The DeLight-ReLight Benchmark

Detecting AI-Generated Video via 3DGS Reconstruction and Lighting-Based Physics-Imperfection Analysis

Maxime Buck — 3 February 2026

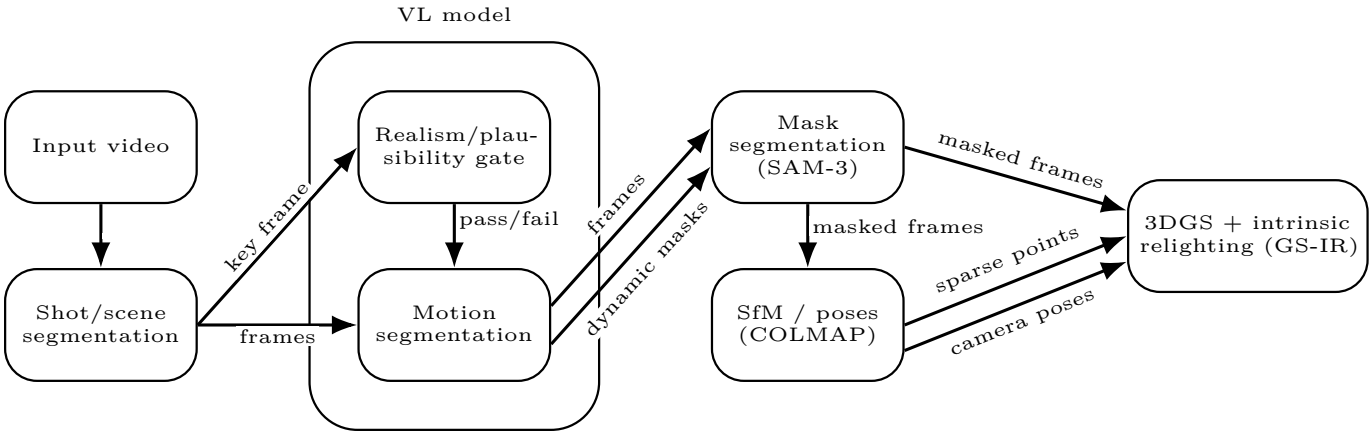

Figure 1: End-to-end video processing pipeline overview.

Abstract

As generative video models achieve exceptional degrees of fidelity, the demarcation between authentic and AI media is evaporating, placing visual forensics at a critical inflection point. While modern generators render convincing textures, they often violate the physical constraints of 3D projection and illumination. This work proposes a physics-grounded detector that operates on videos by first standardizing the input (scene segmentation, frame sampling, dynamic tracking, and camera/scene reconstruction) and then verifying geometric and photometric consistency. The method outputs three scores: a geometric score (S_{geo}) from robust bundle adjustment, a photometric score (S_{photo}) capturing texture stability, and a lighting score (S_{light}) measuring illumination coherence.

1 Introduction

Motivation. Fake news, non-consensual deepfakes, and scams are harmful consequences of the inability to discern AI-generated images/videos from real content. This urgency has escalated in the last year as synthetic content has become imperceptible to the human eye. Indeed, a large cross-country study from 2023 found that participants were worse at detecting fakes than guessing [1]. Moreover, exposure is increasing, with up to 33% of Shorts shown to a brand-new YouTube account being AI-generated [2], while the IAB reports that 40% of all ads will be AI-generated by 2026 [3]. In short, action is no longer optional, as trust in visual media will collapse without verifiable provenance and labeling of synthetic media.

Detector limitations. Current detection methods face fundamental limitations and can be reverse-fine-tuned. Some detectors, such as fingerprint-based GAN detectors [4] and generator-inversion methods [5], require knowledge of the generator. Other approaches, including CNN-based deep learning classifiers [6] and feature-space detectors [7], rely on artifacts/noise learned statistically during training. As a result, these models are dataset-dependent, and their performance drops substantially on unseen generators. Furthermore, retraining for each new generator is costly and leads to a perpetual cat-and-mouse game.

Proposed method. A complementary direction is semantic/reasoning-based detection, which seeks inconsistencies with physical laws or common sense. In other words, whether the content "makes sense" in the real world. This paper focuses specifically on physics-based violations because they depend on scene semantics rather than low-level texture or noise artifacts, and therefore generalize better across generators. Prior work has leveraged 3D reconstruction to expose geometric inconsistencies [8]. However, as generators improve, purely geometric cues may become less reliable. Therefore, this paper investigates a more rigorous constraint: physically accurate illumination and lighting consistency.

2 Problem Setup and Notation

Input. A RGB video $V = \{I_t\}_{t=1}^{T}$, with each frame $I_t \in \mathbb{R}^{H \times W \times 3}$.

