, Jie Zhang, Hao Wang, and Wei Wu. Mvlm: Multimodal vision-language model for cross-domain deepfake detection. *arXiv preprint arXiv:2308.04615*, 2023.

[8] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[9] Ricard Durall, Margret Keuper, and Janis Keuper. Watch your up-convolution: Spectral artifacts in gans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[10] Jonas Frank, Thorsten Eisenhofer, Adam Kortylewski, Christian Riess, Dorothea Kolossa, and Volker Fischer. Leveraging frequency analysis for deep fake image recognition. In *International Conference on Machine Learning (ICML)*, 2020.

[11] Fausto Galvan, Giovanni Puglisi, Arcangelo Ranieri Bruna, and Sebastiano Battiato. First quantization matrix estimation from double compressed jpeg images. *IEEE Transactions on Information Forensics and Security*, 9(8):1299–1310, 2014.

[12] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2014.

[13] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations (ICLR)*, 2015.

[14] Diego Gragnaniello, Gabriele Poggi, and Luisa Verdoliva. Detecting gan-generated imagery using color cues. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2021.

[15] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[16] Yuanfang Guo, Xiaochun Cao, Wei Zhang, and Rui Wang. Fake colorized image detection. *IEEE Transactions on Information Forensics and Security*, 13(8):1932–1944, 2018.

[17] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[18] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018.

[19] Sagar Imambi, Kolla Bhanu Prakash, and GR Kanagachidambaresan. Pytorch. *Programming with TensorFlow: Solution for Edge Computing Applications*, pages 87–104, 2021.

[20] Nikolaos Kanopoulos, Narendra Vasanthavada, and Robert L Baker. Design of an image edge detection filter using the sobel operator. *IEEE Journal of Solid-State Circuits*, 23(2):358–367, 1988.

[21] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2019.

[22] Tero Karras, Samuli Laine, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[23] Black Forest Labs, Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, et al. Flux. 1 kontext: Flow matching for in-context image generation and editing in latent space. *arXiv preprint arXiv:2506.15742*, 2025.

[24] Yuezun Li, Ming-Ching Chang, and Siwei Lyu. In ictu oculi: Exposing ai created fake videos by detecting eye blinking. *arXiv preprint arXiv:1806.02877*, 2018.

[25] Yuying Li, Zeyan Liu, Junyi Zhao, Liangqin Ren, Fengjun Li, Jiebo Luo, and Bo Luo. The adversarial ai-art: Understanding, generation, detection, and benchmarking. In *European Symposium on Research in Computer Security*, pages 311–331. Springer, 2024.

[26] Yabin Liu, Xiao Zhang, Xiaochun Cao, and Jun Zhu. Global and local consistency learning for detecting image manipulation. In *European Conference on Computer Vision (ECCV)*, 2020.

[27] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations (ICLR)*, 2018.

[28] Fernando Martin-Rodriguez, Rocio Garcia-Mojon, and Monica Fernandez-Barciela. Detection of ai-created images using pixel-wise feature extraction and convolutional neural networks. *Sensors*, 23(22):9037, 2023.

[29] Yisroel Mirsky and Wenke Lee. The creation and detection of deepfakes: A survey. *ACM Computing Surveys (CSUR)*, 54(1):1–41, 2021.

[30] James F O'brien and Hany Farid. Exposing photo manipulation with inconsistent reflections. *ACM Trans. Graph.*, 31(1):4–1, 2012.

[31] Utkarsh Ojha, Yuheng Li, and Yong Jae Lee. Towards universal fake image detectors that generalize across generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24480–24489, 2023.

[32] Alin C Popescu and Hany Farid. Exposing digital forgeries by detecting traces of resampling. *IEEE Transactions on signal processing*, 53(2):758–767, 2005.

[33] Alec Radford, Jong Wook Kim, Luke Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020*, 2021.

[34] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022.

[35] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022.

[36] Yuan Rao and Jiangqun Ni. A deep learning approach to detection of splicing and copy-move forgeries in images. In *2016 IEEE international workshop on information forensics and security (WIFS)*, pages 1–6. IEEE, 2016.

