SLVMEval: Synthetic Meta Evaluation Benchmark for Text-to-Long Video Generation

Supplementary Material

A. Human annotation and quality control

This section provides details of the crowdsourced annotation described in Section 4.4.

A.1. Annotation platform and crowd workers

Platform. We collected all human annotations on Yahoo! Crowdsourcing, a commercial crowdsourcing platform operated by Yahoo Japan Corporation.

Crowd workers. In total, 736 unique crowd workers completed 3,793 annotation tasks. We recruited only general users registered on the platform and did not require any specific expert background.

Compensation. For each task, defined as evaluating a single video pair, we set the reward based on the expected time needed to watch and compare the two videos. The unit price was set so that the effective hourly wage was higher than the legal average minimum wage in Japan. Across the entire annotation project, the total payment was 408,520 JPY (approximately 2,685 USD at the time of the experiments).

A.2. Collection of human accuracy scores

As described in Section 3.2, we use accuracy as the meta evaluation metric. To obtain the human performance, we asked crowd workers to solve the same pairwise comparison task as the automatic evaluation systems.

Given a prompt p and a video pair $\{v_p^+, v_p^-\}$, the interface showed the prompt and two video players side by side. Since the original prompts were written in English while our crowd workers were Japanese, we translated each prompt into Japanese using Qwen3-Next-80B-A3B-Instruct-FP8 [39, 47] and presented the both original and translated prompt p. We instructed workers to read the prompt p, watch both videos, and indicate which video they considered to be higher quality with respect to the target aspect. The video player allowed workers to move the playhead using a seek bar and to replay arbitrary segments of each video, so that they could inspect any part of the content if necessary.

We computed human accuracy in the same way as for the evaluation systems, following the definition in Section 3.2, that is, as the proportion of pairs in which the worker selected v_p^+.

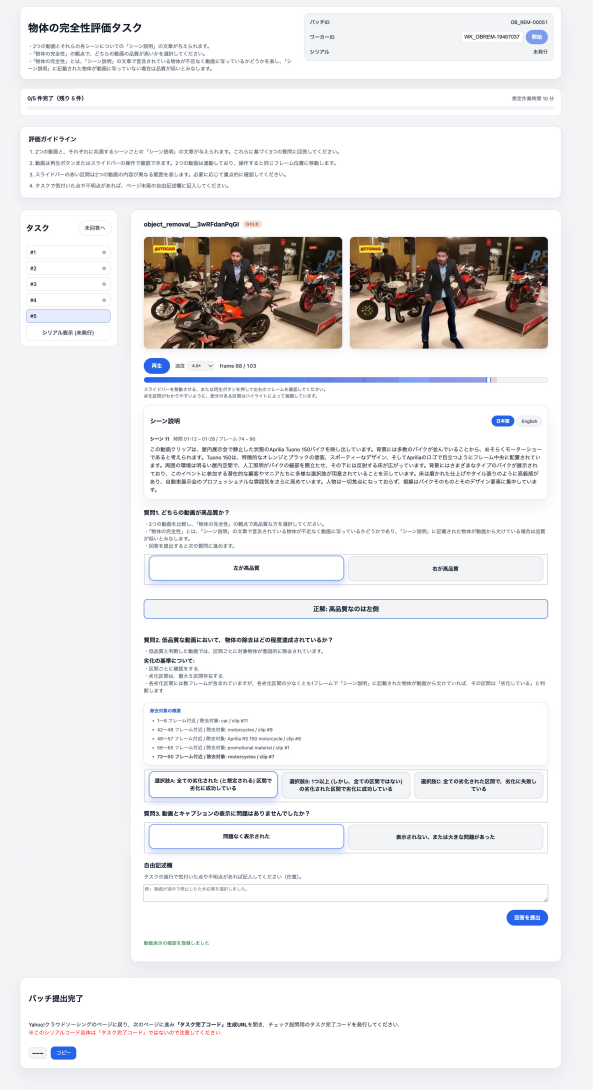

Figure 5. Example of the annotation interface used for the *Object Integrity* aspect. The interface presents the prompt and two videos, collects the worker's choice of the higher quality video, and then asks additional questions to verify whether the synthetic degradation for the target aspect has been applied correctly.

A.3. Annotation UI for degradation checking

Figure 5 shows the annotation interface used for degradation checking. In addition to the pairwise quality judgment, we collected A/B/C ratings that indicate whether the synthetic degradation succeeded for each aspect, as described

Table 3. Filtering statistics for the annotated degraded pairs. For each aspect, we report the number of candidate samples before filtering (Initial), the number of samples excluded because at least one worker selected option C (Excluded ($C \neq 0$)), the number of samples excluded because the number of votes for option A did not exceed that for option B (Excluded ($A \leq B$)), the number of samples retained after filtering (Final), and the resulting retention rate (Retention Rate). The filtering conditions follow the human annotation procedure defined in Section 4.4.

Aspect	**Initial Samples**	**Excluded ($C \neq 0$)**	**Excluded ($A \leq B$)**	**Final Samples**	**Retention Rate (%)**
Aesthetics	446	92	72	282	63.3
Technical Quality	446	243	72	131	29.4
Appearance Style	446	112	22	312	70.0
Background Consistency	446	75	17	354	79.4
Temporal Flow	444	115	44	285	64.2
Comprehensiveness	430	83	112	235	54.7
Object Integrity	375	96	179	100	26.7
Spatial Relationship	435	120	79	236	54.3
Dynamics Degree	444	89	22	333	75.0
Color	327	56	67	204	62.4
Total	4239	1081	686	2472	58.3

in Section 4.4. After each crowd worker finishes judging which of the two videos is better, we present our intended gold label. They then rate on a three-point A/B/C scale whether this gold label is appropriate. For each video pair and aspect, five crowd workers provided A/B/C judgments with the following options:

A. The degradation clearly succeeds in all selected clips.
B. The degradation succeeds in at least one but not all selected clips, or the effect is weak.
C. The degradation fails in all selected clips.

