Struct-VidOCR: Instance-Centric OCR Evidence for Video Captioning via RL Alignment

Anonymous submission

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Abstract

While Vision Language Models (VLMs) excel at static image OCR, Video OCR remains difficult because text moves, is occluded, and updates over time, disrupting persistent tracking and causing temporal hallucinations. We propose **Struct-VidOCR** with **Structured OCR Segments**, explicit lifecycle records that preserve transcription, natural-language position, sampled temporal span, semantic and source attributes, motion, occlusion, persistence, and content variants as reusable OCR evidence for a self-contained caption. We align this record–caption output using Group Relative Policy Optimization (GRPO) and a deterministic matching-derived reward over OCR fidelity, temporal and attribute grounding, and caption consistency. Trained on only 1.5k automatically quality-controlled pseudo-labeled clips, our 8B model improves Qwen3-VL-8B by 2.5 points on MME-VideoOCR and 3.7 points on the caption-compatible VidText subset, while achieving performance comparable to Qwen3-VL-235B-A22B-Instruct. These results use rigorous Caption-Injected QA: a fixed text-only reader answers benchmark questions from the generated caption without video frames; the predicted record provides additional downstream utility. Thus, Struct-VidOCR validates a data-efficient, structurally aligned video-to-text evidence interface under this controlled readout.

Introduction

The rapid advancement of Vision Language Models (VLMs) has revolutionized image OCR via breakthroughs such as DeepSeek-OCR (Wei, Sun, and Li 2025), PaddleOCR-VL (Cui et al. 2025), HunyuanOCR (Team et al. 2025b), and other works (Anonymous 2025; Li et al. 2025b, 2024). However, leveraging VLMs for general Video OCR—specifically the precise extraction, tracking, and reasoning of text in videos—remains largely underexplored. Early frameworks such as EVE (Yu et al. 2025) are confined to subtitles, leaving in-the-wild scene text unaddressed. Furthermore, while models such as Video-R4 (Tang et al. 2025) introduce Chain-of-Thought (CoT) reasoning for text-rich videos, they still struggle to continuously extract and track raw 4D spatiotemporal text dynamics.

This is an anonymized submission for review purposes only. Distribution, citation, or public sharing of this manuscript is strictly prohibited. Copyright and publication details will appear in the final version if accepted.

Tracking dynamic scene text traditionally falls to Video Text Spotting (VTS) methods such as YORO (Cheng et al. 2019), VimTS (Liu et al. 2025), and others (Wu et al. 2024; Liu et al. 2024; He et al. 2025), which lack the semantic reasoning needed to synthesize coherent narratives. Conversely, standard VLMs fail at Video OCR because they typically treat videos as frame sequences rather than tracking text lifecycles. Following the *instance-oriented* paradigm of GAT (Zhang et al. 2025b), we argue that scene text is a complex 4D spatiotemporal entity undergoing temporary occlusions, perspective shifts, and content updates. A flat frame-level text list leaves cross-frame identity, disappearance/reappearance, and sequential updates unresolved, while a caption can compress those distinctions away. We contend that forcing VLMs to implicitly learn these lifecycle relations through unstructured natural language is fundamentally inefficient. Compounding this representation gap is a severe data scarcity issue: collecting frame-accurate, temporally grounded video OCR annotations is notoriously expensive and difficult.

To overcome these intertwined challenges of inefficient representations and data scarcity, we propose **Struct-VidOCR**. First, we introduce **Structured OCR Segments**, an explicit formulation that transforms frame-centric reading into instance-centric tracking by recording each text entity's lifecycle as reusable evidence: transcription, natural-language image-plane position, sampled temporal span, semantic/source attributes, motion, occlusion, persistence, and time-indexed content variants. The record subsequently supports a standalone OCR-enriched caption, so the retained evidence is useful to a text-only reader rather than remaining an auxiliary endpoint. Second, to address data scarcity, we automate the curation of a dense, high-quality dataset of **1.5k video samples** through a two-stage record-then-caption distillation pipeline with input-quality controls. Finally, to empower the VLM to master these complex structured representations, we leverage multimodal Reinforcement Learning (Feng et al. 2025b; Tang et al. 2025; Shao et al. 2024; Meng et al. 2025a; Zhang et al. 2025a), employing **Group Relative Policy Optimization (GRPO)** (Shao et al. 2024) to foster robust structured spatiotemporal tracking and evidence-to-caption generation.

This combination of a structured data format, high-quality curated data, and GRPO-based alignment serves as a strong

inductive bias. Our text-first bipartite matching establishes correspondence among unordered autoregressive records, after which OCR fidelity, sampled temporal spans, lifecycle attributes, and caption consistency are scored; unmatched predictions and targets expose hallucinations and omissions. The resulting interface does not aim to be a coordinate-level spotter or a new end-to-end video-QA model. Instead, it targets a persistent, reusable OCR evidence representation paired with a reader-facing caption. By fine-tuning a Qwen3-VL-8B-Instruct model on merely 1.5k samples, our approach yields a substantial performance leap over the baseline. Evaluated under a strict "Caption-Injected QA" protocol, in which a fixed text-only reader receives the benchmark question and generated caption but no video frames, Struct-VidOCR improves Qwen3-VL-8B by 2.5 points on MME-VideoOCR and 3.7 points on the caption-compatible VidText subset, while achieving performance comparable to the 235B-parameter Qwen3-VL-235B-A22B-Instruct. These results establish the value of data-efficient VLM alignment for the stated video-to-text evidence interface, rather than claiming native end-to-end QA performance.

In summary, our main contributions are as follows:

- We address the underexplored domain of Video OCR in VLMs by proposing **Structured OCR Segments**, an explicit data structure that bridges instance-level 4D text tracking and a reusable OCR evidence representation for caption generation.
- We introduce an automated data pipeline to construct a small, high-quality dataset (1.5k samples) and apply **GRPO** with a deterministic text-first matching reward that jointly aligns OCR fidelity, sampled temporal spans, lifecycle attributes, and caption consistency.
- We demonstrate that our approach improves the extraction, structured description, and caption-level recoverability of video-text evidence. With just 1.5k samples, the 8B-parameter Struct-VidOCR yields significant gains over its baseline and performance comparable to the 235B-parameter Qwen3-VL-235B-A22B-Instruct under the controlled Caption-Injected QA protocol, while structured readout offers a separate downstream-utility diagnostic.

Related Work

Video Text Understanding

While Vision-Language Models (VLMs) have revolutionized static image OCR (e.g., DeepSeek-OCR (Wei, Sun, and Li 2025), PaddleOCR-VL (Cui et al. 2025) and others (Li et al. 2025c; Wei et al. 2024b; Hu et al. 2024; Feng et al. 2025a; Zhang et al. 2023; Wei et al. 2024a; Li et al. 2024; Anonymous 2025; Li et al. 2025b; Team et al. 2025b)), extending this success to video text understanding remains a formidable challenge due to complex 4D dynamics. Traditional Video Text Spotting (VTS) frameworks, such as YORO (Cheng et al. 2019), VimTS (Liu et al. 2025) and others (Wu et al. 2024; Liu et al. 2024; He et al. 2025), effectively track text coordinates but lack the cognitive architecture required for multimodal reasoning. Recent research has begun to shift towards an instance-oriented perspective; GAT (Zhang et al. 2025b) emphasizes tracking text lifecycles over isolated frames, and EVE (Yu et al. 2025) achieves end-to-end subtitle extraction. Pushing cognitive boundaries further, Video-R4 (Tang et al. 2025) employs visual rumination to enhance reasoning in text-rich videos. However, these generalist models primarily optimize for end-to-end QA and struggle to explicitly model fine-grained physical states, such as occlusion, persistence, and motion. Our work bridges this gap by introducing an explicitly structured intermediate representation.

Reinforcement Learning in VLMs

To elicit deeper reasoning capabilities and overcome the limitations of standard Supervised Fine-Tuning (SFT), recent works including (Guo et al. 2025; Feng et al. 2025b; Meng et al. 2025a; Li et al. 2025a; Ouyang et al. 2025; Zhang et al. 2025a; Meng et al. 2025b) have increasingly adopted Reinforcement Learning (RL) for multimodal alignment. However, standard RLHF pipelines typically rely on learned reward models, which are notoriously unstable and prone to reward hacking—particularly in strict, objective tasks like Video OCR where precision is paramount. We address this by combining explicit structural modeling with Group Relative Policy Optimization (GRPO). By utilizing a deterministic matching reward based on the Jonker–Volgenant (Jonker and Volgenant 1987) algorithm, we ground the VLM in the physical dynamics of video text without reliance on a learned critic.

Methodology

Struct-VidOCR trains a VLM to produce an OCR evidence record and a temporally aware caption. As illustrated in fig. 1, the intermediate representation turns video text into persistent instances before the caption verbalizes the evidence. The central novelty is therefore not JSON formatting alone, but the coupling of lifecycle representation, permutation-invariant alignment, and evidence transfer into a self-contained caption. This output interface is designed for videos whose text changes, moves, or is intermittently visible.

Data Construction

Detailed filters and the manual spot audit are reported in the appendix.

For each retained clip, Gemini 2.5 Pro (Comanici et al. 2025) first analyzes frames sampled at 2 FPS and emits the schema-targeted record Y_{struct}. It groups observations into Structured OCR Segments, including sampled temporal spans, source and motion attributes, occlusion, persistence, and time-indexed variants when content changes. A second prompt conditions on this record to generate the standalone OCR-enriched caption Y_{cap}. This decomposition defines the evidence-to-caption supervision interface: the record preserves the selected OCR evidence before the caption verbalizes it. We use the resulting 1.5k record-caption pairs for GRPO alignment; prompts and generation details are deferred to the appendix.

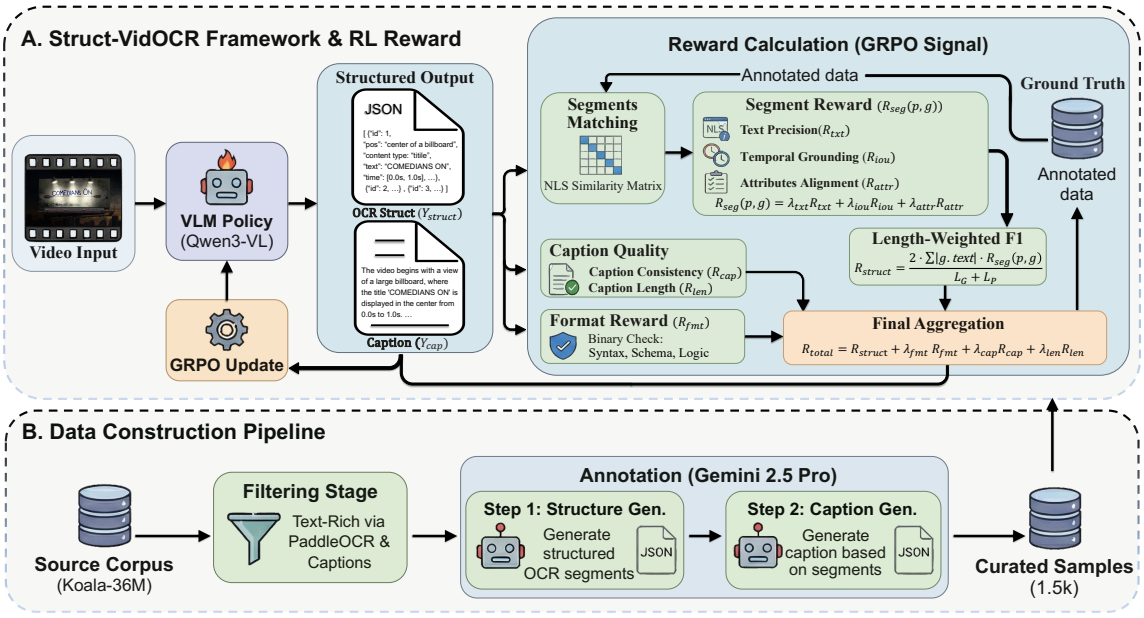


Figure 1: Struct-VidOCR. (A) GRPO aligns the VLM with a deterministic, multi-aspect reward over an evidence record and caption. (B) The data pipeline applies automatic input-quality controls before Gemini 2.5 Pro pseudo-labeling.