Output. The goal is to estimate a probability score $y \in [0, 1]$, where larger values indicate higher confidence that the clip is AI-generated (and smaller values indicate higher confidence that it is real).

3 Method

3.1 Overview

Given an input video clip, the proposed detector first runs the processing pipeline in Figure 1. The outputs are then analyzed via a set of physics/consistency tests; the aggregated features feed the final scoring/decision step (Section 3.7).

3.2 Shot/Scene Segmentation.

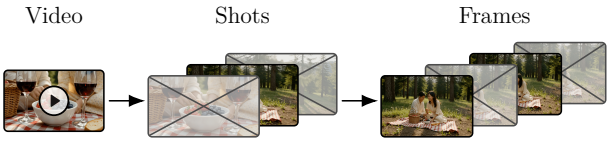

Figure 2: Shot/scene segmentation overview.

The method takes as input a video clip of arbitrary length and format. Frames are sampled at a rate adapted to the apparent camera motion (faster motion → higher sampling rate) and downsampled in resolution to reduce downstream compute, at the cost of fine detail.

To handle hard cuts and scene changes, the clip is segmented into shots (here using PySceneDetect [9]). This step is necessary because abrupt viewpoint discontinuities break the multi-view assumptions used by COLMAP and subsequent 3DGS reconstruction.

The output of this stage is the longest shot-consistent frame sequence from the set of shots, with approximately continuous camera motion (static or dynamic).

3.3 Vision-Language Classification.

The goal of this stage is to provide a low-cost compatibility check, and identify scene elements that should be excluded from geometric reconstruction.

Motivation. Classical SfM/MVS pipelines such as COLMAP [10, 11] (and the subsequent 3DGS representation [12]) assume that a large fraction of feature correspondences originate from a rigid, static scene. Independently moving objects (e.g., pedestrians, vehicles, articulated motion), temporal overlays

(e.g., subtitles, watermarks), and ill-conditioned regions (e.g., sky) generate inconsistent correspondences across frames, which degrade the pose estimation.

Related work. Moving-object detection under camera motion is an active research area; MONA [13], for example, proposes a dedicated framework. Since an open-source implementation was not available at the time and no comparable open-source alternative could be identified, this work introduces a novel method.

Realism/plausibility gate. Given the shot-consistent frame sequence (Section 3.2), a small set of representative keyframes (e.g., at the beginning/middle/end) is selected. The VLM is queried to assess photorealism and compatibility with multi-view reconstruction. The module outputs a continuation flag that enables early termination for clearly incompatible shots, evading unnecessary computation.

Structured scene inventory. For shots that pass the initial check, a VLM (Qwen3-VL Instruct, 8B variant [14]) is queried using a constrained prompt that requests a machine-readable JSON description of all entities and an "exclude" decision for each entity indicating whether it should be masked. The prompt explicitly requests exclusion of (a) objects moving independently of the camera, (b) text/logo overlays, and (c) non-informative background regions. The output of this stage is a JSON specification of regions to be removed (masked).

3.4 Mask Segmentation (SAM 3)

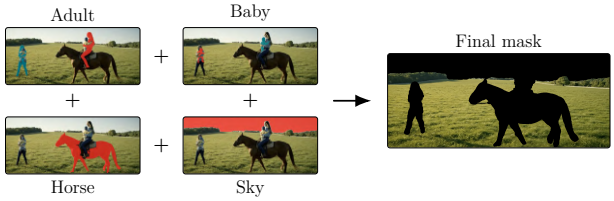

Figure 3: Ignore-mask is the union of per-entity masks

Given the per-shot removal specification, per-frame binary masks are produced using a promptable segmentation model from the Segment Anything family (SAM 3 [15]). For each frame, SAM 3 is prompted to segment the objects/regions flagged by the VLM, and the resulting instance masks are unioned into a single ignore-mask. These masks are then used to suppress masked pixels/features in downstream matching and reconstruction (see Figure 3).

Figure 4: Mask dilation with zoom-in callouts.

Without post-processing, SAM 3 may leave thin residual pixel traces at object boundaries, visible in Figure 4. While typically not fatal for downstream COLMAP/3DGS, these artifacts can introduce wrong features and unnecessary computation. Each binary mask is therefore inflated using morphological dilation to obtain a conservative ignore region.

Concretely, dilation is defined as the Minkowski sum of the mask set with a disk-shaped structuring element of radius r (the outline thickness). The resulting dilated mask is then used as the newly updated ignore-mask filter.