Table 4 lists the evaluation criteria shown to crowd workers for each aspect of the human evaluation.

Viewing enforcement. Long-video evaluation tasks often yield unreliable answers when workers skip large portions of the videos. To reduce this risk, we introduced a viewing enforcement mechanism in the degradation-checking UI. For each pair, we predefined the clip interval where the degradation was applied. In the interface, workers could freely move the seek bar, but the A/B/C response buttons for degradation success remained disabled until the playhead had been moved to the designated degraded interval and the worker clicked a confirmation button indicating that they had checked the degraded region. This design made skipping behavior difficult.

A.4. Crowd worker filtering

We additionally applied worker-level filtering to reduce the impact of unreliable crowd workers. We monitored response patterns across tasks and identified workers who repeatedly exhibited low-effort behavior. We used the following indicators.

1. **Abnormally short completion times.** We flagged workers whose task completion times were extremely short relative to the total duration of the video pair, for example those consistently in the fastest 10% for a given task configuration.
2. **Persistent disagreement on easy cases.** For samples where the agreement among other workers was very high (for example, 4 out of 5 workers chose A), we flagged workers who repeatedly gave clearly inconsistent answers (for example, answering C) on such samples.
3. **Very low task level accuracy.** For tasks where the majority of workers achieved high accuracy (around 80% or higher), we flagged workers whose accuracy remained very low (20% or lower) over many assignments.

Workers who exhibited problematic response patterns according to multiple indicators were manually reviewed. Workers judged to be unreliable were excluded from later task assignments. By the time we finished collecting annotations for the last aspect, this filtering process had excluded 227 workers in total from annotation tasks.

A.5. Filtering statistics

We applied the filtering criteria defined in Section 4.4 to the A/B/C degradation ratings to construct the final testbed. Concretely, we retained a sample only if it satisfied both

$$\text{count}(C) = 0 \quad \text{and} \quad \text{count}(A) > \text{count}(B),$$

where $\text{count}(A)$, $\text{count}(B)$, and $\text{count}(C)$ denote the numbers of workers who chose A, B, and C, respectively.

Table 3 summarizes the statistics of this filtering process for each aspect. For each aspect, the table reports the number of candidate samples before filtering (Initial), the number of samples excluded because at least one worker selected option C (Excluded ($C \neq 0$)), the number of samples

Table 4. Evaluation criteria and instructions provided to crowd workers for each aspect. The Japanese instructions were presented to the workers, and the English instructions are their translations.

Aspect	Japanese Instruction	English Instruction
Aesthetics	「美的品質」とは，各フレームが構図・調和・写真的品質を含む総合的な視覚的魅力を有しているかどうかを表し，「シーン説明」との色合いに矛盾があったり，動画の見た目の好ましさが劣っている場合は品質が低いとみなします.	"Aesthetics" indicates whether each frame possesses overall visual appeal, including composition, harmony, and photographic quality. Quality is considered low if there is a tonal contradiction with the "scene description" or if the visual appeal of the video is inferior.
Technical Quality	「技術的品質」とは，技術的な映像のミス（例：ぼけ・ブレ，ノイズ，低画質，輪郭歪み）がないかどうかを表し，技術的な映像ミスが存在する場合は品質が低いとみなします.	"Technical Quality" indicates whether the video is free from technical artifacts (e.g., blur/shake, noise, low resolution, contour distortion). Quality is considered low if such technical video errors exist.
Appearance Style	「スタイル」とは，「シーン説明」で指定された外観スタイル（例：油絵風，漫画風，水彩画風，スケッチ風など）が動画内で適切に再現されているかどうかを表し，指定されたスタイルと異なる表現にとなっている場合は品質が低いとみなします.	"Appearance Style" indicates whether the visual style specified in the "scene description" (e.g., oil painting, cartoon, watercolor, sketch) is appropriately reproduced in the video. Quality is considered low if the representation differs from the specified style.
Background Consistency	「背景の一貫性」とは，フレーム間で背景が視覚的に安定かつ一貫して保たれているかどうかを表し，連続フレームで背景が一貫していない動画は品質が低いとみなします.	"Background Consistency" indicates whether the background remains visually stable and consistent across frames. Quality is considered low if the background is inconsistent across consecutive frames.
Object Integrity	「物体の完全性」とは，「シーン説明」の文章で言及されている物体が不足なく動画に写っているかどうかを表し，「シーン説明」に記載された物体が動画に写っていない場合は品質が低いとみなします.	"Object Integrity" indicates whether the objects mentioned in the "scene description" are present in the video without omission. Quality is considered low if objects described in the "scene description" do not appear in the video.
Color	「色」とは，各フレーム内における色が「シーン説明」で指定された色（例：物体の色など）と一致しているかどうかを表し，指定された色と異なる場合は品質が低いとみなします.	"Color" indicates whether the colors within frames match those specified in the "scene description" (e.g., object colors). Quality is considered low if the colors differ from the specifications.
Dynamics Degree	「動きの度合い」とは，各フレームに含まれる動きの量が適切かどうかを表し，「シーン説明」の内容に比較して静的すぎたり，過度に不規則な場合は品質が低いとみなします.	"Dynamics Degree" indicates whether the amount of motion in each frame is appropriate. Quality is considered low if the video is too static or excessively erratic compared to the content of the "scene description."
Comprehensiveness	「網羅性」とは，「シーン説明」の内容が動画に反映されているかどうかを表し，「シーン説明」に記述されているシーンが動画内で適切に表示されない場合には品質が低いとみなします.	"Comprehensiveness" indicates whether the content of the "scene description" is reflected in the video. Quality is considered low if scenes described in the "scene description" are not appropriately displayed within the video.
Spatial Relationship	「空間的関係」とは，各フレームの物体が「シーン説明」で示された空間的関係（左右/上下/前後/近接）と一致しているかどうかを表し，いづれかの空間関係に矛盾が生じている場合は品質が低いとみなします.	"Spatial Relationship" indicates whether the objects in each frame match the spatial relationships (left/right, up/down, front/back, proximity) indicated in the "scene description." Quality is considered low if a contradiction arises in any spatial relationship.
Temporal Flow	「時系列の流れ」とは，「シーン説明」の文章のイベントの発生順番と動画内のシーンの順番が適切に対応しているかを表し，「シーン説明」の内容に矛盾した無関係なイベントが発生する動画は品質が低いとみなします.	"Temporal Flow" indicates whether the order of events in the "scene description" text corresponds appropriately to the order of scenes in the video. Quality is considered low if irrelevant events occur or if the order contradicts the "scene description."