Structured OCR Segments

We define a *Structured OCR Segment* S as a tuple describing one text instance across sampled video frames. It records natural-language image-plane position and temporal extent, rather than 3D geometry or numerical box coordinates. Formally:

$$S = \{\text{ID}, \mathbf{p}, \mathbf{t}_{start}, \mathbf{t}_{end}, \mathcal{C}_{content}, \mathcal{T}_{source}, \mathcal{M}_{move}, \mathcal{O}_{occ}, \mathcal{P}_{persist}, \text{Text}\}$$

The attributes encode complementary lifecycle evidence:

- **Instance identity (ID):** Provides output-level bookkeeping for records that may share the same transcription in different contexts; text-first reward matching can remain ambiguous for otherwise identical records.
- **Spatiotemporal anchoring ($\mathbf{p}, \mathbf{t}_{start}, \mathbf{t}_{end}$):** $\mathbf{p}$ is a natural-language image-plane position. Timestamps are sampled at 2 FPS and should be interpreted at this sampling resolution.
- **Semantic and source categorization ($\mathcal{C}_{content}, \mathcal{T}_{source}$):**
  - $\mathcal{C}_{content}$ denotes the semantic role (e.g., "subtitle" or "scoreboard").
  - $\mathcal{T}_{source}$ distinguishes scene text from burned-in graphics.
- **Motion and occlusion ($\mathcal{M}_{move}, \mathcal{O}_{occ}$):**
  - $\mathcal{M}_{move}$ categorizes motion (e.g., *moving_with_object*, *static*, or *fixed_on_screen*).
  - $\mathcal{O}_{occ}$ records occlusion (e.g., *partial* or *severe*) to retain evidence across interruptions.
- **Content evolution ($\mathcal{P}_{persist}$, Text):** Text at a fixed location can change (e.g., clocks, tickers, and scoreboards).
- $\mathcal{P}_{persist}$ flags continuous versus sequentially changing text.
- Sequential-change representation: If $\mathcal{P}_{persist}$ = "sequential_change", the **Text** field is not a string but a list of time-stamped variants:

  $$V = \{(txt_1, t_{1_{in}}, t_{1_{out}}), (txt_2, t_{2_{in}}, t_{2_{out}}), \dots\}$$

  This representation keeps sequential updates in one lifecycle record rather than fragmenting them into disconnected observations.

This representation makes lifecycle evidence available to both the structured output and the caption.

GRPO-based Structured Representation Alignment

We optimize the JSON-caption dual output with deterministic reward signals.

Group Relative Policy Optimization (GRPO) We use Group Relative Policy Optimization (GRPO), which samples a group of G outputs for each input and normalizes every output's reward relative to the group statistics, avoiding a separate value network. For reward r_i of sample i, the token-level advantage is shared from the normalized sample reward:

$$\hat{A}_{i,t} = \frac{r_i - \text{mean}(\{r_1, \dots, r_G\})}{\text{std}(\{r_1, \dots, r_G\})}.$$

This provides a direct training signal for the structured-record and caption pair while keeping the optimization machinery standard. Settings are in the appendix.

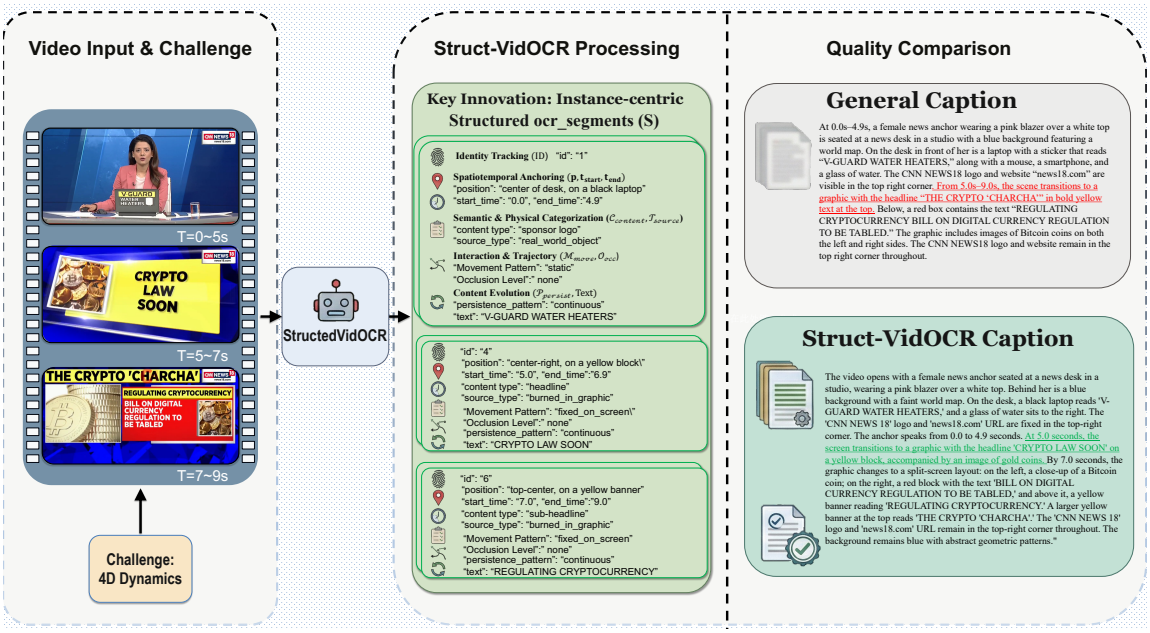


Figure 2: Instance-centric Structured OCR Segments. Relative to a free-form caption, the record retains lifecycle attributes that can be transferred into an OCR-enriched caption.

Target Output Structure The policy emits $y = (Y_{struct}, Y_{cap})$: a schema-targeted set of **Structured OCR Segments** and a **Descriptive Caption**. Y_{struct} is an explicit intermediate evidence representation; Y_{cap} renders its task-relevant evidence as a standalone narrative.

Optimal Segment Matching Predicted segments $P = \{p_1, \ldots, p_N\}$ and target segments $\mathcal{G} = \{g_1, \ldots, g_M\}$ differ in order and cardinality. We therefore use text-first maximum-weight bipartite matching, with normalized Levenshtein similarity between representative transcriptions as its assignment score. The subsequent reward scores temporal spans and attributes for the matched pairs; unmatched predictions and targets are penalized as hallucinations and omissions. This assignment makes set supervision permutation-invariant, rather than proposing a new matching algorithm. Details are in the appendix.

Fine-Grained Segment Reward For every matched pair (p, g), $R_{seg}(p, g) = \lambda_{txt} R_{txt} + \lambda_{iou} R_{iou} + \lambda_{attr} R_{attr}$ combines transcription, temporal IoU sampled, and lifecycle attributes. With $L_G = \sum_g |g.\text{text}|$ and $L_P = \sum_p |p.\text{text}|$, we use $R_{struct} = 2 \sum_{(p,g)\in\pi^*} |g.\text{text}| R_{seg}(p, g)/(L_G + L_P)$. Thus unmatched targets and predictions lower the score as omissions and hallucinations, while matched records receive length-aware partial credit without output-order dependence.

Caption Quality The caption-consistency reward measures coverage of recognized text in Y_{cap}, while a length reward discourages overly brief or verbose captions. These terms prevent OCR evidence from remaining accessible only in an auxiliary JSON field.

Format-Validity Reward We add a binary format-validity reward R_{fmt} for JSON syntax, required data types, and basic temporal consistency; detailed checks are in the appendix.

Total Reward Aggregation Let R_{struct} denote the length-weighted structural alignment reward. The total objective combines it with caption consistency, length, and format validity:

$$R_{total} = R_{struct} + \lambda_{cap} R_{cap} + \lambda_{len} R_{len} + \lambda_{fmt} R_{fmt}.$$

The length weighting reduces the incentive to optimize only short, trivial strings. Full definitions and hyperparameters are in the appendix.

Experiment

Experiment Setups

Implementation Details. We fine-tune **Qwen3-VL-8B-Instruct** (Bai et al. 2025) with GRPO on 1.5k pseudo-labeled Koala-36M clips. Training and inference use 2 FPS sampling, so temporal spans are evaluated at the sampling resolution. Data construction, manual auditing, and training details are deferred to the appendix.

Evaluation Benchmarks. We evaluate representation recoverability on two video-text QA suites. All videos are uniformly sampled at 2 FPS. The trained model uses its training-format prompt; baselines use model-specific prompts designed to elicit the same text-centric objective, namely a chronological OCR-aware description. Thus these are controlled representation-generation comparisons, not every baseline's best published end-to-end QA setting and not evidence of superiority over specialist OCR-plus-VLM cascades. Prompts and decoding details are in the appendix.

- **MME-VideoOCR** (Shi et al. 2025) is a large-scale multimodal benchmark (1,464 videos, 2,000 manually curated

Model	TR	VTQA	TG	AR	CDT	STP	CFTU	TBR	TBVU	RVT	Total
Keye-VL-1.5-8B (Yang et al. 2025a)	56.2	60.1	46.2	56.8	42.8	64.5	28.0	<u>80.3</u>	40.8	**76.7**	55.0
InternVL3.5-8B (Wang et al. 2025)	41.7	58.0	44.0	48.7	42.5	56.0	34.0	76.2	58.1	69.0	51.5
MiMo-VL-7B-RL-2508 (Team et al. 2025a)	52.7	71.1	64.0	**62.4**	**55.5**	67.7	35.4	**80.9**	55.0	65.7	61.1
Qwen3-VL-30B-A3B-Instruct	60.0	72.8	57.0	51.0	51.8	**73.1**	21.2	77.4	61.4	66.7	60.5
Qwen3-VL-32B-Instruct	63.7	72.4	65.5	<u>62.0</u>	**55.5**	68.6	35.1	77.0	59.8	66.2	62.9
Qwen3-VL-235B-A22B-Instruct	**67.0**	<u>73.1</u>	<u>66.1</u>	60.0	51.0	<u>71.1</u>	23.6	78.5	58.5	69.0	<u>63.3</u>
Qwen3-VL-8B-Instruct (Baseline)	58.2	66.9	54.0	56.8	50.0	69.5	35.4	76.4	57.6	66.2	59.7
Struct-VidOCR-8B	63.0	72.0	56.0	57.4	<u>52.0</u>	68.6	<u>37.2</u>	78.0	<u>61.5</u>	69.4	62.2
Struct-VidOCR-8B*	<u>66.4</u>	**74.0**	**68.3**	53.7	51.8	67.2	**38.0**	75.0	**62.8**	<u>70.9</u>	**63.6**

Table 1: MME-VideoOCR fixed-reader caption-readout results, not native video-QA scores. Bold and underline denote best and second-best values, respectively; * is a separate structured-readout diagnostic that additionally provides the predicted record. TR: Text Recognition; VTQA: Visual Text QA; TG: Text Grounding; AR: Attribute Recognition; CDT: Change Detection & Tracking; STP: Special Text Parsing; CFTU: Cross-Frame Text Understanding; TBR: Text-Based Reasoning; TBVU: Text-Based Video Understanding; RVT: Robust Video Testing.