[37] Shaina Raza, Ashmal Vayani, Aditya Jain, Aravind Narayanan, Vahid Reza Khazaie, Syed Raza Bashir, Elham Dolatabadi, Gias Uddin, Christos Emmanouilidis, Rizwan Qureshi, et al. Vldbench evaluating multimodal disinformation with regulatory alignment. *arXiv preprint arXiv:2502.11361*, 2025.

[38] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[39] Simiao Ren, Yao Yao, Kidus Zewde, Zisheng Liang, Ning-Yau Cheng, Xiaoou Zhan, Qinzhe Liu, Yifei Chen, Hengwei Xu, et al. Can multimodal (reasoning) llms work as deepfake detectors? *arXiv preprint arXiv:2503.20084*, 2025.

[40] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion

models. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022.

[41] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[42] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10684–10695, June 2022.

[43] Andreas Rossler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Nießner. Faceforensics++: Learning to detect manipulated facial images. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1–11, 2019.

[44] R Sabitha, A Aruna, S Karthik, and J Shanthini. Enhanced model for fake image detection (emfid) using convolutional neural networks with histogram and wavelet based feature extractions. *Pattern Recognition Letters*, 152:195–201, 2021.

[45] Chitwan Saharia, William Chang, Jonathan Ho, Tim Salimans, David Fleet, Phillip Isola, and Ben Poole. Imagen: Photorealistic text-to-image diffusion models. In *International Conference on Machine Learning (ICML)*, 2022.

[46] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon Drozdov, Andreas J Damgaard, Sigurdur Sigurdsson, Timo Assaf, Benjamin Kees, Philipp Schmitt, Jelena Cvarkova, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *arXiv preprint arXiv:2210.08402*, 2022.

[47] Zeyang Sha, Zheng Li, Ning Yu, and Yang Zhang. De-fake: Detection and attribution of fake images generated by text-to-image generation models. In *Proceedings of the 2023 ACM SIGSAC conference on computer and communications security*, pages 3418–3432, 2023.

[48] Stability AI. Dreamstudio. https://dreamstudio.ai, 2022.

[49] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019.

[50] The New York Times. The deepfake dilemma. https://www.nytimes.com/2021/11/21/technology/deepfake-videos.html, 2021.

[51] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

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

[53] Junke Wang, Zuxuan Wu, Jingjing Chen, Xintong Han, Abhinav Shrivastava, Ser-Nam Lim, and Yu-Gang Jiang. Objectformer for image manipulation detection and localization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2364–2373, 2022.

[54] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024.

[55] Sheng-Yu Wang, Oliver Wang, Andrew Owens, and Alexei A Efros. Exposing gan-generated faces using inconsistencies in facial landmarks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[56] Yuhao Wang, Chao Zhang, Zhi Lin, Yunchao Xu, and Yongzhen Ding. Detecting deepfakes with vision-language models. *arXiv preprint arXiv:2304.02414*, 2023.

[57] Stuart Weir, Muhammad Shahbaz Khan, Naghmeh Moradpoor, and Jawad Ahmad. Enhancing ai-generated image detection with a novel approach and comparative analysis. In *2024 17th International Conference on Security of Information and Networks (SIN)*, pages 1–7. IEEE, 2024.

[58] Mikael Westerlund. The emergence of deepfake technology: A review. *Journal of Strategic Security*, 12(2):19–36, 2019.

[59] Shilin Yan, Ouxiang Li, Jiayin Cai, Yanbin Hao, Xiaolong Jiang, Yao Hu, and Weidi Xie. A sanity check for ai-generated image detection. *arXiv preprint arXiv:2406.19435*, 2024.

[60] Yang Zhang, Wei Sun, Qingqing Du, and Zhenhua Liu. Cifake: A synthetic image dataset for deepfake detection. *arXiv preprint arXiv:2108.06294*, 2021.