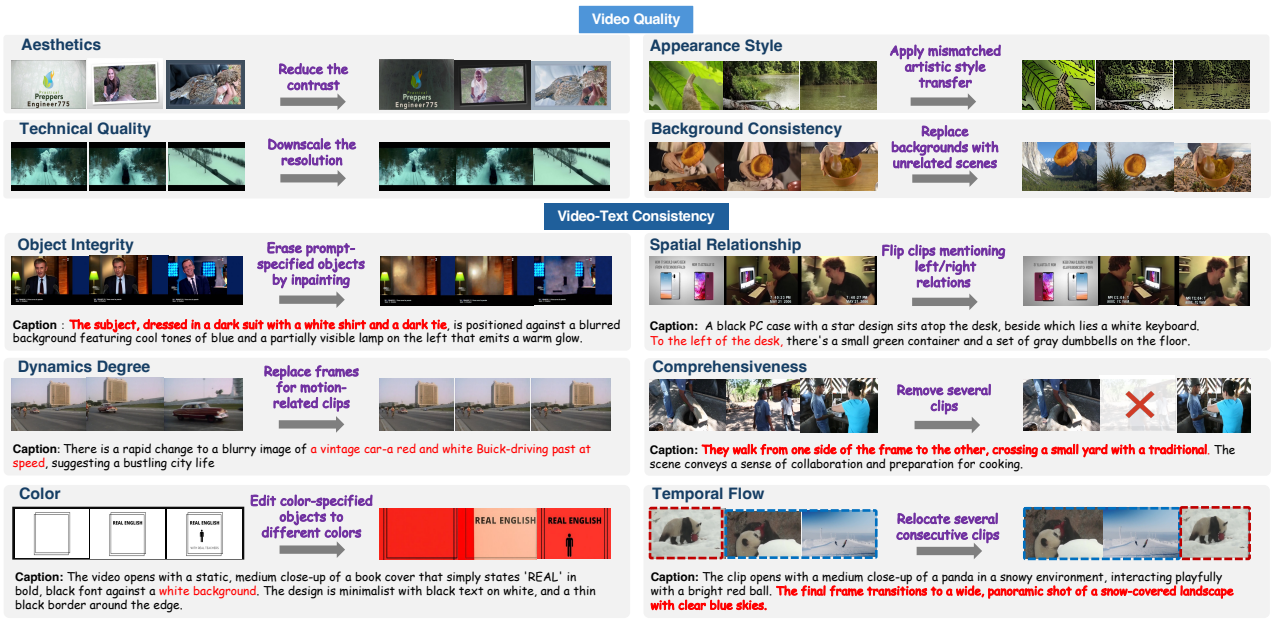

Figure 6. Representative examples of original and degraded video pairs in SLVMEval. For each of the 10 aspects, we show one original video (left) and its aspect-specifically degraded video (right). The upper rows correspond to the *Video Quality* aspects, and the lower rows correspond to the *Video-Text Consistency* aspects. Frames are sampled around the clips where the synthetic degradation is applied.

excluded because the number of votes for A was not greater than the number of votes for B (Excluded ($A \leq B$)), the number of samples retained after filtering (Final), and the resulting retention rate. Under these conditions, at least 100 long video pairs remain for every aspect.

B. Examples of degraded video pairs

Figure 6 shows example pairs of original and degraded videos, one for each of the ten aspects in SLVMEval. For each aspect, we show one source long video and its degraded counterpart, constructed using the aspect-specific operation described in Section 4.3. In each pair, the left column displays frames from the original video and the right column displays frames from the degraded video. We extract frames from around the clips to which the degradation is applied.

The aspects in the *Video Quality* category (Aesthetics, Technical Quality, Appearance Style, Background Consistency) appear in the top row, and those in the *Video-Text Consistency* category (Object Integrity, Color, Dynamics Degree, Comprehensiveness, Spatial Relationship, Temporal Flow) appear in the bottom row.

C. Proxy validation on generated videos

Directly validating SLVMEval on real long generated videos is currently difficult, as existing text-to-long video generation systems do not yet reliably produce videos at the target scale by SLVMEval with sufficiently stable quality. Therefore, we conduct a proxy experiment on short videos to examine whether the relative ranking of evaluation systems induced by our synthetic degradations is consistent with the ranking obtained in a generated-video comparison setting.

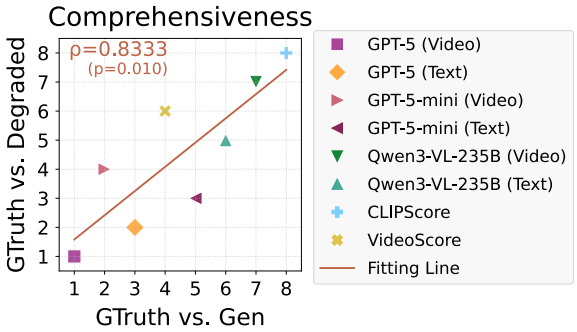

Figure 7. Rank consistency on FETV.