QA pairs across 44 scenarios) for evaluating detection, recognition, temporal tracking, and higher-order comprehension of scene text in dynamic video content.

- **VidText** (Yang et al. 2025b) covers multilingual video-text perception and reasoning. We report its seven caption-compatible tasks; the coordinate-based Text Tracking task is excluded because our output interface has natural-language positions rather than boxes.

Evaluation principle. Caption-Injected QA probes the intended video-to-text product: a question-blind generator produces Y_{cap}, and a fixed text-only reader receives the original question and Y_{cap} but no video frames. With the reader and prompt fixed across systems, scores measure *conditional evidence recoverability*, not native video-QA accuracy. The * setting additionally supplies Y_{struct} as a separate structured-readout diagnostic rather than a caption-only ranking. Controls are in the appendix.

Main Results and Analysis

We compare our Struct-VidOCR model against leading open-source VLMs.

MME-VideoOCR Performance. Table 1 reports the fixed-reader readout. The caption-only Struct-VidOCR row improves the Qwen3-VL-8B baseline from 59.7 to **62.2**, including gains in Text Recognition (+4.8), Text-Based Video Understanding (+3.9), Robust Video Testing (+3.2), and Change Detection & Tracking (+2.0). Under the fixed-reader protocol, these gains show that the full representation-alignment recipe leaves more task-relevant OCR evidence recoverable from the standalone caption. The * row reaches 63.6 when the same reader can additionally access the predicted record; it asks the complementary question of whether that record benefits a downstream text-only consumer.

VidText Performance. On the seven-task VidText subset, the caption-only row improves the average from 29.9 to **33.6**. The structured-readout diagnostic reaches 41.5, with its largest changes in Text Localization (24.7 to 51.5) and Local OCR (16.2 to 23.5). This is expected when the reader can directly use recognized strings together with natural-language positional and temporal evidence from Y_{struct}. The caption-only gains in Spatial Reasoning (+6.7) and Temporal Causal Reasoning (+1.8) show that, under this reader, more of the required evidence is recoverable from Y_{cap}.

Generalizability, Frame-Level Contrast and Caption Quality Beyond the improvements observed on our primary Struct-VidOCR-8B, our proposed data construction pipeline and GRPO training methodology are highly generalizable. Applying the identical training recipe and dataset to other base VLMs yields consistent performance enhancements, demonstrating that our approach is fundamentally model-agnostic. We additionally compare against a frame-level OCR caption control that samples the same frames, inserts specialized-OCR detections into a captioning prompt, and uses the identical text-only reader; its construction and role are detailed in the appendix. Furthermore, we also conducted a comprehensive evaluation of the intrinsic linguistic and descriptive quality of the generated captions. The detailed analyses confirming the quality of our generated captions can also be found in the appendix.

Ablations

Impact of Reward Components The cumulative ablation in table 3 examines how each term changes the evidence-to-caption interface. **Text only** provides basic transcription supervision but no direct temporal or attribute signal. Adding **attributes** raises the caption-only total from 54.8 to 55.5 by supervising position, source, and lifecycle categories. Adding the **temporal** term yields a strong * structured readout (62.7) but a weak caption-only score (46.9), showing that improving the auxiliary record does not automatically produce a usable narrative. **Caption consistency** restores the caption-only total to 60.7 by requiring recognized strings to be exported into Y_{cap}. Finally, the **length** term discourages both truncated and excessively verbose captions, producing the strongest caption-only total of 62.2. Full per-task results are provided in table 7.

Model	HoliOCR	HoliRea.	LocalOCR	LocalRea.	TextLocal.	TempCauRea.	SpaRea.	Avg.
Keye-VL-1.5-8B (Yang et al. 2025a)	23.2	41.6	11.3	<u>52.3</u>	11.9	26.3	29.8	28.1
InternVL3.5-8B (Wang et al. 2025)	22.6	43.2	9.4	51.1	8.9	25.4	28.5	27.0
MiMo-VL-7B-RL-2508 (Team et al. 2025a)	23.1	53.7	16.9	48.8	21.1	28.3	34.2	32.3
Qwen3-VL-30B-A3B-Instruct	24.9	53.2	17.0	48.5	23.4	29.7	32.2	32.7
Qwen3-VL-32B-Instruct	25.2	<u>54.6</u>	<u>17.1</u>	50.3	21.3	31.1	<u>42.6</u>	34.6
Qwen3-VL-235B-A22B-Instruct	<u>25.8</u>	**57.6**	14.7	49.8	20.1	**33.8**	**44.0**	<u>35.1</u>
Qwen3-VL-8B-Instruct (Baseline)	24.7	51.2	12.5	48.6	12.3	28.7	31.6	29.9
Struct-VidOCR-8B	24.9	51.2	16.2	49.5	<u>24.7</u>	30.5	38.3	33.6
Struct-VidOCR-8B*	**32.6**	52.2	**23.5**	**56.0**	**51.5**	<u>33.4</u>	41.5	**41.5**

Table 2: VidText fixed-reader readout results, not native video-QA scores, on seven caption-compatible tasks. Bold and underline denote best and second-best values, respectively; * is a separate structured-readout diagnostic that additionally supplies the predicted record. HoliOCR: Holistic OCR; HoliRea.: Holistic Reasoning; LocalOCR: Local OCR; LocalRea.: Local Reasoning; TextLocal.: Text Localization; TempCauRea.: Temporal Causal Reasoning; SpaRea.: Spatial Reasoning. Avg. excludes coordinate-based Text Tracking.

Metric Design Choices We investigate two critical design choices: the thresholding strategy of NLS and the length weighting mechanism.

Continuous NLS vs. Thresholded: We initially hypothesized that filtering out low-similarity alignments using a hard threshold (τ_{match}) would prevent noisy matching, restricting the optimal assignment space to:

$$\pi^* = \underset{\pi \subset \{(i,j)|W_{ij} \geq \tau_{match}\}}{\arg\max} \sum_{(i,j)\in\pi} W_{ij}$$

Table 4 shows that continuous NLS reaches the strongest caption-only total (62.2) among these variants; its separate * diagnostic is reported alongside, not ranked with caption-only rows. This result is consistent with reward sparsity during early RL exploration. When predicted strings are still noisy, a hard threshold discards partially correct matches and creates broad zero-reward regions. Continuous similarity instead supplies incremental feedback when even a subset of characters is corrected, preserving a denser optimization signal without claiming that every low-similarity assignment is semantically correct.

Length Weighted: In our structural alignment formulation (R_{struct}), the reward is weighted by the character length of the ground truth text. Without this weighting, the model easily games the reward system by overly focusing on short, trivial text segments (e.g., single digits, isolated watermarks) while ignoring long, informative sentences. Table 4 demonstrates that omitting length weighting (*w/o Len-W*) drops overall performance from 62.2 to 61.6. Notably, length weighting provides substantial gains on tasks requiring deep comprehension of complex text blocks, such as Text-Based Reasoning (TBR, $+6.2$) and Change Detection & Tracking (CDT, $+4.6$).

Qualitative Illustration

To intuitively demonstrate the effectiveness of our approach, Figure 3 presents a qualitative comparison between the baseline Qwen3-VL-8B-Instruct and Struct-VidOCR. While the baseline struggles with fine-grained temporal tracking of rapidly changing text and often hallucinates initial states, our model explicitly grounds these dynamics through structured `ocr_segments`. The static logo, temperature reading, and RPM label remain independent persistent records, whereas the speed, gear, and clock are represented by time-indexed variants within their respective lifecycles. This distinction avoids turning a recurring text instance into an unrelated item at every sampled frame, while preserving the observed order needed by a caption. The conditioned caption then transfers the retained updates into a concise chronological account. By successfully capturing the "sequential_change" persistence pattern, Struct-VidOCR generates a significantly more comprehensive and chronologically faithful caption. Additional qualitative results across diverse video scenarios are provided in the appendix.

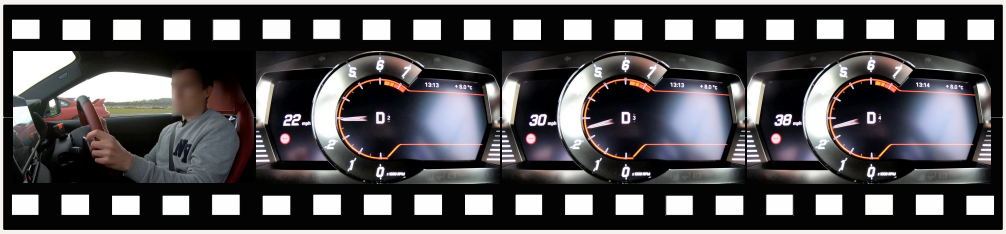
Cumulative reward	Caption	* Readout
R_{txt}	54.8	62.5
$+R_{attr}$	55.5	63.2
$+R_{iou}$	46.9	62.7
$+R_{cap}$	<u>60.7</u>	<u>63.2</u>
$+R_{len}$	**62.2**	**63.6**

Table 3: MME-VideoOCR total for the cumulative reward ablation. Bold and underline denote best and second-best values within each column; "Caption" uses Y_{cap} only, while * additionally supplies the predicted record. Full per-task results are in table 7.

Variant	Caption	* Readout
Thresholded (0.5)	61.2	62.9
Thresholded (0.25)	<u>61.7</u>	<u>63.2</u>
w/o Len-W	61.6	63.1
Continuous + Len-W	**62.2**	**63.6**

Table 4: MME-VideoOCR total for matching and length-weighting variants. Bold and underline denote best and second-best values within each column. Full per-task results are in table 8.

Qualitative Case Study



Qwen3-VL-8B-Instruct: The video begins with a view from inside a car, showing a person wearing a gray sweatshirt with a large "NY" logo on the chest, seated in the driver's seat with their hands on the steering wheel. The car's interior features red leather seats and a modern dashboard. Through the windshield, a red car is visible parked outside, with a grassy field and some trees in the background. The scene then transitions to a close-up of the car's digital instrument cluster, which displays the time as 13:13 and the temperature as +8.0° C. The speedometer starts at 25 mph and gradually increases to 38 mph, while the gear indicator changes from D3 to D4. The tachometer shows RPMs increasing as the car accelerates. The digital display remains consistent in its layout, with the time and temperature displayed in the upper right corner, and the speed and gear information prominently shown in the center.

