CoCoVideo: The High-Quality Commercial-Model-Based Contrastive Benchmark for AI-Generated Video Detection

Supplementary Material

A. Dataset Construction Details

As mentioned in Section 3.2, the overall framework of the dataset construction pipeline is shown in Figure 2, and we elaborate on each of the four key stages in detail below.

Stage I: Data acquisition and selection strategy. All real videos in our dataset are sourced from OpenVid-1M. While deepfake research typically focuses on talking-head and human-interaction content, we deliberately expand the coverage to include food preparation, plant growth, cultural architecture, and natural landscapes to enhance generalizability across diverse AIGC scenarios. Each selected video retains its textual description, which is directly reused as the generation prompt, and its first-frame image. To ensure reliable and temporally aligned references, we only include videos with at least 5s of continuous footage within a single shot. Category composition statistics are reported in Table 6. Note that four models (Kling, Pika, Vidu, Vivago) were added during the second dataset expansion, resulting in slight differences in their category distributions compared to other models.

Stage II: Two-round quality filtering. To ensure alignment and quality between generated outputs and real references, we apply two rounds of rigorous filtering. The first round operates at the video level, removing clips that appear unrealistic or lack temporal coherence, including animation-like videos, pseudo-motion over static images, abrupt shot transitions, severe flicker, geometric distortions, and videos shorter than 5s. The second round evaluates both the first-frame image and textual description. For images, we discard samples with black first frames, heavy blur, or missing main subjects. For text, we remove garbled or non-English descriptions to ensure consistent parsing, and filter out prompts exceeding platform length limits. After these two rounds, all samples meet the first-frame and text requirements, ensuring semantic and visual alignment for subsequent generation.

Stage III: Paired generation using commercial models. The filtered data are organized into batches of 1,000 videos according to specified category ratios. For each real video, its first-frame image and textual description are used as input to generate the corresponding fake video via commercial video generation models. Two generation modes are adopted: platform-based (through official web consoles) and API-based (via official interfaces). When these two modes use different model versions (e.g., Jimeng via platform and Seedance via API), we treat them as distinct generation methods. The generated video duration is fixed to 5 seconds for all models. Resolution is matched to the original video when possible; otherwise, a higher resolution is used to maintain visual quality and comparability. Across 13 commercial models, this stage produces a strictly matched fake counterpart for each real sample, yielding real–fake pairs with strong semantic and visual alignment.

Stage IV: Post-processing normalization. We apply post-processing to all generated pairs and record detailed metadata for each video. Videos with black borders or resolution mismatches are cropped and normalized. However, when models automatically extend content near borders (e.g., Veo3 automatically expands a 720×720 source video to 1080×720), we retain these extended regions rather than cropping them back to preserve each model's native generation characteristics. For each video pair, we provide detailed metadata including video name, caption (content description), camera motion, frame count, FPS, and duration. The final dataset comprises 13,000 real–fake pairs (26,000 videos in total), forming CoCoVideo for subsequent training and validation.

B. Dataset Visual Examples

Figure 5 presents visual examples from our dataset. We select real–fake pairs across different categories for each generation model. For videos with 1:1 aspect ratio, we display five uniformly sampled frames at 1-second intervals; for videos with 3:2 aspect ratio, we display three frames at 1s, 3s, and 5s. The examples illustrate that generated videos are visually similar to their real counterparts and exhibit consistent temporal dynamics.

Figure 6 shows an example of the multimodal metadata provided for each video pair. This example illustrates the annotations including video name, caption, camera motion, frame count, FPS, and duration, demonstrating the richness of metadata in CoCoVideo.

C. More Experimental Setup Details

C.1. Parameters Details

We train the model on a single NVIDIA A6000 GPU for 30 epochs, taking approximately 10 hours. The input video resolution is set to $W = H = 224$ pixels with temporal dimension $T = 16$ frames, and a batch size of 8 video pairs is used. During training, we apply data augmentation operations including: random horizontal flipping with probability 0.5 and color jittering (brightness=0.1, contrast=0.1, saturation=0.1, hue=0.05). We use the AdamW optimizer

Table 6. Distribution of video category proportions (%).