C.1. Dataset and settings

We use a subset of FETV [26], which contains short videos lasting a few seconds with human annotations for multiple aspects, including *alignment*. FETV provides pairs of human-created videos (GTRUTH) and generated videos (GEN), together with human quality signals. We also construct DEGRADED videos by applying our degradation op-

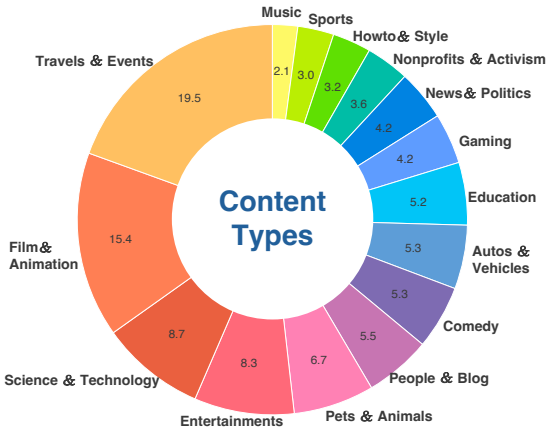

Figure 8. Distribution of video content categories in SLVMEval, showing a balanced representation across 15 categories.

erations to GTRUTH. Since our degradation operations assume a multi-clip structure (Algorithm 1), we concatenate two videos and treat each as a clip.

We consider two pairwise meta-evaluation settings: (i) GTRUTH vs. GEN; the video with a higher human score is regarded as higher quality, (ii) GTRUTH vs. DEGRADED; the ground truth is always regarded as higher quality. For each setting, we evaluate each system using pairwise accuracy (§ 3.2), rank systems by accuracy, and compute the Spearman rank correlation between the two system rankings.

C.2. Results and interpretation

We focus on the FETV aspect *alignment*, which most closely corresponds to our **Comprehensiveness** aspect. As shown in Figure 7, the system ranking derived from GTRUTH vs. DEGRADED is strongly correlated with the ranking derived from GTRUTH vs. GEN ($\rho = 0.8333$). Although limited to a single aspect and short videos, this suggests that the properties of the generated videos and our degraded videos are unlikely to be overly dissimilar.

D. Dataset statistics

Figure 8 summarizes the distribution of video content categories in SLVMEval. As described in Section 4.2, we construct the benchmark from a dense video captioning dataset that spans diverse real-world topics, and then apply aspect-specific degrading operations followed by human filtering. The resulting test set covers 15 distinct categories, including everyday activities, nature, sports, indoor scenes, and others. No single category dominates the dataset, and the categories are approximately balanced so that the benchmark evaluates robustness to long-video evaluation without bias toward a particular type of content.

Figure 9 shows the distribution of video durations. We

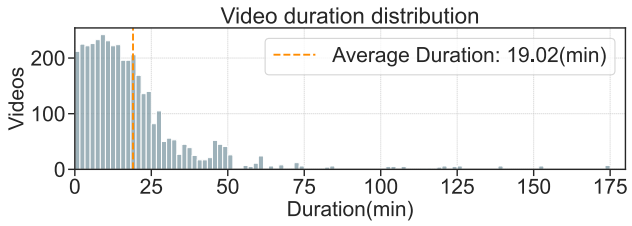

Figure 9. Distribution of video durations in SLVMEval. We divide the video duration into 100-second intervals and plot the number of videos included in each interval.

partition the videos into 100-second bins and plot the number of videos in each bin. Consistent with the description in the main paper, the dataset covers a wide range of durations, from several minutes to very long videos of up to 10,486 seconds (approximately 2 hours and 54 minutes), with an average duration of 1,141 seconds (about 19 minutes). It covers a wide range of video durations, from a few minutes to several hours, allowing us to evaluate the long-form videos targeted by SLVMEval.

E. Details of degrading operations

This section provides details on some of the aspects described in Section 4.3.

E.1. Information Extraction from Captions

As a preprocessing step, we extracted the information necessary for the degrading operations from the caption annotations of each video clip included in the source dataset (Vript [48]). Specifically, we used Qwen3-8B (thinking mode) [39] to extract content mentioning object colors, background elements, and the dynamics degree. We perform inference with vllm [21] and use the structured outputs mode to enforce JSON format for the output. The template for the prompt given to the model is as follows.

```
# Video Caption Analysis Prompt

## Task Description
Analyze the provided single scene video
caption and extract information related
to 7 evaluation criteria. Extract
relevant information for each criterion
and output in JSON format.

## Evaluation Criteria Definitions

- **subject**: Objects in the scene
(e.g., person, animal, car, etc.)
  - Distinguish from scenes which refer
  to places or environments
```

- **background**: References to background elements

- **dynamic_degree**: Object movements within the caption (e.g., person running, jumping, car driving, bird flying)
  - Exclude camera movements (zoom, pan, etc.)
  - Record only object movements

- **human_action**: Human actions (e.g., running, jumping, walking, sitting)
  - Record only human actions
  - Use terms like "person", "man", "woman" for agents

- **color**: Color names only (e.g., red, white, blue, green)
  - For "red car", record "red" and "car" separately
  - Map colors to their corresponding objects

- **spatial_relationship**: Clear spatial relationships between objects (left, right, up, down, front, back, etc.)
  - Extract only when clear positions are stated like "upper right", "lower left"
  - Map positions to objects at those locations

- **scene**: Scene types or environments (e.g., river, ocean, city, mountain, etc.)
  - Refers to places or environments, distinct from individual objects

Analysis Instructions

Analyze the caption in detail and execute the following:

1. **Information Extraction**: Extract specific information related to each evaluation criterion
2. **Empty Cases**: If no relevant information exists, output as empty list []

Output Format

Output in the following JSON format:

```json
{
  "subject": ["object1", "object2"],
  "background": ["background_element1",
  "background_element2"],
  "dynamic_degree": [
    {
      "agent": "actor",
      "action": "action_description"
    }
  ],
  "human_action": [
    {
      "agent": "person",
      "action": "action_description"
    }
  ],
  "color": [
    {
      "color": "color_name",
      "object":
      "object_with_that_color"
    }
  ],
  "spatial_relationship": [
    {
      "location": "position (up, down,
      left, right, etc.)",
      "object":
      "object_at_that_position"
    }
  ],
  "scene": ["place1", "place2"]
}
```