Struct-VidOCR:

{"ocr_segments": [
{"id": "1",
"position": "center of the driver's gray sweatshirt",
"content_type": "logo",
"source_type": "real_world_object",
"movement_pattern": "moving_with_object",
"persistence_pattern": "continuous",
"text": "NY",
"start_time": 0.0,
"end_time": 1.0,
"occlusion_level": "none"},

{"id": "2",
"position": "left side of dashboard",
"content_type": "speedometer reading",
"source_type": "real_world_object",
"movement_pattern": "static",
"persistence_pattern": "sequential_change",
"variants": [
{"text": "22 mph", "start_time": 1.0,"end_time":2.0},
{"text": "30 mph","start_time": 2.0,"end_time": 3.0},
{"text": "38 mph","start_time": 3.0,"end_time": 4.0}],
"occlusion_level": "none"},

{"id": "3",
"position": "top right of dashboard, right of clock",
"content_type": "temperature reading",
"source_type": "real_world_object",
"movement_pattern": "static",
"persistence_pattern": "continuous",
"text": "+8.0° C",
"start_time": 1.0,
"end_time": 4.0,
"occlusion_level": "none"},

{"id": "4",
"position": "bottom center of dashboard",
"content_type": "RPM scale label",
"source_type": "real_world_object",
"movement_pattern": "static",
"persistence_pattern": "continuous",
"text": "x 1000 RPM",
"start_time": 1.0,
"end_time": 4.0,
"occlusion_level": "none",

{"id": "5",
"position": "center of dashboard",
"content_type": "gear indicator",
"source_type": "real_world_object",
"movement_pattern": "static",
"persistence_pattern": "sequential_change",
"variants": [
{"text": "D2","start_time": 1.0,"end_time": 2.0},
{"text": "D3","start_time": 2.0,"end_time": 3.0},
{"text": "D4","start_time": 3.0,"end_time": 4.0}],
"occlusion_level": "none"},

{"id": "6",
"position": "top right of dashboard",
"content_type": "clock",
"source_type": "real_world_object",
"movement_pattern": "static",
"persistence_pattern": "sequential_change",
"variants": [
{"text": "13:13","start_time": 1.0,"end_time": 3.0},
{"text": "13:14","start_time": 4.0,"end_time": 4.0}],
"occlusion_level": "none"}],

"caption": "The video begins with a view from inside a car, showing a person wearing a gray sweatshirt with a 'NY' logo on the chest, seated in the driver's seat with hands on a red leather steering wheel. The driver's face is blurred. Outside the windshield, a red car is visible on a wet road with a grassy field and an orange cone in the background. At 1.0s, the camera then shifts to a close-up of the car's digital dashboard, which displays a speedometer reading that sequentially increases from '22 mph' to '30 mph' to '38 mph' (1.0–4.0 seconds). The gear indicator in the center changes from 'D2' to 'D3' to 'D4' during the same period. The dashboard also shows a fixed 'x 1000 RPM' label at the bottom center with the tachometer numbers from 0 to 7, a clock reading '13:13' (changing to 13:14 at 4.0s), and a temperature reading of '+8.0° C' at the top right. The dashboard remains static in layout while the speed and gear indicators animate."}

Figure 3: Selected qualitative illustration of the evidence-to-caption interface. The structured record retains observed sequential changes in speed, gear, and time at the 2 FPS sampling resolution; the caption verbalizes these observations. This example is illustrative rather than a statistical error analysis.

Conclusion

Struct-VidOCR treats video OCR as an evidence-to-caption interface. Structured OCR Segments bind repeated observations into lifecycle records, matching-based GRPO provides order-invariant supervision, and caption consistency requires retained OCR facts to appear in a standalone caption. The 1.5k pseudo-labeled training examples make this interface explicit through record-then-caption supervision, while deterministic input gates and a 100-example manual spot check provide bounded quality controls for the data construction process. Under a fixed-reader, no-video readout, the full recipe improves the OCR evidence recoverable from captions.

The scope is deliberate: Struct-VidOCR does not claim coordinate-level tracking, improved raw character recognition, end-to-end QA, or replacement of specialist OCR cascades. Caption-Injected QA is consequently interpreted as conditional evidence recoverability from the generated text rather than reader-independent caption quality. It remains limited by tiny or blurred text, changes shorter than the 0.5-second sampling interval, ambiguous repeated strings, and long records in dense or lengthy videos. These limitations motivate complementary OCR front ends, matched frame-level controls, and more compact evidence representations.

References

Anonymous. 2025. DianJin-OCR-R1: Enhancing OCR Capabilities via a Reasoning-and-Tool Interleaved Vision-Language Model. *arXiv preprint arXiv:2508.13238*.

Bai, S.; Cai, Y.; Chen, R.; Chen, K.; Chen, X.; Cheng, Z.; Deng, L.; Ding, W.; Gao, C.; Ge, C.; et al. 2025. Qwen3-VL Technical Report. *arXiv preprint arXiv:2511.21631*.

Chen, J.; Xiao, S.; Zhang, P.; Luo, K.; Lian, D.; and Liu, Z. 2024. M3-embedding: Multi-linguality, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. In *Findings of the association for computational linguistics: ACL 2024*, 2318–2335.

Cheng, Z.; Lu, J.; Niu, Y.; Pu, S.; Wu, F.; and Zhou, S. 2019. You only recognize once: Towards fast video text spotting. In *Proceedings of the 27th ACM international conference on multimedia*, 855–863.

Comanici, G.; Bieber, E.; Schaekermann, M.; Pasupat, I.; Sachdeva, N.; Dhillon, I.; Blistein, M.; Ram, O.; Zhang, D.; Rosen, E.; Marris, L.; Petulla, S.; et al. 2025. Gemini 2.5: Pushing the Frontier with Advanced Reasoning, Multimodality, Long Context, and Next Generation Agentic Capabilities. *arXiv preprint arXiv:2507.06261*.

Cui, C.; Sun, T.; Liang, S.; Gao, T.; Zhang, Z.; Liu, J.; Wang, X.; Zhou, C.; Liu, H.; Lin, M.; Zhang, Y.; Zhang, Y.; Zheng, H.; Zhang, J.; Zhang, J.; Liu, Y.; Yu, D.; and Ma, Y. 2025. PaddleOCR-VL: Boosting Multilingual Document Parsing via a 0.9B Ultra-Compact Vision-Language Model. *arXiv preprint arXiv:2510.14528*.

Feng, H.; Wei, S.; Fei, X.; Shi, W.; Han, Y.; Liao, L.; Lu, J.; Wu, B.; Liu, Q.; Lin, C.; Tang, J.; Liu, H.; and Huang, C. 2025a. Dolphin: Document Image Parsing via Heterogeneous Anchor Prompting. *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (ACL)*.

Feng, K.; Gong, K.; Li, B.; Guo, Z.; Wang, Y.; Peng, T.; Wu, J.; Zhang, X.; Wang, B.; and Yue, X. 2025b. Video-R1: Reinforcing Video Reasoning in MLLMs. *arXiv preprint arXiv:2503.21776*.

Guo, D.; Yang, D.; Zhang, H.; Song, J.; Wang, P.; Zhu, Q.; Xu, R.; Zhang, R.; Ma, S.; Bi, X.; Zhang, X.; et al. 2025. DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning. *Nature*, 645(8081): 633–638.

He, H.; Zhang, J.; Ye, M.; Liu, J.; Du, B.; and Tao, D. 2025. GoMatching++: Parameter- and Data-Efficient Arbitrary-Shaped Video Text Spotting and Benchmarking. *arXiv preprint arXiv:2505.22228*.

Hu, A.; Xu, H.; Ye, J.; Yan, M.; Zhang, L.; Zhang, B.; Zhang, J.; Jin, Q.; Huang, F.; and Zhou, J. 2024. mPLUG-DocOwl 1.5: Unified Structure Learning for OCR-free Document Understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, 3096–3120.

Jonker, R.; and Volgenant, T. 1987. A shortest augmenting path algorithm for dense and sparse linear assignment problems. *Computing*, 38: 325–340.

Li, X.; Yan, Z.; Meng, D.; Dong, L.; Zeng, X.; He, Y.; Wang, Y.; Qiao, Y.; Wang, Y.; and Wang, L. 2025a. VideoChat-R1: Enhancing Spatio-Temporal Perception via Reinforcement Fine-Tuning. *arXiv preprint arXiv:2504.06958*.

Li, Y.; Yang, G.; Liu, H.; Wang, B.; and Zhang, C. 2025b. dots.ocr: Multilingual Document Layout Parsing in a Single Vision-Language Model. *arXiv preprint arXiv:2512.02498*.

Li, Z.; Liu, Y.; Liu, Q.; Ma, Z.; Zhang, Z.; Zhang, S.; Guo, Z.; Zhang, J.; Wang, X.; and Bai, X. 2025c. MonkeyOCR: Document Parsing with a Structure-Recognition-Relation Triplet Paradigm. *arXiv preprint arXiv:2506.05218*.

Li, Z.; Yang, B.; Liu, Q.; Ma, Z.; Zhang, S.; Yang, J.; Sun, Y.; Liu, Y.; and Bai, X. 2024. Monkey: Image resolution and text label are important things for large multi-modal models. In *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 26763–26773.

Liu, H.; Sun, D.; Wang, J.; Liu, Y.; and Pan, G. 2024. LOGO: Video Text Spotting with Language Collaboration and Glyph Perception Model. *arXiv preprint arXiv:2405.19194*.

Liu, Y.; Huang, M.; Yan, H.; Deng, L.; Wu, W.; Lu, H.; Shen, C.; Jin, L.; and Bai, X. 2025. VimTS: A unified video and image text spotter for enhancing the cross-domain generalization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(4): 2957–2972.

Meng, D.; Huang, R.; Dai, Z.; Li, X.; Xu, Y.; Zhang, J.; Huang, Z.; Zhang, M.; Zhang, L.; Liu, Y.; and Wang, L. 2025a. VideoCap-R1: Enhancing MLLMs for Video Captioning via Structured Thinking. *arXiv preprint arXiv:2506.01725*.

Meng, J.; Li, X.; Wang, H.; Tan, Y.; Zhang, T.; Kong, L.; Tong, Y.; Wang, A.; Teng, Z.; Wang, Y.; and Wang, Z. 2025b. Open-o3 Video: Grounded Video Reasoning with Explicit Spatio-Temporal Evidence. *arXiv preprint arXiv:2510.20579*.

Ouyang, K.; Liu, Y.; Wu, H.; Liu, Y.; Zhou, H.; Zhou, J.; Meng, F.; and Sun, X. 2025. SpaceR: Reinforcing MLLMs in Video Spatial Reasoning. *arXiv preprint arXiv:2504.01805*.

Shao, Z.; Wang, P.; Zhu, Q.; Xu, R.; Song, J.; Bi, X.; Zhang, H.; Zhang, M.; Li, Y.; Wu, Y.; and Guo, D. 2024. DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models. *arXiv preprint arXiv:2402.03300*.

Shi, Y.; Wang, H.; Xie, W.; Zhang, H.; Zhao, L.; Zhang, Y.-F.; Li, X.; Fu, C.; Wen, Z.; Liu, W.; Zhang, Z.; Chen, X.; Zeng, B.; Yang, S.; Zhang, Y.; Wan, P.; Wang, H.; and Yang, W. 2025. MME-VideoOCR: Evaluating OCR-Based Capabilities of Multimodal LLMs in Video Scenarios. *arXiv preprint arXiv:2505.21333*.

Tang, Y. Y.; Shimada, D.; Hua, H.; Huang, C.; Bi, J.; Feris, R.; and Xu, C. 2025. Video-R4: Reinforcing Text-Rich Video Reasoning with Visual Rumination. *arXiv preprint arXiv:2511.17490*.

Team, C.; Yue, Z.; Lin, Z.; Song, Y.; Wang, W.; Ren, S.; Gu, S.; Li, S.; Li, P.; et al. 2025a. MiMo-VL Technical Report. *arXiv preprint arXiv:2506.03569*.

Team, H. V.; Lyu, P.; Wan, X.; Li, G.; Peng, S.; Wang, W.; Wu, L.; Shen, H.; Zhou, Y.; Tang, C.; et al. 2025b. HunyuanOCR Technical Report. *arXiv preprint arXiv:2511.19575*.

Wang, W.; Gao, Z.; Gu, L.; Pu, H.; Cui, L.; Wei, X.; Liu, Z.; Jing, L.; Ye, S.; et al. 2025. InternVL3.5: Advancing Open-Source Multimodal Models in Versatility, Reasoning, and Efficiency. *arXiv preprint arXiv:2508.18265*.