Model	Talking Head	Person Interaction	Cultural Architecture	Natural Landscape	Food Preparation	Plant Growth	Others
Jimeng	56.3	5.4	3.9	21.7	1.8	6.8	4.1
Hailuo	55.4	4.7	5.3	26.6	1.6	3.1	3.3
Luma	57.9	5.9	4.8	22.6	1.0	4.7	3.1
Pixverse	56.3	4.4	4.5	25.6	1.2	4.0	4.0
Runway	80.5	1.7	1.2	11.8	0.8	1.7	2.3
Seedance	56.3	5.7	6.2	24.0	1.1	3.1	3.6
Veo	51.2	4.0	5.9	28.4	1.1	5.7	3.7
Kling	28.0	57.9	1.2	3.5	1.6	4.0	3.8
Pika	30.2	49.7	0.6	3.6	2.4	2.6	10.9
Vidu	31.6	55.6	1.1	2.2	2.3	2.8	4.4
Vivago	33.3	48.5	1.4	2.1	1.3	2.5	10.9
Midjourney	54.7	4.5	3.8	23.7	2.3	6.8	4.2
Sora	55.0	4.8	6.0	24.7	2.1	4.1	3.3

with an initial learning rate of 1×10^{-4} and weight decay of 1×10^{-4}, with a cosine annealing learning rate schedule over 30 epochs.

For the model-specific parameters, the classification loss weight is set to $\alpha = 0.65$, the paired contrastive loss uses a margin parameter of $m = 1.0$, and the confidence-gated inference mechanism employs a threshold of $\tau = 0.9$.

C.2. MLLM Configuration

We employ LLaVA-NeXT-Video-7B as the semantic reasoning model for analyzing uncertain samples routed by the confidence-gated mechanism. The model is configured with maximum output tokens of 512 and temperature of 0.5, using the default frame sampling rate. As shown in Figure 7, a task-specific prompt is used to guide the MLLM's reasoning process, which is jointly processed with the sampled video frames to generate the final authenticity prediction.

D. Additional Experiments

D.1. Hyperparameter Sensitivity Analysis

Loss Function Weights. We analyze the sensitivity of CoCoDetect to the loss function hyperparameters α and margin m by fixing one parameter and varying the other. To isolate the impact of the contrastive learning component, we conduct these experiments without MLLM integration. Table 7 compares the performance across different configurations in terms of Acc, F1, Recall, and AUC. The results show that $\alpha = 0.65$ and $m = 1.0$ achieve the best Acc, F1, and AUC scores. While some configurations yield higher recall, they sacrifice precision, leading to lower F1-Score and accuracy, confirming that our hyperparameter choice provides the optimal balance.

Confidence Threshold Selection. The confidence threshold τ controls the trade-off between detection accuracy and sample coverage in the contrastive learning module. Table 8 compares the performance under different threshold values ranging from 0.6 to 0.9. As τ increases, the accuracy improves but the sample coverage decreases. To balance these two metrics, we compute an F_β-like score:

$$F_\beta = \frac{(1+\beta^2) \cdot \text{Cov} \cdot \text{Acc}}{\beta^2 \cdot \text{Cov} + \text{Acc}} \tag{9}$$

where *Acc* denotes accuracy and *Cov* denotes coverage rate. We set $\beta = 2$ to emphasize accuracy, which is prioritized in this detection task. The results show that $\tau = 0.9$ achieves the highest F_2 score, providing the optimal balance.

Table 7. Performance under different loss weight configurations (%). **Bold** indicates the best performance.

α	m	Acc	F1	Recall	AUC
Varying α (fixed $m = 1.0$)					
0.5	1.0	87.79	88.05	89.85	94.41
0.8	1.0	88.69	88.70	88.72	95.01
0.9	1.0	87.08	87.00	86.46	93.62
Varying m (fixed $\alpha = 0.65$)					
0.65	0.5	86.08	85.52	82.26	93.38
0.65	1.5	83.38	84.97	**93.90**	93.18
Final configuration					
0.65	**1.0**	**88.92**	**88.79**	87.74	**95.46**

D.2. Backbone Comparison

To validate the effectiveness of our backbone choice, we conduct comparative experiments with different architectures. Table 9 compares the performance of ResNet-18