Analysis Hints

Keywords and expressions to look for in each evaluation criterion:

- **subject**: person, man, woman, child, animals (dog, cat, bird, etc.), car, bicycle, building, object names
- **background**: background, behind, backdrop, in the back
- **dynamic_degree**:
  - Include: move, run, fly, flow, rotate, travel (object actions)
  - Exclude: zoom in, zoom out, pan, tilt (camera actions)

````
- **human_action**: run, walk, sit,
stand, jump, speak, eat, read, write,
point, touch
- **color**:
  - Color names: red, blue, green,
  yellow, white, black, pink, purple,
  orange
  - Also record corresponding objects
- **spatial_relationship**:
  - Clear positions only: left, right,
  up, down, front, back, upper left,
  upper right, lower left, lower right,
  center, below, above
  - Do not extract: next to, between,
  near, far (ambiguous expressions)
- **scene**:
  - Places/environments: park, street,
  ocean, mountain, river, forest,
  indoor, outdoor, city, nature,
  tunnel, road, room
  - Not objects: car, building, tree
  (these are classified as subject)

## Important Notes

- Extract only information explicitly
stated in the caption
- Avoid speculation or assumptions,
record only certain information
- If no information exists, leave as
empty list []

## Caption to Analyze

Analyze the following caption:
```txt
<<<|SCENE_CAPTION|>>>
```
````

We insert the caption (prompt) of Vript into the $<<<$ |SCENE_CAPTION| $>>>$ placeholder. We set the inference hyperparameters according to the officially recommended parameters [39]: top-p=0.95, top-k=20, temperature=0.6, and min-p=0. Additionally, we set a repetition-penalty of 1.2 to suppress repetition in the output.

E.2. Degradation algorithms

We performed degrading operations corresponding to each evaluation aspect using the following procedures.

Aesthetics. To synthesize v_p^-, we apply FFmpeg's `eq` filter to five clips from the original video with the `contrast` option set to -0.8. In this process, the luminance (luma, Y) component is first inverted ($Y \rightarrow 1 - Y$), and then its dynamic range is compressed to 80%. The resulting video v_p^- has a dull, flat appearance with reduced contrast.

Technical Quality. To synthesize v_p^-, we first resize all frames of the original video to a unified base resolution where the longer side is 512 px while preserving the aspect ratio. We then downscale only the clips targeted for degradation to a longer side of 256 px using LANCZOS resampling, again preserving the aspect ratio. Finally, we upscale these clips back to the base resolution (longer side 512 px). As a result, v_p^- has reduced sharpness and a loss of fine details.

Appearance Style. To synthesize v_p^-, we randomly select one of the following five appearance styles for each video: cartoon, detail enhancement, oil painting, colored pencil sketch, and watercolor. We then apply the selected style transformation to all frames in the clips targeted for degradation. Each style transformation is implemented using OpenCV's [3] stylization filters. Details of each transformation are provided below.

Cartoon Style. For the cartoon style transformation, we first smooth color regions using `cv2.edgePreservingFilter` (`flags`=1, `sigma_s`=40, `sigma_r`=0.20). We then convert the image to grayscale with `cv2.cvtColor` and apply `cv2.medianBlur` (kernel size of 5). Next, we generate an edge mask by applying `cv2.adaptiveThreshold` (maximum value of 255, `ADAPTIVE_THRESH_MEAN_C`, `THRESH_BINARY`, `blockSize`=11, `C`=3), followed by another `cv2.medianBlur` (kernel size of 5), and finally combine it with the smoothed image using `cv2.bitwise_and`.

Detail Enhancement. For the detail enhancement style, we apply `cv2.detailEnhance` (`sigma_s`=5, `sigma_r`=0.08) to each frame.

Oil Painting Style. For the oil painting style transformation, we apply `cv2.xphoto.oilPainting` (`size`=5, `dynRatio`=1) to each frame.

Colored Pencil Sketch Style. For the colored pencil sketch style transformation, we apply `cv2.pencilSketch` (`sigma_s`=40, `sigma_r`=0.05, `shade_factor`=0.015).

Watercolor Style. For the watercolor style transformation, we apply `cv2.stylization` (`sigma_s`=40, `sigma_r`=0.25) to each frame.

Background Consistency. To synthesize v_p^-, we apply the following three-step procedure: 1. Randomly sample five clips from the original video, 2. Remove the background from all frames in each sampled clip, 3. Add a new background image to the background-removed frames.

Table 5. Accuracy (%) of CLIPScore under different prompt-handling strategies. Numbers are accuracy % ± 95% CI (percentage points). The chance level is 50%.

	Video Quality				Video-Text Consistency					
	Aesthetics	Technical Quality	Appearance Style	Background Consistency	Object Integrity	Color	Dynamics Degree	Comprehen-siveness	Spatial Relationship	Temporal Flow
IGNMAX	48.9±5.8	76.2±7.3	62.8±5.4	70.9±4.7	71.0±8.9	66.2±6.5	55.3±5.3	51.1±6.4	60.2±6.2	56.1±5.8
EACHTRUNC	54.3±5.8	72.3±7.7	66.0±5.3	76.6±4.4	75.0±8.5	68.1±6.4	55.0±5.3	60.4±6.3	56.4±6.3	54.4±5.8

In step 1, the sampling process is restricted to clips whose captions extracted by Qwen3 contain background-related information. For step 2, we remove the background using the Python library rembg [8], which internally uses U^2-Net [33]. In step 3, we randomly sample new background images from the nature-dataset [29] and paste them onto the background-removed frames. As a result, v_p^- becomes a video whose background is inconsistent across adjacent frames.

Object Integrity. As described in Section 4.3, we degrade object integrity in two steps: (i) detecting the positions of objects mentioned in the caption and (ii) removing these objects by inpainting. We randomly select target objects for deletion based on the information extracted from the caption (see Appendix E.1). To obtain bounding boxes (bbox) of the target objects in each frame, we use IDEA-Research/grounding-dino-base [24]. We first resize each frame so that its longer side is 512 px while preserving the aspect ratio, and then estimate the bounding boxes. During detection, we set the thresholds to box-threshold=0.4 and text-threshold=0.3.