Wei, H.; Kong, L.; Chen, J.; Zhao, L.; Ge, Z.; Yang, J.; Sun, J.; Han, C.; and Zhang, X. 2024a. Vary: Scaling up the vision vocabulary for large vision-language model. In *European Conference on Computer Vision*, 408–424. Springer.

Wei, H.; Liu, C.; Chen, J.; Wang, J.; Kong, L.; Xu, Y.; Ge, Z.; Zhao, L.; Sun, J.; Peng, Y.; et al. 2024b. General OCR Theory: Towards OCR-2.0 via a Unified End-to-end Model. *arXiv preprint arXiv:2409.01704*.

Wei, H.; Sun, Y.; and Li, Y. 2025. DeepSeek-OCR: Contexts Optical Compression. *arXiv preprint arXiv:2510.18234*.

Wu, W.; Cai, Y.; Shen, C.; Zhang, D.; Fu, Y.; Zhou, H.; and Luo, P. 2024. End-to-end video text spotting with transformer. *International Journal of Computer Vision*, 132(9): 4019–4035.

Yang, B.; Wen, B.; Ding, B.; Liu, C.; Chu, C.; Song, C.; Rao, C.; Yi, C.; Li, D.; et al. 2025a. Kwai Keye-VL 1.5 Technical Report. *arXiv preprint arXiv:2509.01563*.

Yang, Z.; Shu, Y.; Wang, J.; Yang, Z.; Zhang, Y.; Li, Y.; Lu, K.; Zeng, G.; Liu, S.; Zhou, Y.; and Sebe, N. 2025b. VidText: Towards Comprehensive Evaluation for Video Text Understanding. *arXiv preprint arXiv:2505.22810*.

Yu, H.; Zhao, M.; Lu, J.; Niu, K.; Wang, Y.; Yin, W.; Jia, W.; Fu, T.; Liu, Y.; Liu, J.; and Chen, H. 2025. EVE: Towards End-to-End Video Subtitle Extraction with Vision-Language Models. *arXiv preprint arXiv:2503.04058*.

Zhang, C.; Wang, Z.; Ma, Y.; Peng, J.; Wang, Y.; Zhou, Q.; Song, J.; and Zheng, B. 2025a. ReWatch-R1: Boosting Complex Video Reasoning in Large Vision-Language Models through Agentic Data Synthesis. *arXiv preprint arXiv:2509.23652*.

Zhang, Y.; Zeng, G.; Wu, D.; Shen, H.; Li, B.; Zhou, Y.; Ma, C.; and Bi, X. 2025b. Gather and trace: Rethinking video textvqa from an instance-oriented perspective. In *Proceedings of the 33rd ACM International Conference on Multimedia*, 876–885.

Zhang, Y.; Zhang, R.; Gu, J.; Zhou, Y.; Lipka, N.; Yang, D.; and Sun, T. 2023. LLaVAR: Enhanced Visual Instruction Tuning for Text-Rich Image Understanding. *arXiv preprint arXiv:2306.17107*.

Appendix

Detailed Reward Formulations and Thresholds

Temporal Grounding Reward (R_{iou}): The temporal alignment is computed using the Intersection over Union (IoU) of the time intervals:

$$R_{iou} = \text{IoU}([p.t_{start}, p.t_{end}], [g.t_{start}, g.t_{end}]) \tag{1}$$

For segments containing sequential variants, we compute IoU from the aggregate lifespan $[\mathbf{t}_{start}, \mathbf{t}_{end}]$ rather than individual variant timestamps. At 2 FPS, this avoids over-interpreting transition boundaries below the sampling resolution while still scoring the text instance's overall visible span.

Attribute Alignment Reward (R_{attr}): This score evaluates the categorical and semantic match:

$$R_{attr} = \sum_{k \in \mathcal{K}} w_k \cdot \mathbb{I}(p.k \approx g.k) \tag{2}$$

where $\mathcal{K}$ includes the attributes evaluated with their respective weights w_k: *position* (0.2), *content_type* (0.2), *source_type* (0.15), *movement_pattern* (0.15), *persistence_pattern* (0.15), and *occlusion_level* (0.15). We use exact matching for categorical flags. For open natural-language fields (e.g., *content_type* and *position*), we use multilingual M3E embeddings (Chen et al. 2024) only to tolerate paraphrases in the pseudo-label vocabulary; this auxiliary reward is not used for benchmark scoring and is not a claim of language-independent attribute evaluation:

$$S_{sem}(p, g) = \begin{cases} 1.0 & \text{if } \text{sim}_{\text{M3E}}(p, g) \geq \tau_{high} \\ 0.5 & \text{if } \tau_{low} \leq \text{sim}_{\text{M3E}}(p, g) < \tau_{high} \\ 0.0 & \text{otherwise} \end{cases} \tag{3}$$

We set $\tau_{high} = 0.9$ and $\tau_{low} = 0.8$, rewarding only high-similarity paraphrases.

Caption Consistency Reward (R_{cap}): Let N_{total} be the number of unique text segments in Y_{struct} and N_{match} be the number of these texts explicitly mentioned in Y_{cap}. The consistency score is defined as:

$$R_{cap} = \begin{cases} 1.0 & \text{if } N_{total} = 0 \text{ or } N_{match} = N_{total} \\ 0.5 & \text{if } \frac{N_{match}}{N_{total}} > \tau_{cov} \\ 0.0 & \text{otherwise} \end{cases} \tag{4}$$

We employ a threshold $\tau_{cov} = 0.7$, rewarding captions that expose enough recognized evidence for the intended text-only readout.

Caption Length Reward (R_{len}): We evaluate the length ratio $\rho = |Y_{cap}|/|G_{cap}|$ between generated and ground-truth captions. The reward follows a piecewise step function governed by predefined thresholds:

$$R_{len} = \begin{cases} 1.0 & \text{if } \tau_{len1} \leq \rho \leq \tau_{len2} \\ 0.5 & \text{if } \tau_{len3} \leq \rho \leq \tau_{len4} \\ 0.0 & \text{otherwise} \end{cases} \tag{5}$$

We set the inner bounds $[\tau_{len1}, \tau_{len2}] = [0.75, 1.25]$ and outer bounds $[\tau_{len3}, \tau_{len4}] = [0.5, 1.5]$ to discourage captions whose density is far from the pseudo-label reference.

Format Compliance Reward (R_{fmt}): It is the product of three indicator functions: $\mathbb{I}_{syn}$ (valid JSON syntax), $\mathbb{I}_{sch}$ (presence of mandatory keys and correct data types), and $\mathbb{I}_{log}$ (logical consistency, e.g., $t_{end} \geq t_{start}$):

$$R_{fmt} = \mathbb{I}_{syn} \cdot \mathbb{I}_{sch} \cdot \mathbb{I}_{log} \tag{6}$$

Consequently, $R_{fmt} \in \{0, 1\}$. It is an auxiliary validity reward: invalid outputs receive zero for this term, whereas valid outputs receive its weighted bonus. R_{fmt} is additive in R_{total}, not a hard decoding constraint; it encourages, but does not guarantee, valid JSON.

Reward Hyperparameters: In our experiments, the segment-level coefficients are $\lambda_{txt} = 1.0$, $\lambda_{iou} = 0.2$, and $\lambda_{attr} = 0.15$. The global weights are $\lambda_{cap} = 0.1$, $\lambda_{len} = 0.05$, and $\lambda_{fmt} = 0.2$. These values balance text, sampled-span, attribute, caption, length, and validity signals during exploration.

Data Construction Details

We construct an automatically quality-controlled, text-dense pool from Koala-36M and supplement the automatic checks with a manual spot audit. The procedure is intended to improve input legibility and pseudo-label reliability; it is not an exhaustive verification of every generated field or a benchmark-overlap audit.

Initial Heuristic Filtering. We first use preliminary Koala-36M captions and retain candidates containing more than five pairs of quotation marks, which serves only as a text-density heuristic. This retrieval stage does not use Gemini pseudo-label confidence.

OCR-Based Input Audit. We uniformly sample eight frames from each candidate and process them with PaddleOCRv5. A video is retained only when it satisfies all of the following visual-input criteria:

1. **Text Presence:** At least one text bounding box is detected ($num > 0$).
2. **High Confidence:** At least 95% of the detected text instances possess a high recognition score (≥ 0.95).
3. **Sufficient Resolution:** At least 95% of the boxes have height ≥ 32 pixels.
4. **Cropping Constraint:** No more than 20% of boxes touch a frame boundary.

Class Balancing and Final Sampling. To reduce dominance by subtitle/display-heavy clips, a candidate whose preliminary caption contains "subtitles" or "display" is discarded with probability 30%. We then randomly select 1,500 clips from the retained pool. This improves input diversity, but does not by itself establish source disjointness from downstream benchmarks.

Two-Stage Pseudo-Labeling. Gemini 2.5 Pro produces labels only after the input audit. In the first stage, it analyzes frames sampled at 2 FPS and emits the Structured OCR Segments, including temporal spans, source/motion attributes, occlusion, persistence, and time-indexed content variants. In the second stage, it receives the predicted record and generates a fluent caption intended to verbalize the retained evidence. The exact structure-generation, conditioned-caption, and training prompts are illustrated in figs. 4 to 6.

Manual Quality Audit. We randomly sampled 100 constructed examples for manual inspection and found that more than 90% were accurate, supporting the quality of the constructed set.

Training Details

We implement the reinforcement-learning pipeline with **ms-swift** and perform full-parameter fine-tuning of Qwen3-VL-8B-Instruct on one node with 8 NVIDIA H200 GPUs. DeepSpeed ZeRO-3 is used with optimizer and model offloading. Video inputs contain at most 64 sampled frames, and the maximum number of visual tokens is 512.

During GRPO exploration, vLLM runs in co-located mode (`vllm_mode=colocate`) with tensor parallel size 8 and GPU-memory utilization 0.3. For each input, we sample $G = 6$ completions at temperature 0.8 with a maximum completion length of 4,096 tokens.

Optimization uses `bfloat16` precision for two epochs, per-device batch size 3, and two gradient-accumulation steps. The peak learning rate is 10^{-6} with warmup ratio 0.01; gradients are clipped to maximum norm 0.5, and the GRPO KL coefficient is $\beta = 0.001$.

Evaluation Details

Inference Hyperparameters. All generators use temperature 0.2, `top_p`=0.95, and `max_tokens`=16,384. These controls match sampling conditions and visible-output budget across systems; they are not intended to reproduce every baseline's native end-to-end QA configuration.

Model Prompting Strategy. Struct-VidOCR uses the same composite-output instruction as training. Baseline VLMs receive the text-centric prompt in fig. 7, which requests chronological transcription, natural-language image-plane relations, and temporal order without requiring the proposed schema. For Keye and MiMo, `'./no_think'` is appended to suppress hidden reasoning output and keep the visible representation comparable.

Caption-Injected QA Protocol. Generation is question-blind: neither the trained-model prompt nor the baseline caption prompt contains the benchmark question. Gemini 2.5 Pro is then used as a frozen text-only reader, not a visual QA model in this protocol. It receives the benchmark question plus Y_{cap} and no video frames; in the separately reported * diagnostic it additionally receives predicted Y_{struct}. Benchmark reference answers determine correctness. The prompt is in fig. 8. Because the same model also supplies pseudo-labels, scores are interpreted as controlled within-protocol evidence-readout comparisons rather than reader-independent caption quality or end-to-end QA claims.