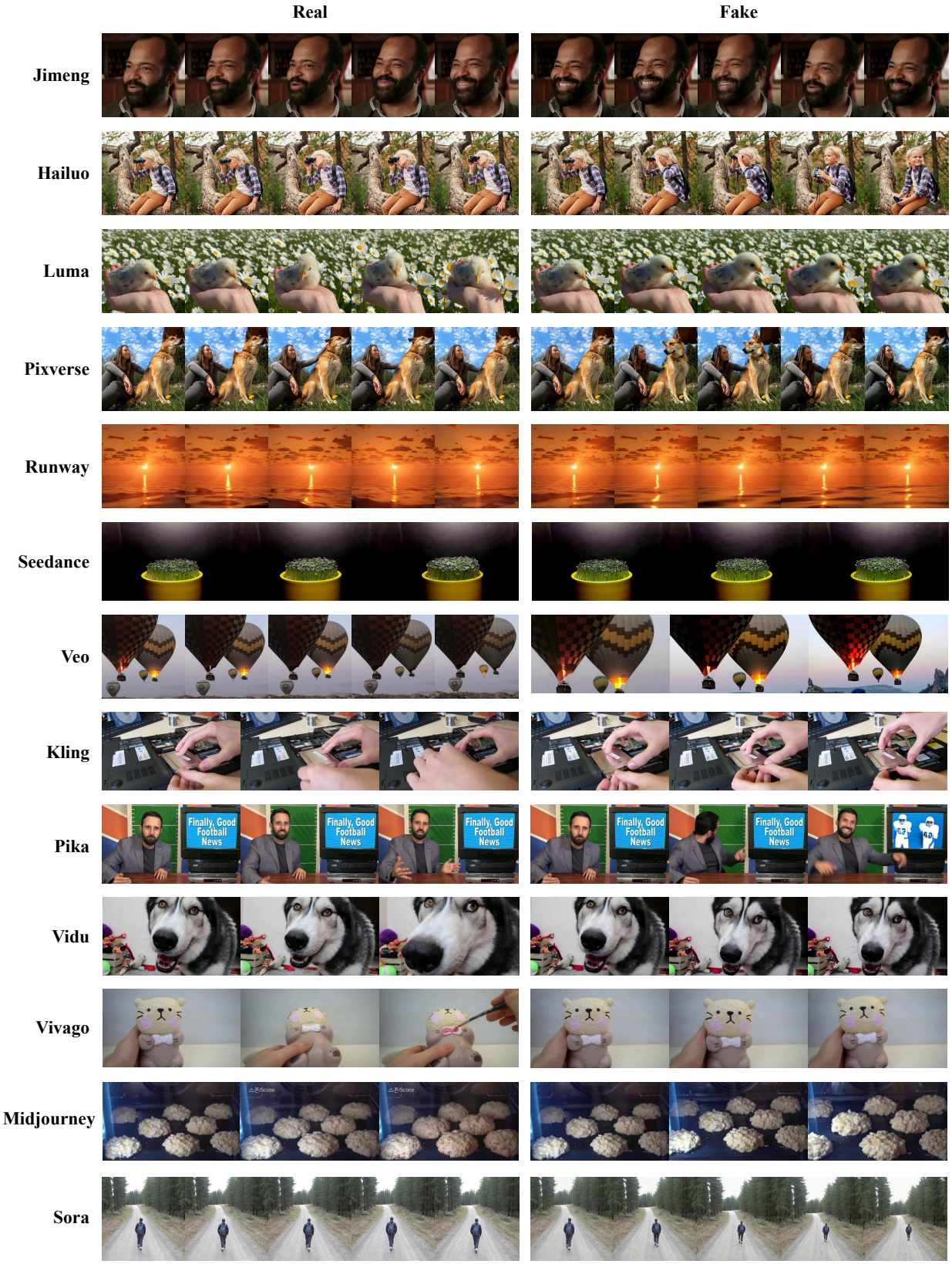

Figure 5. Visual examples of real–fake video pairs across different generation models.

Metadata

video: pexels_landscape_landscape_4797177_merged.mp4
caption: a serene scene of a beach at dusk. The sky, painted in a soft shade of pink, meets the darker hues of the ocean at the horizon. The ocean itself is a light blue, with white waves rhythmically crashing onto the shore. The sandy beach, a light beige in color, stretches out towards the right side of the frame. In the distance, a line of trees can be seen, their silhouettes adding depth to the landscape. The photo is taken from a low angle, making the viewer feel as if they are standing right there on the beach, looking out at the vast expanse of water and sky. The overall composition of the scene suggests a sense of tranquility and peace.
camera motion: zoom_in+tilt_up
frame: 122
fps: 24
seconds: 5.083333333333333

Figure 6. Example of the multimodal annotations provided for each video pair in our dataset.