For object removal via inpainting, we use stable-diffusion-v1-5/stable-diffusion-inpainting [34]. Similarly, we resize each frame so that its longer side is 512 px while preserving the aspect ratio, and set denoising-steps to 50 and guidance-scale to 7.5.

Color. To synthesize v_p^-, we use Qwen-Image-Edit-2509 [45], an image editing model to locally change only the colors of objects.

In this degradation procedure, we first randomly sample clips whose captions contain object color attribute information. We extract this attribute information using the method described in Appendix E.1. For every frame in a sampled clip, we use Qwen-Image-Edit to change the color of a target object from its original color to a new one (e.g., "a red car" → "a blue car"). We choose the new color by randomly sampling one color that is different from the original from a predefined set of colors (black, white, red, green, yellow, blue, brown, purple, pink, orange, gray). We provide Qwen-Image-Edit with the prompt `"Change the color of the {object} to {color}."` to instruct it to change only the color of the specified object.

F. Effectiveness of our frame sampling strategy

Many current evaluation systems impose strict input limits (e.g., maximum frames or context length), making it impractical to feed all frames of long videos. Since SLVMEval explicitly targets videos that range from minutes to hours, a sampling strategy is required to make evaluation feasible while retaining coverage over the full temporal span.

F.1. Our sampling design

We adopt a clip-based sampling strategy: we detect clip boundaries using FFmpeg and extract the center frame from each detected clip (described in §5.1 and §5.2). If the resulting frame sequence still exceeds the evaluation system's maximum frame budget, we randomly subsample frames to satisfy the limit. This strategy is widely used in long-video understanding to improve temporal coverage under limited budgets [32, 49].

The key rationale is that uniform per-clip sampling spreads the budget across the entire video, reducing the risk that the evaluation systems sees only early segments. This is particularly important for SLVMEval, because degradations can appear at arbitrary temporal positions.

F.2. Sequence length limits

We evaluate CLIPScore under different prompt-length handling policies to examine whether our conclusions are sensitive to our sampling design. In addition to the default setting(§5.2), we consider two alternatives: (i) IGNMAX; which ignores the model' s maximum sequence length [9] and (ii) EACHTRUNC; which evenly truncates each clip caption so that the final concatenated prompt fits within the model' s maximum length. Per-aspect results are reported in Table 5.

As a result, the Average CLIPScore accuracy (%) across all aspects was DEFAULT (§5.2): 60.7%, IGNMAX: 60.5%, and EACHTRUNC: 62.8%, indicating no substantial differences.

[9] We cap the input at 31,000 tokens due to GPU memory constraints

Table 6. Correspondence between aspect, dimension, and description. The dimension and description are included in the prompt.

aspect	dimension	description
Aesthetics	aesthetics	This viewpoint indicates whether each frame of the video exhibits overall visual appeal, including composition, harmony, and photographic quality.
Technical Quality	technical_quality	This viewpoint indicates whether each frame is free from technical artifacts (e.g., blur, noise, compression artifacts, over- or under-exposure).
Appearance Style	appearance_style	This viewpoint indicates whether the required appearance style in the text prompt (e.g., oil painting style, black-and-white style, watercolor style, cyberpunk style) is appropriately represented.
Background Consistency	background_consistency	This viewpoint indicates whether the background remains visually stable and consistent across frames.
Object Integrity	object_removal	This viewpoint indicates whether a specific object (e.g., dog, car) mentioned in the text prompt is correctly generated.
Color	color	This viewpoint indicates whether the colors specified in the text prompt (for instance, the color of an object) are correctly reproduced.
Dynamics Degree	dynamics_degree	This viewpoint indicates whether the generated video contains an appropriate amount of motion (neither too static nor overly erratic).
Comprehensiveness	scene	This viewpoint indicates whether the overall scene (for example, “ocean” when prompted with “ocean”) is generated correctly according to the text prompt.
Spatial Relationship	spatial_relationship	This viewpoint indicates whether generated objects follow the text prompt’s spatial instructions across four relation types: Horizontal, Vertical, Proximity/Adjacency, and Depth/Occlusion.
Temporal Flow	move_scene	This viewpoint assesses whether the generated video preserves a coherent, chronological flow of scenes, avoiding unintended shuffling, repetition, or time-jumps-in line with the order implied by the text prompt.

G. Details of baseline systems

G.1. Pairwise VLM-as-a-judge

We used three VLMs: GPT-5 (gpt-5-2025-08-07), GPT-5-mini (gpt-5-mini-2025-08-07), and Qwen3 (Qwen/Qwen3-VL-235B-A22B-Thinking-FP8). For all models, we performed inference using the Structured Outputs mode to enforce JSON-formatted outputs. We set the inference hyperparameters for Qwen3-VL to top-p = 0.8, top-k = 20, temperature = 0.7, and min-p = 0, and for GPT-5 we used temperature = 1.0.[10]

[10]We avoided greedy decoding for Qwen3 inference because it is deprecated to prevent repetition. Additionally, for GPT-5 we used a temperature of 1.0 because it is the only supported and fixed value.

H. Correlation between accuracy and video duration

In the main paper (Figure 3), we analyzed how the accuracy of each evaluation system changes as a function of video duration. Here we provide a more detailed analysis based on Spearman rank correlation coefficients ρ_S between video duration and accuracy for each evaluation aspect and each evaluation system.

For every aspect–system pair, we follow the procedure described in the caption of Figure 3. Table 7 reports the resulting ρ_S values, and Table 8 reports the corresponding

Table 7. Spearman rank correlation coefficients ρ_S between video duration and accuracy for each evaluation system (rows) and evaluation aspect (columns). For each aspect and system, we sort the test samples by video duration, divide them into 50 bins, compute the accuracy within each bin, and then compute Spearman's ρ_S between the bin index (corresponding to video duration) and the bin-wise accuracy. Negative values indicate that accuracy tends to decrease as video duration increases.