3.5 SfM Pose Estimation (COLMAP)

Given the shot-consistent and masked frame sequence, camera intrinsics/extrinsics and a sparse 3D point cloud are estimated using a standard Structure-from-Motion (SfM) pipeline implemented in COLMAP [10, 11]. This stage serves two roles: (i) it provides an explicit geometric explanation of the video in terms of a single rigid scene observed by a moving camera, and (ii) it exposes common SfM failure/degeneracy modes (e.g., low-parallax motion,

poor two-view geometry consistency, or unstable/ill-conditioned self-calibration).

Camera model and intrinsic calibration. SfM requires a camera projection model. A pinhole camera with radial distortion is assumed (as in COLMAP's standard camera models), with intrinsics $\mathbf{K}$ parameterized by focal lengths and principal point, plus distortion coefficients. Since the true capture device and metadata are unavailable (and meaningless for synthetic footage), intrinsics are treated as unknown and are estimated jointly with poses by minimizing reprojection error. Concretely, for a 3D point $\mathbf{X}_j \in \mathbb{R}^3$ observed in frame i, the predicted pixel location is

$$\hat{\mathbf{x}}_{ij} = \pi(\mathbf{K}, \boldsymbol{\theta}, \mathbf{R}_i, \mathbf{t}_i, \mathbf{X}_j)$$

where $(\mathbf{R}_i, \mathbf{t}_i)$ is the camera extrinsic (world-to-camera) transform for frame i, $\boldsymbol{\theta}$ denotes distortion parameters, and $\pi(\cdot)$ is the projection function.

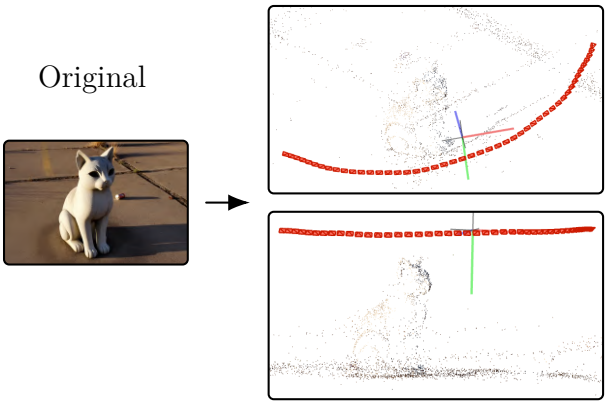

Figure 5: COLMAP example: sparse point cloud with red camera placements.

Feature extraction. For each frame, local keypoints and descriptors are extracted (COLMAP's default is SIFT [16]). Matching is then performed across temporally adjacent frames and (optionally) longer-range pairs to increase baseline while maintaining overlap. The computed ignore-masks are applied by discarding features whose pixel locations fall inside masked regions. From matches between frame pairs, COLMAP estimates relative pose via epipolar geometry with RANSAC [17]. Starting from an initial pair, it incrementally registers additional frames by triangulating 3D points from multi-view tracks and estimating each new camera pose, while adding new points as more views become available.

Bundle adjustment/refinement The full set of camera parameters $\{\mathbf{R}_i, \mathbf{t}_i\}$, intrinsics $(\mathbf{K}, \boldsymbol{\theta})$, and 3D points $\{\mathbf{X}_j\}$ is refined via bundle adjustment (BA), minimizing a robustified sum of reprojection errors:

$$\min_{\mathbf{K},\boldsymbol{\theta},\{\mathbf{R}_i,\mathbf{t}_i\},\{\mathbf{X}_j\}} \sum_{(i,j)\in\mathcal{O}} \rho\left(\|\mathbf{x}_{ij} - \pi(\mathbf{K}, \boldsymbol{\theta}, \mathbf{R}_i, \mathbf{t}_i, \mathbf{X}_j)\|_2^2\right)$$

where $\rho(\cdot)$ is a robust loss function. Here, $\mathcal{O}$ is the set of observed feature-track measurements. The BA outcome provides (a) refined per-frame poses, (b) an estimated focal length/distortion that acts as a proxy for "camera settings", and (c) diagnostic residual statistics.

3.6 3D Gaussian Splatting (RTR-GS)

Given estimated camera intrinsics/extrinsics from SfM, an explicit radiance-field representation based on 3D Gaussian Splatting (3DGS) is fitted [12], using the RTR-GS [18]. This stage is directly relevant to the proposed detector because it answers: can the input frames be explained by a single, stable 3D scene that renders consistently under small viewpoint changes?