Benchmark-Specific Exclusions (VidText). We omit VidText Text Tracking because it requires numerical bounding boxes at first and last appearances, while our output supplies natural-language image-plane positions. The reported average is therefore over seven caption-compatible tasks only, not a full VidText aggregate.

Frame-Level OCR Caption Contrast

We include a Frame-Level OCR Caption Contrast to distinguish an external OCR front end from the proposed lifecycle representation. For each evaluation video, we sample frames at the same 2 FPS rate used by Struct-VidOCR and run a specialized OCR model on every sampled frame. Its recognized strings, frame timestamps, and image-plane detections are serialized in chronological order and inserted into a text-centric captioning prompt. The resulting caption is evaluated by the same question-blind, fixed text-only reader as the other systems.

This control supplies explicit per-frame transcription evidence, but it does not merge repeated detections into persistent instances or encode occlusion, motion, and content changes as lifecycle attributes. It is therefore complementary to Struct-VidOCR rather than a coordinate-level tracking comparison: the control tests whether a caption can be formed from independently

Prompt for Gemini 2.5 Pro to generate the explicitly Structured OCR Segments

System Prompt: You are a professional video-OCR understanding expert. Your goal is to produce an extremely accurate and comprehensive structured OCR description of video clips. Strictly follow the rules and field definitions below.
General rules:
1. Accurately recognize all visible text appearing in the video, including textual content, time range, position, semantic type, source, and dynamic characteristics.
2. The output must be only a single JSON text (plain text) containing "ocr_segments" (an array). Do not output any additional explanations.
3. Frame-rate rule: frames are sampled at 2 FPS (each frame = 0.5 second), with frame indices starting at 0. The time point of frame n = n / 2 seconds.
4. Merging rule: if text at the same position and from the same source undergoes sequential changes across consecutive frames, group them into the same ocr_segment (persistence_pattern="sequential_change"); otherwise split into separate segments. If the same text region disappears and reappears within <= 1.0 second (merge_gap_threshold = 1.0s) and its style/source/position are consistent, merge into a single ocr_segment and record the interruption as an occlusion_event; otherwise split into separate segments.
5. Reading order rule: first sort ocr_segments by chronological order of appearance (start_time ascending), when multiple OCR items appear at the same time, order them by on-screen reading order: left to right, then top to bottom.
Fields required for each ocr_segment:
- id (string): unique identifier, e.g., "1".
- position (string): natural-language screen-position description (e.g., "top-left", "bottom-center", "middle of the right wall"). Do not write coordinates.
- content_type (string): semantic type of the text (e.g., "subtitle", "shop sign", "poster text", "scoreboard", "advertising slogan").
- source_type (string):
* "real_world_object": text printed/painted on physical objects.
* "burned_in_graphic": post-production overlaid text.
- movement_pattern (string):
* "static"
* "moving_with_object"
* "fixed_on_screen"
* "animated_overlay"
- persistence_pattern (string):
* "continuous"
* "sequential_change"
For persistence_pattern != "sequential_change":
- text (string): verbatim text.
- start_time (number)
- end_time (number)
For persistence_pattern == "sequential_change":
- start_time (number)
- end_time (number)
- variants (array), each containing:
* text (string)
* start_time (number)
* end_time (number)
Occlusion description:
- occlusion_level (string): degree of visual occlusion affecting readability:
* "none": text is fully visible and unobstructed throughout its visible duration.
* "partial": text is partially blocked, blurred, cropped, or intermittently obscured but still readable.
* "severe": text is largely unreadable due to heavy obstruction, extreme blur, truncation, or being mostly out of frame.
Note:
All merge/split decisions must be based strictly on observable visual evidence (text appearance, position, style, motion). Do not infer unseen content.

User Prompt: I will provide a sequence of images sampled at 2 FPS (each second = 2 frames) with frame indices starting at 0. Treat these frames as consecutive frames from the same video clip and complete the task below.
Output must be only a single JSON text containing "ocr_segments". Do not include explanations or notes.
Task requirements:
1) Extract all visible text and create one ocr_segment for each distinct text region. Fields must strictly follow the system_prompt definitions.
2) For text that changes sequentially at the same position and from the same source, merge them into one ocr_segment with persistence_pattern="sequential_change" and list variants chronologically.
3) For non-sequential text, represent with a single text, start_time, and end_time.
4) Time calculation rule: the time of frame n = n / 2 seconds. If the same text appears in consecutive frames i..j, then start_time = i/2 and end_time = j/2; if it appears in a single frame only, start_time = end_time = n/2.
5) Apply merge_gap_threshold = 1.0 second:
If identical text disappears and reappears within this duration and style/source/position remain consistent, merge into the same ocr_segment; otherwise create a new segment.
6) Ordering rule:
First sort ocr_segments by start_time in chronological order.
If multiple text regions appear at the same time, order them by on-screen reading order: left to right, then top to bottom.
7) position must use natural English phrases. Do not output coordinates or bounding boxes.
8) occlusion_level must be "none", "partial", or "severe" based on actual visual obstruction severity.

Figure 4: Prompt 1 for Gemini 2.5 Pro to generate the explicitly Structured OCR Segments from the sampled video frames.

Prompt for Gemini 2.5 Pro to generate temporally grounded and fluent video captions

System Prompt: You are a professional video-OCR understanding and visual-description expert. Your goal is to, given an input sequence of frames (treated as a complete video, fps=2) and ocr_segments, generate a high-quality, single-paragraph English chronological description (caption). Strictly follow the requirements below:
[Temporal Order and Video Content Requirements]
- Treat the input frame sequence as a video and describe the visual content strictly following the progression of time.
- The description should cover but is not limited to:
* People (appearance features, posture, actions, order of actions, interactions)
* Objects (attributes, color, shape, material, quantity, relative position, motion or changes)
* Scene environment (foreground and background elements, structure, layout)
* Events and how they evolve over time
- All descriptions must be based on content visible in the images; do not infer information outside the frame.
[Language Constraints]
- Do not use subjective aesthetic words (e.g., "beautiful", "attractive", "stunning").
- Do not use atmospheric or mood descriptors (e.g., "tense atmosphere", "mysterious feeling").
- Do not use image-quality descriptors (e.g., "blurry", "high-definition", "clearly visible").
- Use objective, neutral, concrete, and accurate language throughout.
[Time and Timestamp Requirements — MANDATORY]
- fps = 2. The time of frame n = n / 2 seconds. All time values MUST be written as seconds and the suffix "s" (e.g., "0.0s", "1.5s").
- **You MUST include explicit timestamps (X.Xs) when describing every ocr_segment's appearance and disappearance, and when describing any occlusion event or change.** Use the start_time and end_time values from the provided ocr_segments.
- When describing variants in a sequential_change segment, include each variant's timestamp range in X.Xs format.
[OCR Information Integration Requirements (must be strictly followed)]
The caption must naturally, completely, and fluently integrate all key information from every ocr_segment, and **all text fields in all ocr_segments (including every text in variants) must be mentioned in the caption without omission (except for scrolling text).** Specific requirements are:
1. **Unique identifier (id)**
- The id does not need to be said in the caption, but ensure the corresponding text information is fully integrated.
2. **position (natural spatial description in the frame)**
- Use natural English spatial expressions, such as "top-left", "center-right of the frame", "on the upper area of the right wall"; do not write coordinates.
- If text moves with an object or the camera, reflect the relative change of position.
3. **content_type (semantic type of the text)**
- Naturally state the type of the text, for example "text in the subtitles...", "the sign reads...", "an advertising slogan displays...", "a news ticker appears at the bottom of the screen...".
4. **source_type (text source)**
- If source_type is real_world_object:
* Integrate naturally (e.g., "the sign reads...", "a label on the wall reads...").
- If source_type is burned_in_graphic:
* Describe as overlaid/burned-in text (e.g., "an overlay appears at the lower-left...").
5. **movement_pattern (motion characteristic)**
- static: indicates the text position remains unchanged and need not be specially described.
- moving_with_object: indicate that it moves with an object or camera, e.g., "as the person moves, the line 'XX' also moves in the frame."
- fixed_on_screen: indicate it is a fixed screen overlay.
- animated_overlay: describe the animation entry method in natural language, e.g., "then a line 'XX' slides in from the left."
6. **persistence_pattern (persistence mode)**
- continuous: no special expression needed other than saying it remains visible with exact timestamps.
- sequential_change (including variants):
* You must describe every change in the variants in their chronological order.
* Use explicit timestamps (X.Xs) for each variant (e.g., "initially at 0.0s shows 'A' until 0.5s; then at 0.5s changes to 'B' until 1.0s").
* You must explicitly write each variant's text; summarizing or skipping any variant is not allowed.
7. **text (text content)**
- The caption must explicitly state the exact text content; vague expressions are not allowed.
- Every single text in variants must appear at least once in the caption and must not be omitted.
- For scrolling text, you do NOT need to describe each temporal segment or variant separately. Instead, write the full combined textual content that appears during the scrolling once in the caption.
8. **start_time / end_time (time fields)**
- You must use explicit timestamps (X.Xs) derived from the ocr_segments for all time references in the caption.
- Use the duration (end_time - start_time) to choose phrasing such as "appears briefly", "appears for a short while", "remains for a period", or "remains visible throughout", but timestamps are still required.
9. **variants (only for sequential_change scenarios)**
- All variant texts must be presented clearly in order with their timestamps.
- If position is fixed, you may write "text at the same position changes from 'XX' to 'YY' (0.0s→0.5s)...".
- If variants involve occlusion or movement, reflect those changes and the corresponding circumstances and times.
10. **occlusion_level and occlusion relations **
- The caption must describe occlusion relationships when observable: explicitly state *what* occludes *which* text (for example, "a hand", "a person", "a passing object", or "the left frame edge") and the exact time range of the occlusion in X.Xs.
- Also include the occlusion degree using only the allowed terms "none", "partial", or "severe", integrated naturally (e.g., "the sign 'SALE' is partially occluded by a hand between 1.0s and 1.5s").
- Do not infer occlusion causes beyond what is visible; describe only observed occluding objects/elements and times.
[Style Requirements]
- The entire caption must be coherent and natural; do not produce field-like or list-like wording.
- OCR information must be smoothly integrated into the description; avoid bluntly listing OCR entries one by one.
- The caption must be a single paragraph, without line breaks or list structures.

User Prompt: Please, based on the provided frame sequence (fps=2) and the corresponding ocr_segments, generate a single-paragraph English caption in accordance with the inputs. **Output only a single JSON text; do not add any other explanations or comments.**
Please ensure:
1. The description is complete, objective, and strictly follows temporal order, covering but not limited to people's appearance features, order of actions, interactions, object attributes and placements, event development, and scene foreground/background.
2. Language must be objective and neutral; do not include mood, aesthetic, or image-quality terms.
3. All information from every ocr_segment (text content, position, source, movement feature, persistence mode, occlusion information) must be naturally integrated into the description, not listed bluntly. **Emphasize: every text field in ocr_segments must be mentioned; if an ocr_segment contains variants, present the variants in the exact order given and explicitly list each displayed text with its timestamps.**
4. The caption must be a single paragraph, with no line breaks, no bulleting, and no extra commentary.
5. You MUST mention specific timestamps for every ocr_segment and for every occlusion event. Always format times as seconds and the suffix 's' (e.g., '0.0s', '1.5s'). Use the start_time and end_time values from the ocr_segments for timestamps.
6. Describe occlusion relations clearly: state what occludes the text (object, person, or frame edge), the occlusion degree ("none", "partial", or "severe") integrated naturally in the sentence, and the exact time range of the occlusion (X.Xs). Do not infer unseen occlusions.
7. All ocr_segments must be described in ascending order of start_time. If multiple text regions start at the same time, follow on-screen reading order (left→right, top→bottom).
8. Do not invent text or objects not present in the frames; base all descriptions strictly on visible evidence.
The input ocr_segments are: ***OCR_SEGMENTS***
Strictly output in the following JSON format:
{ "caption": "A single-paragraph caption that meets all the above requirements." }

Figure 5: Prompt 2 for Gemini 2.5 Pro to generate temporally grounded and fluent video captions conditioned on the Structured OCR Segments.