	Video Quality				Video-Text Consistency					
	Aesthetics	Technical Quality	Appearance Style	Background Consistency	Object Integrity	Color	Dynamics Degree	Comprehen-siveness	Spatial Relationship	Temporal Flow
Video-based										
GPT-5	-0.6806	-0.4930	-0.1150	0.0487	0.0541	-0.1198	0.1446	-0.2200	-0.1216	-0.4927
GPT-5-mini	-0.5422	-0.3558	-0.2473	-0.2874	-0.2409	-0.3055	-0.0402	-0.2239	-0.1750	-0.3623
Qwen3	-0.1368	-0.0572	-0.0636	-0.2014	0.3848	-0.0683	0.1418	0.0200	-0.2934	0.1239
Text-based										
GPT-5	-0.4650	-0.1872	-0.3206	-0.5072	-0.2892	-0.3270	0.1386	-0.3076	-0.0776	-0.4547
GPT-5-mini	-0.4985	-0.2639	-0.3791	-0.6065	-0.2899	-0.3620	0.1504	-0.2847	-0.0541	-0.4107
Qwen3	-0.0713	-0.2954	-0.4476	-0.6920	-0.1338	-0.5632	0.0020	-0.0061	-0.3756	-0.4559
CLIPScore	-0.2555	-0.0218	-0.2681	-0.4483	-0.2538	-0.7167	-0.0716	-0.2165	-0.1322	-0.1484
VideoScore	0.1677	-0.0094	-0.4636	-0.2936	-0.3749	-0.4406	-0.2625	-0.4824	0.3212	-0.1139
Human	-0.3847	-0.3902	0.2202	0.2360	-0.1905	0.0158	-0.4254	-0.3855	-0.4621	0.0845

Table 8. Two-sided p-values for the Spearman rank correlations in Table 7. Each entry is rounded to four decimal places. Bold values indicate statistically significant correlations with $p < 0.05$.

	Video Quality				Video-Text Consistency					
	Aesthetics	Technical Quality	Appearance Style	Background Consistency	Object Integrity	Color	Dynamics Degree	Comprehen-siveness	Spatial Relationship	Temporal Flow
Video-based										
GPT-5	**0.0000**	**0.0003**	0.4263	0.7371	0.7090	0.4073	0.3164	0.1248	0.4002	**0.0003**
GPT-5-mini	**0.0000**	**0.0112**	0.0834	**0.0430**	0.0919	**0.0310**	0.7818	0.1181	0.2241	**0.0097**
Qwen3	0.3434	0.6934	0.6606	0.1607	**0.0058**	0.6373	0.3259	0.8902	**0.0387**	0.3915
Text-based										
GPT-5	**0.0007**	0.1929	**0.0232**	**0.0002**	**0.0416**	**0.0205**	0.3369	**0.0298**	0.5922	**0.0009**
GPT-5-mini	**0.0002**	0.0640	**0.0066**	**0.0000**	**0.0411**	**0.0098**	0.2970	**0.0451**	0.7090	**0.0030**
Qwen3	0.6226	**0.0373**	**0.0011**	**0.0000**	0.3542	**0.0000**	0.9889	0.9664	**0.0072**	**0.0009**
CLIPScore	0.0733	0.8806	0.0598	**0.0011**	0.0753	**0.0000**	0.6213	0.1311	0.3601	0.3036
VideoScore	0.2443	0.9484	**0.0007**	**0.0385**	**0.0073**	**0.0014**	0.0656	**0.0004**	**0.0229**	0.4311
Human	**0.0058**	**0.0051**	0.1244	0.0990	0.1851	0.9132	**0.0021**	**0.0057**	**0.0007**	0.5595

two-sided p-values.

Overall, most automatic evaluation systems exhibit negative correlations on many aspects, indicating that their accuracy tends to decrease as video duration increases. This tendency is especially pronounced for aspects such as *Background Consistency*, *Color*, and *Temporal Flow*, where several systems show relatively large negative ρ_S with statistically significant p-values ($p < 0.05$). In contrast, for aspects such as *Dynamics Degree*, many systems already perform near chance level for both short and long videos. As a result, the correlations are small and often not statistically significant.

We also report the correlation values for human evaluators. Humans show only weakly negative correlations, and their p-values are generally larger than those of the automatic systems, reflecting that human performance remains high and relatively stable across different video durations. This further highlights the gap between human robustness and the current limitations of automatic evaluation systems for T2LV generation.

Video-based evaluation. The prompt template for the video-based evaluation is as follows. The `input_text` placeholder holds the input prompt for the T2V model. See Section H.1 for the text used for the `dimension` and `description` placeholders.

```
[System prompt]
You are a meticulous AI evaluator for
video.
You will be provided with the
evaluation viewpoint and its
description, the text used for
generation, and frames of the videos
generated by two different videos.
```

Algorithm 2 Implementation of Video-based evaluation

Require: Prompt p, video u_p, video v_p, aspect a, maximum number of frames $F_{\max}$
Ensure: u_p or v_p

$u_p^s \leftarrow \text{SAMPLEFRAMES}(u_p)$
$v_p^s \leftarrow \text{SAMPLEFRAMES}(v_p)$
for $s \in \{u_p^s, v_p^s\}$ **do**
  if $|s| > F_{\max}$ **then**
    $s \leftarrow \text{RANDOMSAMPLE}(s, F_{\max})$
  end if
end for
$d \leftarrow \text{VLM}(a, p, u_p^s, v_p^s)$ ▷ Return "first" or "second"
if $d =$ "first" **then**
  return u_p^s
else if $d =$ "second" **then**
  return v_p^s
end if

Algorithm 3 Implementation of Text-based evaluation

Require: Prompt p, video u_p, video v_p, aspect a, maximum number of frames $F_{\max}$
Ensure: u_p or v_p