Representation and differentiable rendering. 3DGS represents a scene as a set of N anisotropic 3D Gaussian primitives with parameters $(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \alpha_k, \mathbf{c}_k)$: mean $\boldsymbol{\mu}_k \in \mathbb{R}^3$, covariance $\boldsymbol{\Sigma}_k$ (stored via scale–rotation), opacity $\alpha_k \in (0, 1)$, and view-dependent appearance coefficients $\mathbf{c}_k$ (e.g., spherical harmonics) [12]. For a training view i, each 3D Gaussian is projected to an elliptical 2D footprint and rasterized via alpha compositing in approximate depth order, producing a rendered image $\hat{\mathbf{I}}_i$ (and auxiliary buffers such as approximate depth/visibility), with gradients back to all parameters [12].

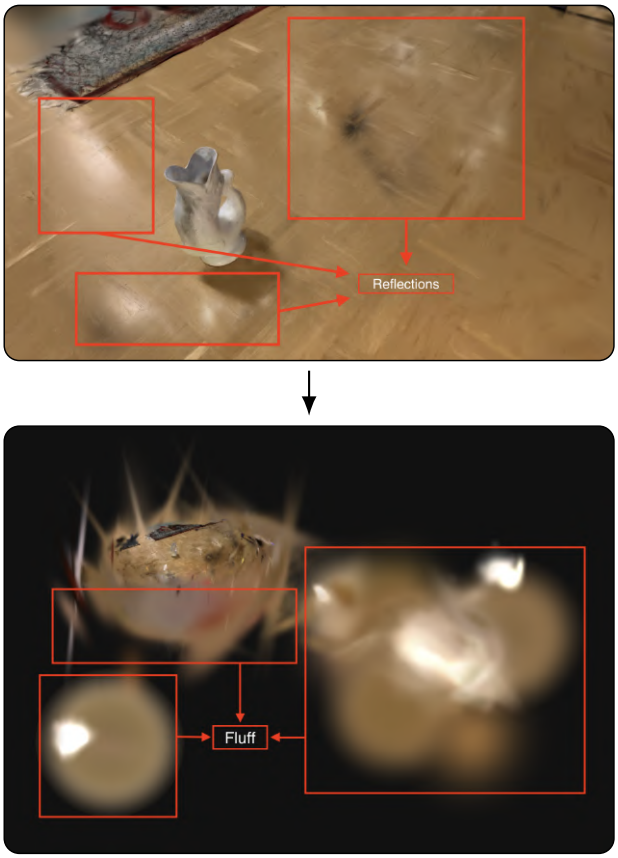

Figure 6: Reflections and "fluff" artifacts from a 3DGS visualization.

Initialization from SfM. Gaussian means are initialized from the sparse SfM point cloud so that optimization begins near plausible surfaces rather than searching in empty space [12]. If SfM is unreliable (non-rigid content, inconsistent illumination, synthetic artifacts), the sparse points tend to be noisy; in practice this often induces characteristic 3DGS failure modes (slow/unstable convergence, "floaters", aggressive densification), which later become features for scoring.

Iterative fitting: gradient steps. Training alternates over iterations: sample a view i, render $\hat{\mathbf{I}}_i$, and update Gaussian parameters by backpropagation to minimize a photometric reconstruction objective (in the baseline 3DGS setup, commonly a weighted combination of ℓ_1 and SSIM) [12]. Early iterations typically explain coarse structure (large, smooth Gaussians), while later iterations refine edges and high-frequency details by adapting positions, covariances, opacities, and appearance coefficients.

Iterative fitting/density control A fixed number of Gaussians is insufficient to represent fine structure everywhere. 3DGS therefore interleaves optimization with discrete density-control operations [12]:

- **Densification:** Gaussians with large accumulated gradients are duplicated/split to add local degrees of freedom where the current model underfits (often at edges/thin structures).
- **Pruning:** Gaussians that remain nearly transparent or contribute negligibly are removed to control size and suppress unstable "floaters".

For detection, the key observation is that coherent real shots tend to densify toward surfaces and then stabilize, while inconsistent shots can exhibit runaway splitting, oscillatory prune caused by regrow cycles, or persistent transparent clutter.

RTR-GS: inverse rendering with radiance transfer and reflection. [18] The baseline 3DGS model can absorb complex view-dependent phenomena into the appearance coefficients, sometimes at the expense of geometry. RTR-GS introduces an inverse-rendering motivated objective that explicitly models radiance transfer and reflection via a hybrid rendering formulation, aiming to better separate transport from reflection and thus produce more stable geometry in reflective regions [18].

Key outputs of RTR-GS. RTR-GS produces (i) material attributes (PBR maps) such as albedo (base color without shading), metallic and roughness maps; (ii) geometric structures including per-pixel surface normals (enabling physically plausible relighting) and the optimized set of 3D Gaussian primitives representing scene geometry; (iii) lighting estimates for

the capture conditions, often parameterized as environmental illumination; and (iv) visual renderings, including relighted images rendered under novel illumination (e.g., transforming a daylight scene into a sunset appearance).