Please analyze whether this video is a real video shot by a camera or generated by AIGC(fake), considering the following aspects: frame continuity, image realism, stylistic realism, and whether it conforms to the laws of the physical world.

*We have obtained a confidence score **p={confidence_score}** for this video through a contrastive learning network at the texture level.*
A score closer to 0 indicates the model is more certain it is a fake video, while a score closer to 1 indicates the model is more certain it is a real video.
This score can serve as a reference for your judgment, but you should primarily focus on providing your analysis from the video semantic perspective.

You should make a judgment based on the following aspects:
1) Frame discontinuity, obvious artifacts or other traces of fake generation.
2) If the video is clearly not actually filmed, or if its style is not realistic but rather resembles a cartoon or watercolor painting, then it is judged to be AI-generated.
3) If the content between frames does not conform to the physical logic of the real world, or if certain elements appear and disappear out of thin air, then it is determined to be AI-generated.

Please provide your analysis in the following JSON format:

```
{
   "prediction": "fake" or "real",
   "confidence": 0.0 to 1.0,
   "reasoning": "brief explanation"
}
```

Where:
- "prediction": "fake" = fake/AI-generated, "real" = real/authentic
- "confidence": a decimal between 0.0 and 1.0 indicating how confident you are in your prediction (0.0 = no confidence, 1.0 = absolute confidence)
- "reasoning": a brief explanation of your judgment

Please respond ONLY with the JSON format, no additional text.

Figure 7. The text prompt provided to the MLLM for video authenticity reasoning.

Table 8. Performance comparison under different confidence thresholds (%). **Bold** indicates our choice with the best F_2-Score.

τ	Accuracy	Coverage	F_2-Score
0.6	89.03	99.62	90.97
0.7	89.26	99.08	91.06
0.8	89.49	98.59	91.18
0.9	**89.77**	**97.51**	**91.22**

with other commonly used backbones including ResNeXt, VideoMAE, and MViT-v2. The results show that ResNet-18 achieves the best F1, Acc, and AUC scores. Although MViT-v2 achieves higher recall, it results in lower F1, Acc, and AUC due to reduced precision. We attribute this to the unique characteristics of our dataset, where real and fake videos share strong semantic correlations due to the paired generation process. More complex architectures with higher capacity tend to overfit to subtle training-specific patterns, leading to reduced generalization compared to the simpler ResNet-18 structure. This finding suggests that lightweight models are more suitable for detecting AI-

generated videos with high semantic similarity to their real counterparts.

Table 9. Performance comparison of different backbone architectures (%). **Bold** indicates the best performance.

Backbone	Acc	F1	Recall	AUC
ResNeXt	80.49	80.65	81.33	88.16
VideoMAE	79.64	77.98	72.10	89.26
MViT-v2	86.21	86.95	**91.90**	94.28
ResNet-18 (Ours)	**88.92**	**88.79**	87.74	**95.46**

D.3. Robustness Analysis

To evaluate the robustness of CoCoDetect under realistic perturbations, we apply various distortions to the test videos and measure the performance degradation. We consider six common perturbation types with the following configurations: (1) CRF compression with quality factor CRF=28 to simulate video transmission compression; (2) grayscale conversion to remove all color information; (3) Gaussian noise with standard deviation $\sigma = 0.1$ added to normalized pixel values; (4) Gaussian blur with kernel size 5×5 and $\sigma = 2.0$ to simulate defocus; (5) geometric transformation including random rotation within ± 15 and scaling by factor $0.9 \sim 1.1$; and (6) local occlusion with 20% of the frame area randomly masked. Table 10 presents the performance of CoCoDetect under these perturbations in terms of Acc, F1-Score, and AUC. The results show that CoCoDetect maintains relatively stable performance under geometric transformation and local occlusion. MLLM helps correct some misclassified samples through semantic reasoning. However, Gaussian noise and blur cause more significant performance drops, as these perturbations lead to high-confidence incorrect predictions by the contrastive learning module, preventing samples from being routed to MLLM for correction.