Prompt for Training

User Prompt: Generate a single JSON text (no extra commentary) with:

1) ocr_segments: Detect all visible text and specify its position (using natural English directional terms), content_type, source_type, movement_pattern, persistence_pattern, the corresponding text content, and its time range. If the persistence_pattern is sequential_change, include all variants.
2) "caption": A English paragraph, chronological, factual (no mood/quality adjectives), integrating every OCR segment and variant with timestamps for appearances/changes/occlusions; briefly note people (clothing/posture/actions), objects (color/shape/position/motion), scene layout (foreground/background, left/right). No inference beyond visible frames.

Required rules:
- id starts from "1" and increases sequentially.
- Order ocr_segments by start_time (ties: left→right, top→bottom).
- Merge: identical text that vanishes and returns within ≤1.0s with same style/position → merge and record the interruption as an occlusion event.
- Occlusion(in caption): name occluder (object/person/frame edge), time range (X.Xs), degree (none/partial/severe); report only observed occlusions.
- Caption must cover all ocr_segments and every variant, use timestamps (X.Xs–X.Xs) to state exactly what text is visible during each time span (including changes and occlusions). Scrolling text exception: include combined full text once with overall start/end and any occlusions (no per-frame variants).

Use this JSON structure exactly:

```
{
"ocr_segments": [
{
"id": string,
"position": string,
"content_type": string,
"source_type": "real_world_object" | "burned_in_graphic",
"movement_pattern": "static" | "moving_with_object" | "fixed_on_screen" | "animated_overlay",
"persistence_pattern": "continuous" | "sequential_change",
// for continuous:
"text": string,
"start_time": number,
"end_time": number,
// for sequential_change:
"variants": [{"text": string, "start_time": number, "end_time": number}],
"occlusion_level": "none" | "partial" | "severe"
}
],
"caption": string
}
```

Figure 6: The exact training prompt utilized to instruct the VLM during the GRPO alignment phase to produce both the Structured OCR Segments and the descriptive caption.

observed OCR outputs, whereas our representation explicitly learns how observations should be associated and verbalized across time. We use the same frame rate, decoding settings, reader, and benchmark questions to keep the comparison focused on the evidence interface.

Prompt for Baseline models captioning

User Prompt: Generate a single English paragraph caption describing this video chronologically.

Please focus heavily on recognizing and transcribing all visible text (OCR). For any text that appears in the video, you must include the exact text content, its position within the frame, and its corresponding timestamps (X.Xs–X.Xs) indicating when it is visible.

Alongside the text, briefly describe the visual content factually: note people (clothing/posture/actions), objects, and the general scene layout. Do not use mood or quality adjectives, and do not make any inferences beyond what is explicitly visible in the frames. Output only the plain text paragraph.

Figure 7: The text-centric captioning prompt utilized for the baseline models. It explicitly instructs the models to generate a chronological narrative prioritizing precise OCR transcription and spatiotemporal grounding.

Prompt template for Caption-Injected QA evaluation

[System Context Injection]
(Optional for Joint Injection setting)
The OCR information in video: {ocr_segments_json}

Video Caption: {generated_caption_text}

[General Instruction]
Based on the [OCR information and] video caption, directly answer the question below.

[Question & Task Constraints]
Question: {Original question and specific format constraints from the benchmark}

(Optional) Options:
A. {option_A}
B. {option_B}
C. {option_C}
D. {option_D}

[Output]
The answer is:

Figure 8: The prompt template utilized for Gemini 2.5 Pro during the "Caption-Injected QA" evaluation. The LLM relies exclusively on the injected text representations to answer the benchmark questions.

Generalizability across Architectures

InternVL3.5-8B and MiMo-VL-7B-RL-2508 use the primary full-parameter recipe; Qwen3-VL-32B-Instruct uses LoRA ($r = 8$, $\alpha = 32$) due to compute constraints. Detailed per-backbone results are reported in tables 5 and 6.

Caption Quality Analysis

We assess whether the OCR-focused alignment changes caption behavior on text-visible subsets of VidCapBench-AE and DREAM-1K. The generation prompt is in the appendix; these are complementary caption-quality measures rather than evidence-readout QA scores.

Because the method targets text-rich videos, we retain clips that Gemini 2.5 Pro classifies as `has_visable_text: true` with `confidence_score: 1`; the filtering prompt is in fig. 9. This yields 350 DREAM-1K and 283 VidCapBench-AE videos. The automatic filter selects a text-visible subset.

Detailed caption-quality metrics are reported in tables 9 and 10; the main paper summarizes their precision–recall implications.

Model	TR	VTQA	TG	AR	CDT	STP	CFTU	TBR	TBVU	RVT	Total
InternVL3.5-8B	41.7	58.0	44.0	48.7	42.5	56.0	34.0	76.2	58.1	69.0	51.5
+ Ours	42.1	57.6	44.1	49.2	42.1	61.4	34.3	77.3	58.3	70.2	52.7
+ Ours*	43.5	58.7	46.2	53.3	42.9	61.4	32.5	78.0	58.7	71.0	53.9
MiMo-VL-7B-RL-2508	52.7	71.1	64.0	62.4	55.5	67.7	35.4	80.9	55.0	65.7	61.1
+ Ours	54.3	68.3	64.3	61.7	56.9	68.7	36.6	79.6	59.6	67.6	61.3
+ Ours*	57.3	69.0	66.5	62.6	57.8	71.5	36.4	77.3	60.6	67.9	62.2
Qwen3-VL-32B-Instruct	63.7	72.4	65.5	62.0	55.5	68.6	35.1	77.0	59.8	66.2	62.9
+ Ours (LoRA)	63.8	74.2	66.8	62.8	55.6	69.9	36.2	77.4	61.4	68.4	64.1
+ Ours (LoRA)*	66.0	76.8	68.5	60.1	55.8	76.9	37.4	76.8	64.5	70.2	65.8

Table 5: Results on additional tested backbones for the fixed-reader MME-VideoOCR protocol. "+ Ours" applies the representation-alignment recipe. * denotes the separately reported structured-readout diagnostic, in which predicted records are provided alongside captions.

Model	HoliOCR	HoliRea.	LocalOCR	LocalRea.	TextLocal.	TempCauRea.	SpaRea.	Avg.
InternVL3.5-8B	22.6	43.2	9.4	51.1	8.9	25.4	28.5	27.0
+ Ours	22.2	43.3	10.4	51.5	16.0	26.7	35.4	29.4
+ Ours*	30.2	43.3	18.6	52.7	26.4	27.5	36.3	33.6
MiMo-VL-7B-RL-2508	23.1	53.7	16.9	48.8	21.1	28.3	34.2	32.3
+ Ours	24.0	52.2	22.0	47.7	28.7	29.2	35.7	34.2
+ Ours*	28.2	51.7	24.1	50.2	49.7	32.5	37.1	39.1
Qwen3-VL-32B-Instruct	25.2	54.6	17.1	50.3	21.3	31.1	42.6	34.6
+ Ours (LoRA)	26.3	53.5	24.1	52.2	48.3	44.3	37.5	40.9
+ Ours (LoRA)*	29.8	54.2	24.3	53.9	54.0	35.8	45.7	42.5

Table 6: Results on additional tested backbones for the seven-task caption-compatible VidText protocol. "+ Ours" applies the representation-alignment recipe. * denotes the separately reported structured-readout diagnostic.

Additional Qualitative Results

We provide selected qualitative case studies from MME-VideoOCR. They illustrate how the representation can retain lifecycle evidence at the 2 FPS sampling resolution; they are not a statistical error analysis.

- **Reduced temporal/content confusion in these examples:** The record preserves text observations and their sampled spans rather than relying on a single free-form summary.
- **Source and state categories:** The examples distinguish subtitles, burned-in graphics, and scene text.
- **Sequential changes:** Time-indexed variants can group incremental typing or score changes without fragmenting the caption into disconnected frame descriptions.

R_{txt}	R_{attr}	R_{iou}	R_{cap}	R_{len}	TR	VTQA	TG	AR	CDT	STP	CFTU	TBR	TBVU	RVT	Total
✓					51.7	60.9	44.9	48.9	47.7	57.4	35.8	76.0	60.7	64.3	54.8
					60.6	74.4	64.9	**54.4**	47.4	70.8	34.8	**79.8**	62.5	66.1	62.5
✓	✓				53.3	67.1	47.7	52.4	43.4	55.5	31.1	73.1	60.2	**71.7**	55.5
					64.3	72.9	64.5	53.1	42.3	**73.8**	35.5	**79.8**	**66.3**	**73.3**	63.2
✓	✓	✓			35.4	52.8	37.9	43.6	35.7	55.0	29.8	65.8	57.4	63.3	46.9
					63.9	72.3	**70.2**	52.3	42.8	71.0	35.5	77.8	63.6	67.2	62.7
✓	✓	✓	✓		**63.9**	**72.5**	**59.5**	54.7	46.9	62.7	**38.6**	72.5	60.1	67.5	60.7
					66.8	**75.5**	65.1	53.1	42.8	70.1	37.9	78.0	61.7	69.9	63.2
✓	✓	✓	✓	✓	63.0	72.0	56.0	**57.4**	**52.0**	**68.6**	37.2	**78.0**	**61.5**	69.4	**62.2**
					66.4	74.0	68.3	53.7	**51.8**	67.2	**38.0**	75.0	62.8	70.9	**63.6**

Table 7: Full per-task reward-component ablation. R_{fmt} is applied throughout. For each configuration, the top row is caption-only; the bottom * row additionally supplies the predicted record.

Variant	**TR**	**VTQA**	**TG**	**AR**	**CDT**	**STP**	**CFTU**	**TBR**	**TBVU**	**RVT**	**Total**
Thresholded(0.5)	**63.9**	**73.7**	60.5	51.4	50.5	66.4	35.9	71.8	58.0	65.7	61.2
Thresholded(0.5)*	63.7	74.2	66.3	**55.9**	**52.0**	68.2	37.5	76.6	60.1	65.9	62.9
Thresholded(0.25)	62.8	73.3	55.2	**58.4**	50.0	68.4	33.3	77.1	62.3	69.2	61.7
Thresholded(0.25)*	65.7	**76.5**	65.1	52.0	47.5	70.5	31.6	**77.3**	61.6	**71.8**	63.2
w/o Len-W	63.5	73.1	**62.9**	54.1	47.4	65.4	35.1	71.8	**62.8**	**70.7**	61.6
w/o Len-W*	65.2	73.3	66.7	50.0	48.5	**71.0**	35.1	75.3	61.9	71.6	63.1
Ours (Continuous & Len-W)	63.0	72.0	56.0	57.4	**52.0**	**68.6**	**37.2**	**78.0**	61.5	69.4	**62.2**
*Ours (Continuous & Len-W)**	66.4	74.0	68.3	53.7	51.8	67.2	38.0	75.0	62.8	70.9	63.6

Table 8: Full per-task ablation on matching and length-weighting choices. * rows additionally supply predicted records alongside captions.