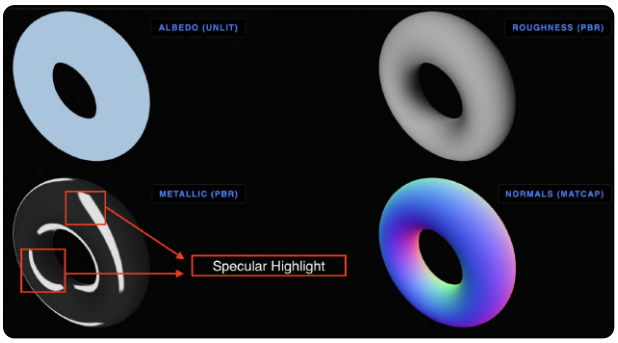

Figure 7: Materials visualization.

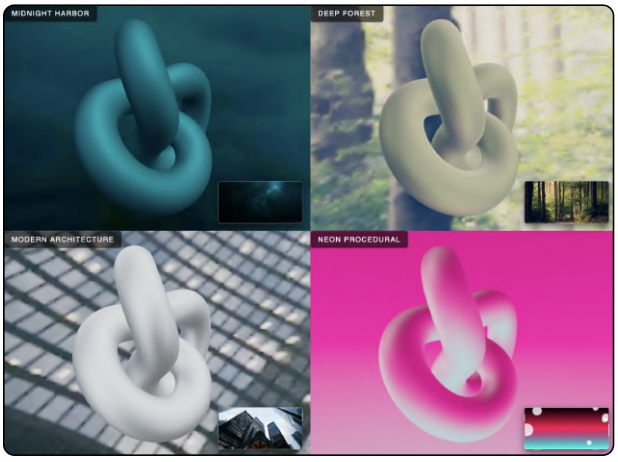

Figure 8: Environment light map visualization.

3.7 Scoring and Decision

To classify a video as real or AI-generated, the pipeline aggregates inconsistency signals from the multi-view geometry stage (COLMAP) and the inverse-rendering stage (3DGS and intrinsic relighting).

Geometric Consistency Score. Real videos of a mostly static world (after masking independently moving objects and overlays) admit a single rigid camera trajectory that explains the observed feature motion. In contrast, AI-generated videos often exhibit temporally inconsistent geometry (e.g., "breathing" structure or warping), leading to elevated bundle-adjustment residuals. We therefore define S_{geo} as the mean robust reprojection error over all valid feature-track observations in COLMAP bundle adjustment:

$$S_{\text{geo}} = \frac{1}{|\mathcal{O}|} \sum_{(i,j)\in\mathcal{O}} \rho\left(\|\mathbf{x}_{ij} - \pi(\mathbf{K}, \boldsymbol{\theta}, \mathbf{R}_i, \mathbf{t}_i, \mathbf{X}_j)\|_2\right)$$

where $\mathcal{O}$ is the set of all observed track measurements, $\mathbf{x}_{ij}$ is the pixel location of point j in frame i, $\pi(\cdot)$ is the projection function, and $\rho(\cdot)$ is the robust loss used in bundle adjustment. A higher S_{geo} indicates that a single rigid scene and camera path cannot explain the feature correspondences.

Photometric Fidelity Score. After fitting the 3DGS representation, the scene is rendered back to the original camera viewpoints to obtain $\hat{I}_k$ for each input frame I_k. We compute the mean PSNR:

$$S_{\text{photo}} = \frac{1}{N} \sum_{k=1}^{N} \text{PSNR}(I_k, \hat{I}_k).$$

Intuitively, real videos typically support a high-fidelity reconstruction (high PSNR), whereas synthetic videos with frame-to-frame texture flicker, inconsistent specularities, or non-rigid appearance changes tend to yield blurrier or artifact-heavy splatting reconstructions (lower PSNR), as the Gaussian primitives fail to converge to a stable explanation.

Illumination Stability Score. Using the intrinsic-relighting module, per-frame global lighting parameters L_k are estimated. For a static scene, global illumination in the world frame should be approximately constant over time (up to modest exposure changes). Temporal instability is measured via the variance of the estimated lighting parameters:

$$S_{\text{light}} = \text{Var}(L_1, \ldots, L_N).$$

AI-generated sequences may implicitly "bake" shadows into textures or shift effective light sources across frames; a physically grounded intrinsic solver can only compensate by oscillating the estimated lighting, producing a higher S_{light}.