D.4. MLLM Selection Study

To select the most suitable MLLM for semantic reasoning, we conduct a comprehensive comparison across three key dimensions: response speed, reasoning accuracy, and output format compliance (the ability to consistently generate structured outputs adhering to the required JSON format). We use 500 videos from the GVD dataset as our evaluation benchmark for two reasons: (1) their short duration requires fewer frames to be transmitted to the MLLM, enabling faster response times; (2) all videos in GVD are AI-generated with obvious semantic inconsistencies that violate real-world physics, making it effective for validating whether MLLMs can identify physical implausibilities.

Table 10. Robustness evaluation of CoCoDetect under different perturbations. All metrics are in percentage (%). **Bold** indicates the best performance.

Perturbation	Acc	F1	AUC
CRF Compression (28)	74.82	77.61	85.43
Grayscale	76.13	73.51	84.19
Gaussian Noise ($\sigma = 0.1$)	59.49	62.16	63.01
Gaussian Blur ($\sigma = 2.0$)	66.87	68.16	69.01
Geometric Transform (± 15)	84.92	85.60	92.85
Local Occlusion (20%)	83.41	84.69	92.82
None (Original)	**90.69**	**90.62**	**95.93**

We compare four representative 7B-parameter models: LLaVA-NeXT-Video-7B, Qwen2.5-VL-7B-Instruct, LLaVA-1.5-7B-hf, and DeepSeek-VL2. Table 11 summarizes the results. LLaVA-NeXT-Video-7B achieves the highest reasoning accuracy with near-perfect format compliance, demonstrating the best balance for our task despite slightly longer response time than LLaVA-1.5-7B-hf. While LLaVA-1.5-7B-hf has the fastest response time, its significantly lower accuracy makes it unsuitable for this task. Therefore, we adopt LLaVA-NeXT-Video-7B in our framework. Note that due to computational constraints, we restrict our evaluation to 7B models; larger models may yield further improvements in reasoning accuracy, though at the cost of increased inference time and resource requirements.

Table 11. Comparison of different MLLMs on the GVD dataset. *T* denotes Response Time (seconds), *Acc* denotes Accuracy (%), and *F.C.* denotes Format Compliance (%). **Bold** indicates the best performance.

Model	T	Acc	F.C.
LLaVA-1.5-7B-hf	**2.19**	44.00	**100.00**
DeepSeek-VL2	3.28	79.60	95.40
Qwen2.5-VL-7B-Instruct	3.91	92.40	97.60
LLaVA-NeXT-Video-7B	3.72	**95.38**	99.60

Figure 8 illustrates a correctly formatted MLLM response from LLaVA-NeXT-Video-7B. Common format compliance failures include: (1) adding extraneous conversational text (e.g., *"Sure, here is my JSON output..."*); (2) omitting curly braces around JSON content; and (3) missing quotation marks around field names or values. These issues highlight the importance of format compliance in our evaluation.

*{\n \"**prediction**\": \"fake\",\n \"**confidence**\": 0.9,\n \"**reasoning**\": \"The image features a high level of frame discontinuity, obvious artifacts, and does not conform to the laws of the physical world. It resembles a cartoon or watercolor painting, which are indicative of AI-generated content. The dog's pose and the guitar are also unrealistic, suggesting a faked video. The confidence is high as it aligns well with AI-generated content.\"\n}*

Figure 8. Example of correctly formatted MLLM output with proper JSON structure.

E. Copyright and Ethical Considerations

We prioritize copyright compliance throughout the construction of CoCoVideo-26K to ensure legal and transparent data sourcing. For the real-video component, all samples are sourced exclusively from the OpenVid-1M dataset, which is released under a CC-BY-4.0 license that allows redistribution and adaptation with proper attribution. We follow the attribution requirements specified by the license in all downstream usage. For the synthetic-video component, we employ only official commercial model platforms or their authorized APIs, which operate under standard usage policies that permit generated content to be used for academic research purposes. It aims to align the generation process with model providers' terms of service, thereby reducing the risk of unauthorized use or potential copyright infringement.

Beyond legal compliance, we carefully consider the ethical implications of constructing and releasing a deepfake detection benchmark. Our primary motivation is to advance defensive technologies against malicious AI-generated content, thereby contributing to a safer digital media ecosystem. We deliberately avoid including sensitive categories to minimize potential misuse while maintaining research value. The dataset is made available and released exclusively for academic research purposes. By transparently documenting our data sources, generation methods, and ethical safeguards, we aim to promote responsible research on AI-generated media that balances scientific advancement with societal responsibility.