$u_p^s \leftarrow \text{SAMPLEFRAMES}(u_p)$
$v_p^s \leftarrow \text{SAMPLEFRAMES}(v_p)$
for $s \in \{u_p^s, v_p^s\}$ **do**
  if $|s| > F_{\max}$ **then**
    $s \leftarrow \text{RANDOMSAMPLE}(s, F_{\max})$
  end if
end for
$caption_{u_p} \leftarrow \text{VLM}_{\text{cap}}(a, u_p^s)$
$caption_{v_p} \leftarrow \text{VLM}_{\text{cap}}(a, v_p^s)$
$d \leftarrow \text{LM}(a, p, caption_{u_p}, caption_{v_p})$ ▷ Return "first" or "second"
if $d =$ "first" **then**
  return u_p
else if $d =$ "second" **then**
  return v_p
end if

```
Based on the given viewpoint, compare
the quality of the two videos in
relation to the input text.
Finally, output which video, 'first
video' or 'second video', has the
higher quality in JSON format.

[User prompt]
Given the provided information below,
please evaluate which video, the first
video or the second video, generated a
higher-quality video.

Viewpoints and descriptions: '''
- {dimension}: {description}
'''

Text input used for generation: '''
{input_text}
'''
```

Algorithm 2 presents the pseudocode for the entire procedure.

Text-based evaluation. The prompt template for the text-based evaluation is as follows. The `input_text` placeholder contains the input prompt for the T2V model, while the `first_model_text` and `second_model_text` placeholders hold the generated captions for the respective videos. See Section H.1 for the text used for the `dimension` and `description` placeholders.

```
[System prompt]
You are a meticulous AI evaluator for
video.
You will be provided with the
evaluation viewpoint and its
description, the text used for
generation, and transcriptions of the
videos generated by two different
videos.
Based on the given viewpoint, compare
the quality of the two videos in
relation to the input text.
Finally, output which video, 'first
video' or 'second video', has the
higher quality in JSON format.

[User prompt]
Given the provided information below,
please evaluate which video, the first
video or the second video, generated a
higher-quality video.

Viewpoints and descriptions: '''
- {dimension}: {description}
'''

Text input used for generation: '''
{input_text}
'''

Transcription of the video generated by
the first video: '''
{first_model_text}
'''
```

Table 9. Computing resources used in our experiments.

	CPU resource	**GPU resource**
CPU Model	2 x Intel Xeon Platinum 8490H (Sapphire Rapids)	2 x Intel Xeon Platinum 8490H (Sapphire Rapids)
GPU Model	None	4 x NVIDIA H100 (HBM2e)
System Memory (per node)	512 GiB (DDR5)	1 TiB (DDR5)
GPU Memory (per node)	N/A	94 GiB per GPU
Operating System	Rocky Linux	Rocky Linux

```
Transcription of the video generated by
the second video: '''
{second_model_text}
'''
```

Algorithm 3 presents the pseudocode for the entire procedure.

Captioning. The prompt we use to generate caption in the text-based and hybrid evaluations is as follows.

```
[System prompt]
You are an assistant that converts
video content into a thorough textual
description.
You will be given one or more frames
from a video (or a description of
them).
Your goal is to accurately describe all
visible details and actions, without
adding information that cannot be
inferred from the video.
For each provided evaluation viewpoint,
please represent the video content as
text and output the result in JSON
format.

Important notes: '''
- Only include details that can be
directly observed from the given
frames.
- Do not speculate or assume any
information not clearly visible.
- Describe objects, people, scenes,
backgrounds, movements, aesthetics,
video quality and any relevant visual
details.
- Be concise and factual. Avoid
subjective or interpretive language.
- If multiple frames are provided,
indicate changes or continuity across
them.
- Output your description in a
structured JSON format.
'''

Viewpoints and descriptions: '''
Viewpoint: {aspect}
Description: {description}
'''

[User prompt]
Please generate descriptions based on
the evaluation viewpoints for the given
videos.
```

Table 10. Mapping between our defined aspects and those defined in VideoScore.

Our aspect	VideoScore aspect
Aesthetics	visual quality
Technical Quality	visual quality
Appearance Style	visual quality
Background Consistency	factual consistency
Object Integrity	text-to-video alignment
Color	visual quality
Dynamics Degree	dynamic degree
Comprehensiveness	text-to-video alignment
Spatial Relationship	text-to-video alignment
Temporal Flow	text-to-video alignment

H.1. Prompt component mapping

Table 6 shows the correspondence between the evaluation aspect and the `dimension` and `description` text provided to the evaluation system.

H.2. VideoScore

As described in Section 5.3, we use VideoScore-v1.1 [11] as one of our baseline evaluation systems. Because the aspects we consider differ from those in VideoScore, we construct a mapping between our aspects and those of VideoScore. This mapping is summarized in Table 10.

I. Computational Resources

We used "Genkai," a computational resource provided by Kyushu University, for our experiments. Table 9 details the computational resources we use.

J. Ethics statement

We outsourced data annotations to crowd workers on Yahoo! Crowdsourcing. Before participation, we explained the purpose of the task to all workers and obtained their informed consent. We did not collect any personally identifiable information, and we paid workers at rates above the legal minimum wage in the country where the platform operates.

K. Limitations

The proposed SLVMEval benchmark benchmark utilizes a synthetic dataset; thus, its distribution differs from the outputs expected from future T2LV models and may not cover all failures that those models could produce. Therefore, additional benchmarks are required to evaluate models under more complex and realistic settings. Such benchmarks will need to be developed in parallel with advancements in T2LV generation models because their design will depend on analyzing model failure patterns. Thus, we anticipate further progress in this research area.

In addition, the settings of the degrading operations, e.g., the magnitude of the contrast changes and the fraction of clips degraded, function as hyperparameters, and varying these hyperparameters is expected to change the difficulty of the benchmark. Currently, the benchmark is easy for humans to judge, which suits the evaluation goal considered in the current study; however, creating variants with different difficulty levels would enable broader analysis.