The three scores are combined using either a simple logical gate (for interpretability) or a lightweight logistic regressor trained on held-out data:

$$P(\text{Real}) = \sigma(w_1 S_{\text{geo}} + w_2 S_{\text{photo}} + w_3 S_{\text{light}} + b),$$

where $\sigma(\cdot)$ is the logistic sigmoid. Videos with $P(\text{Real}) <$ threshold are flagged as AI-generated.

3.8 Complexity and Runtime

The detector combines several computationally intensive stages. Let N be the number of processed frames in the shot (after sampling), and let S denote the number of detected shots in the original clip.

(1) Segmentation and gating. Shot segmentation is linear in video duration. The VLM gate is evaluated only on a sparse set of keyframes (e.g., $K \in \{3, 5\}$ per shot), making the total number of VLM evaluations $\mathcal{O}(SK)$.

(2) Masking (SAM 3). Processing operates at $\mathcal{O}(N)$ with respect to frame count (one model pass per frame), with a constant factor that depends on the chosen image resolution and prompt complexity.

(3) Geometric reconstruction (COLMAP). COLMAP's incremental structure-from-motion has a worst-case feature-matching complexity of $\mathcal{O}(N^2)$ image pairs under exhaustive matching.

(4) Inverse rendering (RTR-GS). Both 3D Gaussian Splatting optimization and intrinsic relighting are iterative. For T optimization iterations, the cost is approximately $\mathcal{O}(N\,T)$ forward/backward rasterization passes, with a larger constant when optimizing additional factors (e.g., normals, albedo, and lighting). Testing found the best trade-off between stable output quality and runtime at approximately $T \approx 7000$ iterations.

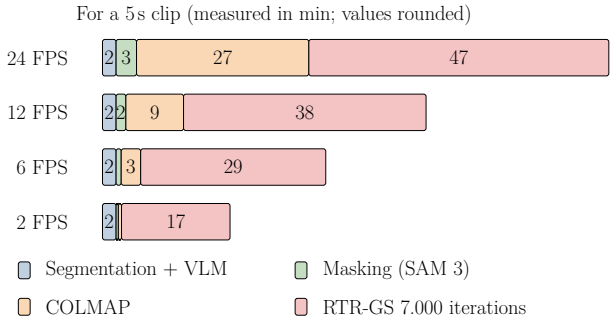

Figure 9: Pipeline example runtime for a 5 s clip at different sampling rates, evaluated on a Kaggle P100 GPU. Using a high-end consumer GPU (e.g., RTX 4090–5090), one could expect an overall speed increase of $\times 25$. Using a data-center GPU (e.g., H200), one could expect an overall speed increase of $\times 80$.

4 Datasets

Unbiasedness. To avoid overfitting to artifacts from any single dataset or generator, the datasets below are used to analyze and stress-test the method across diverse content and generators, but not to determine the final metric (e.g., to tune a decision threshold).

Composition. To ensure broad coverage ("all kinds of videos" & "all kinds of generators"), a composite dataset is built by sampling from:

- **GenVidBench** [19]: a broad multi-generator set (including real videos) used as the backbone.
- **ShareVeo3** [20]: a state-of-the-art generator dataset (Veo 3) from Google DeepMind.
- **Sora2** [21]: another modern generator dataset (Sora 2) from OpenAI.
- **AutoShot** [22]: real, short-form, edited content with social-media style post-processing.

- **MovieNet trailers (ECCV 2020)** [23]: real, professional trailers with cinematic lighting and motion.
- **Duke AdViews (real adverts)** [24]: real advertisements with varied compression.

Distribution. Appropriate samples from each dataset are selected to obtain an adapted distribution over (i) real vs. generated, (ii) generator family, and (iii) content type (trailers, adverts, short-form, etc.), avoiding domination by any single source.

5 Results

This section reports a quantitative evaluation designed to test the core hypothesis: real videos should be more consistent and physically grounded under multi-view reconstruction and inverse rendering than AI-generated videos.

This work reports results for the established geometric and photometric consistency scores, S_{geo} and S_{photo} (see Section 3.1 and related work). Since these metrics are well studied, they are not analyzed in depth here; instead, emphasis is placed on the contribution of the proposed pipeline, which broadens the range of footage for which these scores can be computed. In particular, by relaxing the requirement for perfectly static viewpoints and by leveraging the dynamic content present in most online videos, the pipeline reduces view-related limitations that previously prevented evaluation on many real-world clips.

In addition, this work introduces S_{light}, a novel metric intended to capture lighting- and inverse-rendering consistency.