[61] Hanqing Zhao, Wenbo Zhou, Dongdong Chen, Tianyi Wei, Weiming Zhang, and Nenghai Yu. Multi-attentional deepfake detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2185–2194, 2021.

[62] Peng Zhou, Xiaoyun Han, Vlad I Morariu, and Larry S Davis. Two-stream neural networks for tampered face detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

Appendix

A. *Generalization performance*

Inference after Fine-Tuning: We considered the case where the models were first fine-tuned on a train partition of the DreamStudio dataset before evaluating their performance on the same test partition used in Table IV.

Since vision-language models (VLMs) require paired image-text data for fine-tuning, we could not fine-tune Qwen2.5-VL [54] and Llama-3.2 [15], as the DreamStudio dataset only contains some caption information used to generate the fake sample; while the real data lacks accompanying textual descriptions.

As presented in Table VI, fine-tuning DE-FAKE [47] resulted in a shift where the model greatly increased its claim of real, therefore greatly increasing the TNR, which produced an imbalance that is visible on the F1 score.

UFID [31] benefited from fine-tuning, albeit to a lesser extent. While its TNRs remained high, the TPRs saw modest gains, with the highest being 44.60% in the Disaster category. The limited improvement in TPRs suggests that UFID's architecture may be less flexible in adapting to new data, potentially due to its initial design or training constraints.

AIDE [59] also showed improvements after fine-tuning, achieving balanced gains across all categories with F1 scores of 74.13% on Art and 71.54% on Disaster.

ObjectFormer [53] exhibited smaller but consistent improvements, such as 58.81% on Pixiv and 60.78% on News. But it maintained an overall low F1 score value.

Lastly, GANDCT [10] was unable to correctly differentiate between the overall features of real and generated images. Resulting and high TPR but very low TNR, suggesting that the model is classifying most of the sample as fake.

Notably, or proposed method **Light2Lie**, demonstrated superior performance even after fine-tuning. It achieved the highest TPRs across all categories, with values such as 90.40% in Art and 88.24% in Instagram. Although its TNRs were slightly lower than DE-FAKE's, they remained robust, contributing to the highest F1 scores among all methods. This indicates that **Light2Lie** not only effectively identifies fake images but also maintains a commendable rate of correctly recognizing real images, ensuring a balanced and reliable detection system.

In summary, while fine-tuning enhances the performance of existing detectors, **Light2Lie** consistently outperforms both DE-FAKE and UFID in detecting AI-generated images across diverse categories.

TABLE VI: Comparison of **Light2Lie** with DE-FAKE [47] and Universal Fake Image Detector [31] on the OOD DreamStudio's test partition [48]. All approaches fine-tuned on DreamStudio's training data. All values in percentage.

Method	**Metric**	**Art**	**Disaster**	**Instagram**	**News**	**Pixiv**
DE-FAKE [47]	TPR	50.04	57.46	40.63	56.82	49.25
	TNR	96.96	95.87	92.97	90.24	82.73
	F1	65.41	71.02	65.76	72.06	52.43
UFID [31]	TPR	27.20	44.60	31.20	27.00	19.60
	TNR	92.20	93.30	97.70	98.10	98.20
	F1	40.57	58.81	46.70	41.85	32.27
AIDE [59]	TPR	75.21	72.02	61.72	64.16	59.04
	TNR	72.31	70.67	75.93	68.21	65.60
	F1	74.13	71.54	63.05	65.48	61.04
ObjFormer [53]	TPR	51.68	55.11	56.23	59.56	62.24
	TNR	57.87	50.11	60.33	61.72	55.69
	F1	54.11	51.82	58.42	60.78	58.81
GANDCT [10]	TPR	94.67	92.89	11.54	98.73	95.81
	TNR	13.27	22.90	37.25	12.56	17.35
	F1	65.17	62.72	12.99	64.90	63.99
Light2Lie	TPR	**90.40**	**90.20**	**88.24**	**87.56**	**83.36**
	TNR	**87.84**	**77.68**	**71.40**	**67.00**	**64.48**
	F1	**89.00**	**82.87**	**77.96**	**74.68**	**71.20**