5.1 Evaluation

Because the end-to-end pipeline has a long runtime, an initial evaluation was conducted on a small subset of the datasets presented in Section 4. Specifically, $N_{\text{real}} = \mathbf{10}$ real videos and $N_{\text{ai}} = \mathbf{10}$ AI-generated videos were processed. This sample size is not yet sufficient for strong statistical claims; however, it is adequate to observe a preliminary trend that aligns with the hypothesis. These results are therefore treated as an early indicator, and additional runs are ongoing to increase coverage, with the goal of reporting a more comprehensive analysis (including summary statistics and visualizations) for the in-person competition.

5.2 Preliminary results for S_{light}

Table 1 lists the per-video S_{light} scores for the current subset. In this convention, *lower* S_{light} indicates better consistency (i.e., fewer lighting-related violations under reconstruction and inverse rendering), so real videos are expected to achieve lower values than AI-generated ones.

Table 1: Preliminary S_{light} results on a small subset of real and AI-generated videos.

Video	**Real**	**AI-generated**
1	0.38	0.51
2	0.41	0.69
3	0.36	0.77
4	0.44	0.73
5	0.39	0.66
6	0.43	0.55
7	0.35	0.72
8	0.57	0.59
9	0.40	0.68
10	0.37	0.74
Mean (↓)	0.410	0.664
Std. dev.	0.065	0.089

Overall, the current results suggest that S_{light} separates real from AI-generated footage on this subset, supporting the hypothesis. Qualitatively, the separation appears stronger for older AI footage (e.g., earlier-generation models such as Pika), whereas the gap is reduced for more recent generators. A clearer picture is expected once N_{real} and N_{ai} are increased and the results are sorted by generator and content type.

6 Future Plans

More metrics. Right now, only three scores are used, but nothing limits us to these. To enhance AI detection, I have some ideas for additional metrics that could be built on the current pipeline without many changes.

1. Light drift: compute an environment map for the first and last frames, verify that the lighting remains approximately constant throughout the video, and then check whether the two environment maps diverge. If they do, it may indicate AI generation. This reflects a trend in some image models: they tend to become more yellow over time, which might also be observable in video models.

2. Wobbling shadows: detect temporal inconsistencies in shadows, although I am unsure how to implement this robustly.

3. Render the 3DGS at the estimated camera poses and compute the average divergence.

Free-to-use website. After some searching, I found that the availability of publicly usable tools is limited. Existing websites are either paid services or outdated detectors. Although many open-source detection methods exist, they are often not accessible to the general public. Making these tools easy to use and widely available increasingly feels like a basic need. More metrics can help, provided their outputs are interpreted correctly. As such, I plan to create a website that serves as a hub for multiple methods, using ad revenue and/or free compute to run some metrics on servers or, if they are lightweight, directly in the browser. Finally, I would also implement guidance for interpreting the results, as non-technical users may struggle if presented with a panoply of scores.

Toward a generator benchmark. Beyond detection, I want to turn DeLight-ReLight into a benchmark for *generators* as well. The plan is to define a fixed set of prompts, camera motions, and evaluation protocols, then run the same 3DGS reconstruction and lighting-consistency metrics on videos produced by different models. This would enable direct, apples-to-apples comparisons of generators in terms of physical plausibility (e.g., stable illumination, consistent geometry, and temporally coherent shadows), and provide a standardized way to track progress over time.

7 Limitations

Highly dynamic footage. A practical limitation of this pipeline is that reliable COLMAP reconstruction and 3DGS require sufficiently static content across frames. As explained previously, moving objects are therefore ignored. Nevertheless, if a large fraction of a frame is masked, the remaining unmasked pixels may be insufficient for stable estimation, which can affect the metrics. To mitigate this, a frame filter is applied that keeps only frames with at least a minimum threshold of unmasked pixels. As a consequence, highly dynamic footage (e.g., battle scenes) may yield too few usable frames, and the program may terminate early.

Static/near-static camera. SfM relies on parallax induced by camera motion. If the camera is static (or nearly static, e.g., tripod footage with only minor jitter), there may be insufficient baseline to reliably estimate camera poses and triangulate 3D points, causing COLMAP to fail or produce an unstable reconstruction. In such cases, downstream metrics that depend on SfM/3DGS are unavailable or unreliable, and the pipeline may terminate early. The VLM checks for this, so a flag is raised.

Scope. This work is not intended to distinguish AI-generated content from human-made CGI/animation. Instead, it focuses on whether an image/video is photorealistic and therefore plausibly misleading. Stylized content (e.g., Pixar-like animation or clearly non-realistic CGI) is typically recognized by viewers as non-real and interpreted more critically, so we prioritize realistic-looking visuals that may be misinterpreted as depicting reality.

8 Conclusion

The DeLight-ReLight benchmark addresses the erosion of digital trust driven by the rise of generative-AI video.

By repurposing 3D Gaussian Splatting as a forensic instrument rather than a creative one, this project establishes a deterministic framework for verifying reality. The DeLight-ReLight pipeline treats each video as a physical hypothesis. If a sequence of frames cannot be decomposed into a stable 3D volume with a temporally coherent illumination field, then the hypothesis of "reality" is falsified. These results suggest that, while modern generators can emulate the effects of light, they fail to simulate the mechanisms of light transport. The introduction of the Illumination Stability Score (S_{light}) provides a metric that is indifferent to AI "style" or "fidelity" and concerned only with immutable physical laws.

Preliminary results demonstrate that even state-of-the-art generators (e.g., Veo 3, Sora 2) struggle to maintain sub-pixel geometric stability and temporally coherent illumination. Unlike classifier-based detectors, which are destined to engage in a perpetual "cat-and-mouse" game with evolving generators, this physics-based approach generalizes across model architectures. Additionally, thanks to this robust pipeline, which can handle dynamic objects, older metrics become usable across a wider range of footage types. As long as generative models prioritize perceptual fidelity over physical simulation, the question "Does this video obey the laws of physics?" will remain a reliable discriminator of truth and help combat fake news, scams, and deepfakes.

AI Disclaimer

LLMs were only used to proofread for grammar/clarity, and to propose synonyms/alternate phrasing. At no point was new information or a full sentence added by an LLM.

Image Sources

All figures in this paper are created by the author unless otherwise stated.

References

[1] Joel Frank, Franziska Herbert, Jonas Ricker, Lea Schönherr, Thorsten Eisenhofer, Asja Fischer, Markus Dürmuth, and Thorsten Holz. A representative study on human detection of artificially generated media across countries, 2023.

[2] Liam Curtis. Ai slop report: The global rise of low-quality ai videos. Kapwing Blog, November 2025.

[3] Kate Dougherty and Brittany Tibaldi. Nearly 90% of advertisers will use gen ai to build video ads, according to iab's 2025 video ad spend & strategy full report. IAB News, July 2025.

[4] Ning Yu, Larry Davis, and Mario Fritz. Rooting deepfake attribution in training data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[5] Antonia Creswell and Anil Anthony Bharath. Inverting the generator of a generative adversarial network, 2018.

[6] A. Heidari et al. Deepfake detection using deep learning methods: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 2024.

[7] N. Tasnim et al. Ai-generated image detection: An empirical study and future research directions, 2025.

[8] Yongjian Hu, Huimin She, Beibei Liu, Xiangquan Chen, and Guangyao Liu. Deepfake video detection using 3dmm facial reconstruction information. *Geomatics and Information Science of Wuhan University*, 49(2):190–196, 2024.

[9] Brandon Castellano and PySceneDetect Contributors. Pyscenedetect: Video scene detection and split tool. GitHub repository, 2025.

[10] Johannes L. Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *European Conference on Computer Vision (ECCV)*, 2016.

[11] Johannes L. Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[12] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering, 2023.

[13] Boxun Hu, Mingze Xia, Ding Zhao, and Guanlin Wu. Mona: Moving object detection from videos shot by dynamic camera, 2025.

[14] Shuai Bai et al. Qwen3-vl technical report, 2025.

[15] Meta Superintelligence Labs. Sam 3: Segment anything. GitHub repository, 2025.

[16] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision (IJCV)*, 60(2):91–110, 2004.

[17] Martin A. Fischler and Robert C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[18] Yongyang Zhou, Fang-Lue Zhang, Zichen Wang, and Lei Zhang. Rtr-gs: 3d gaussian splatting for inverse rendering with radiance transfer and reflection, 2024.

[19] Zhenliang Ni, Qiangyu Yan, Mouxiao Huang, Tianning Yuan, Yehui Tang, Hailin Hu, Xinghao Chen, and Yunhe Wang. Genvidbench: A challenging benchmark for detecting ai-generated video, 2025.

[20] Wenhao Wang et al. Shareveo3 (veo 3 gallery) dataset. Hugging Face Datasets.

[21] YF789. sora2 (dataset). Hugging Face Datasets.

[22] Wentao Zhu, Yufang Huang, Xiufeng Xie, Wenxian Liu, Jincan Deng, Debing Zhang, Zhangyang Wang, and Ji Liu. Autoshot: A short video dataset and state-of-the-art shot boundary detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2023.

[23] Qingqiu Huang, Yu Xiong, Anyi Rao, Jiaze Wang, and Dahua Lin. Movienet: A holistic dataset for movie understanding. In *European Conference on Computer Vision (ECCV)*, 2020.

[24] John W. Hartman Center for Sales, Advertising & Marketing History and Duke University Libraries. Adviews digital collection. Duke University Libraries Digital Collections, 2009.