Model	**VidCapBench-AE**				
	Overall Acc / Pre Cov / Con	**Video Aesthetics** Acc / Pre Cov / Con	**Video Content** Acc / Pre Cov / Con	**Video Motion** Acc / Pre Cov / Con	**Physical Laws** Acc / Pre Cov / Con
Qwen3-VL-8B-Instruct	20.8 / 64.4 85.2 / 5.5	17.1 / 52.3 76.1 / 4.5	21.7 / 69.1 88.9 / 5.8	12.6 / 49.0 90.1 / 3.3	35.1 / 64.9 80.0 / 9.3
Struct-VidOCR-8B (Ours)	23.3 / 69.1 86.3 / 7.4	19.6 / 60.3 75.9 / 6.2	24.5 / 73.2 90.6 / 7.8	13.5 / 45.6 91.0 / 4.3	33.0 / 63.5 80.0 / 10.5

Table 9: Caption-quality measures on the text-visible VidCapBench-AE subset. We report Accuracy (Acc), Precision (Pre), Coverage (Cov), and Consistency (Con).

Prompt for Gemini 2.5 Pro to filter benchmarks

Role
You are an expert in multimodal visual analysis and Optical Character Recognition (OCR), specializing in detecting legible text within dynamic video sequences.

Task
Analyze the provided sequence of image frames to determine if any clearly visible and identifiable text content is present.

Guidelines
1. **Text Definition**: Include headlines, body text, subtitles, street signs, UI elements (labels/buttons), watermarks, and clear handwriting.
2. **Exclusions**: Ignore natural textures that mimic text (e.g., crossing branches, stone cracks), severe motion blur where characters are indistinguishable, and pixelated noise.
3. **Logic**: If at least one frame in the sequence contains identifiable characters, set `has_visible_text` to `true`.

Output Format
Return ONLY a valid JSON object. Do not include any preamble, conversational filler, or markdown formatting outside the JSON block.

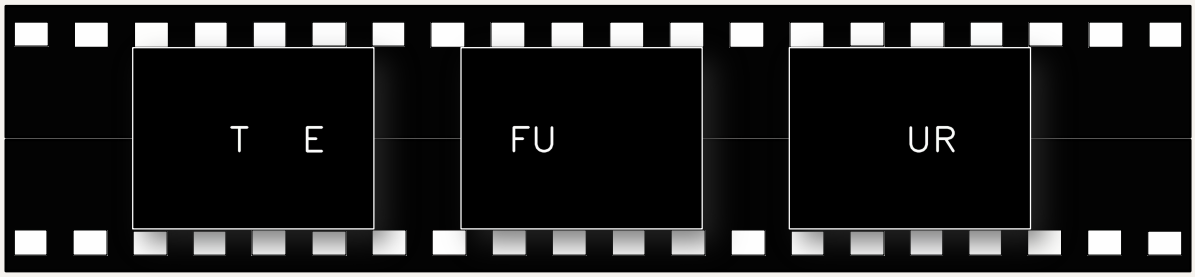
JSON Structure:
{
"has_visible_text": boolean,
"confidence_score": float, // Range 0.0 to 1.0
"detected_frames": [number], // Indices of frames containing text (starting from 0)
"text_content_sample": "string", // A brief snippet of the detected text
"reasoning": "string" // A concise explanation of your judgment
}

Figure 9: Prompt used to select text-visible subsets of VidCapBench-AE and DREAM-1K: clips are retained when Gemini returns `has_visable_text: true` and `confidence_score: 1`.

Model	Live-action	Animation	YouTube	Shorts	Stock	Overall
Qwen3-VL-8B-Instruct	42.4/44.1/40.9	31.5/29.6/33.7	30.4/33.4/27.9	39.0/48.0/32.9	47.4/48.0/46.7	38.4/40.6/36.4
Struct-VidOCR-8B (Ours)	43.3/43.5/43.1	31.8/32.4/31.3	32.0/38.3/27.4	36.7/45.6/30.7	44.2/46.2/42.4	37.8/41.2/35.0

Table 10: Caption-quality measures on the text-visible DREAM-1K subset. Metrics are F1/Precision/Recall; results show a precision–recall trade-off.

Qualitative Case Study



Qwen3-VL-8B-Instruct: The video begins with a black screen at 0.0 seconds. At 0.1 seconds, the letter \"T\" appears in the center of the screen. At 1.2 seconds, the letter \"E\" appears to the right of the \"T\", forming the word \"TE\". At 2.2 seconds, the letter \"U\" appears below the \"E\", forming the word \"TEU\". At 3.2 seconds, the letter \"R\" appears to the right of the \"U\", forming the word \"TEUR\". The video ends with the word \"TEUR\" displayed on the screen.

Qwen3-VL-235B-A22B-Instruct: At 0.0s, the frame is entirely black with no visible text or objects. At 0.2s, the letters \"FU\" appear in light gray, sans-serif font, centered horizontally and vertically on the black background. At 1.2s, the letters \"UR\" appear in the same light gray, sans-serif font, also centered horizontally and vertically on the black background. No people, objects, or scene elements other than the text are visible in any frame.

Struct-VidOCR

{
"ocr_segments": [
{
"id": "1",
"position": "center-left",
"content_type": "title",
"source_type": "burned_in_graphic",
"movement_pattern": "fixed_on_screen",
"persistence_pattern": "continuous",
"text": "T",
"start_time": 0.0,
"end_time": 0.5,
"occlusion_level": "none"
},
{
"id": "2",
"position": "center-right",
"content_type": "title",
"source_type": "burned_in_graphic",
"movement_pattern": "fixed_on_screen",
"persistence_pattern": "continuous",
"text": "E",
"start_time": 0.0,
"end_time": 0.5,
"occlusion_level": "none"
},
{
"id": "3",
"position": "center-left",
"content_type": "title",
"source_type": "burned_in_graphic",
"movement_pattern": "fixed_on_screen",
"persistence_pattern": "continuous",
"text": "FU",
"start_time": 1.0,
"end_time": 1.5,
"occlusion_level": "none"
},
{
"id": "4",
"position": "center-right",
"content_type": "title",
"source_type": "burned_in_graphic",
"movement_pattern": "fixed_on_screen",
"persistence_pattern": "continuous",
"text": "UR",
"start_time": 2.0,
"end_time": 2.5,
"occlusion_level": "none"
}],

"caption": "The video begins with a black screen displaying the letter 'T' on the center-left and the letter 'E' on the center-right, both appearing at 0.0 seconds and remaining static until 0.5 seconds. At 1.0 seconds, the screen transitions to display the letters 'FU' positioned slightly left of center, which persists until 1.5 seconds. At 2.0 seconds, the screen displays the letters 'UR' , positioned slightly right of center, remaining static until 2.5 seconds. All text is burned-in, fixed on screen, and unoccluded throughout."}

Figure 10: Selected transient-text example. The video displays independent flashes "T", "E", "FU", and "UR". The baseline combines them into "TEUR", while Struct-VidOCR represents them as separate sampled-span observations (e.g., "T" at 0.0–0.5 s and "FU" at 1.0–1.5 s).

Qualitative Case Study



Qwen3-VL-8B-Instruct: In the video, three students in white school uniforms with dark pants enter a warmly lit dining room through a wooden door. A man in a black shirt over a white t-shirt stands near the doorway, greeting them with a smile. The students, carrying backpacks, walk into the room, and the man gestures welcomingly, saying, "Hello. Our senior high students. Welcome high students. Come in. Go wash your hands." He then moves to a dining table set with various dishes and an orange juice bottle, commenting, "That's a lot of side dishes. Good because I'm starving." As the students approach the table, the girl in the foreground turns to face the man and says, "Wait a second, I have some big news for you." The boy behind her responds, "Oh, really?" while the man replies, "Let's eat." The room features a calendar on the wall with Chinese characters, a plant on a shelf, and a curtain with a leaf pattern. The students appear eager to eat, and the man seems pleased with the abundance of food.

Qwen3-VL-235B-A22B-Instruct: At 0.0s–0.2s, a girl in a white school uniform with a dark collar and trim enters a room through a doorway, followed by two boys in similar uniforms; one boy carries a backpack. A man in a black button-up shirt over a white t-shirt stands near a dining table, turning toward them. Subtitles read: "- Hello. - Dad, we're home." The room has wooden furniture, a shelf with decorative items, a window with blinds, and a wall calendar with red Chinese characters. At 0.3s–0.5s, the man gestures with his right hand toward the children, smiling, as the subtitle reads: "Our senior high students." At 0.6s–0.7s, the man continues speaking with a welcoming gesture, and the subtitle reads: "Welcome home." At 0.8s–0.9s, the man gestures toward the table and says, "Come in. Go wash your hands," as the children move further into the room. At 1.0s–1.1s, the man bends slightly toward the table, looking at the food, and the subtitle reads: "That's a lot of side dishes." At 1.2s–1.3s, the man sits at the table, smiling, and says, "Good because I'm starving," as the girl turns toward him. At 1.4s–1.5s, the girl stands facing the man, raising her right hand slightly, and says, "Wait a second, I have some big news for you," while the boys stand behind her. At 1.6s–1.7s, the girl turns away from the man and walks toward the back of the room, while the man looks at her and replies, "Oh, really?" The table is set with multiple dishes, a bottle of orange juice, and glasses. The logo "华策影视 HUACE FILM&TV" appears in the top right corner throughout the clip.

Struct-VidOCR

{ "ocr_segments": [

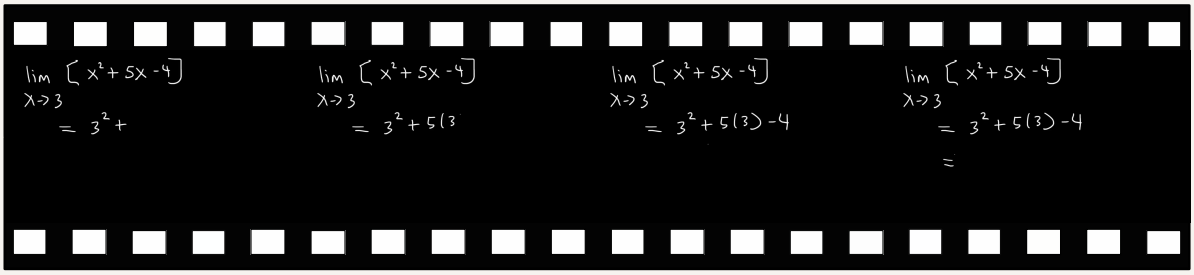
{"id": "1",
"position": "top_right",
"content_type": "logo",
"source_type": "burned_in_graphic",
"movement_pattern": "static",
"persistence_pattern": "continuous",
"text": "华策影视 HUACE FILM&TV",
"start_time": 0.0,
"end_time": 12.0,
"occlusion_level": "none"},
{"id": "2",
"position": "right_center",
"content_type": "text_on_wall",
"source_type": "real_world_object",
"movement_pattern": "static",
"persistence_pattern": "continuous",
"text": "福",
"start_time": 0.0,
"end_time": 12.0,
"occlusion_level": "none"}

{"id": "3",
"position": "bottom-center",
"content_type": "subtitle",
"source_type": "burned_in_graphic",
"movement_pattern": "fixed_on_screen",
"persistence_pattern": " sequential_change",
"variants": [
{"text": "Hello. Dad, we're home.","start_time": 0.0,"end_time": 1.5},
{"text": "Our senior high students.","start_time": 1.5,"end_time": 2.5},
{"text": "Welcome high students.",start_time": 2.5,"end_time": 3.5},
{"text": "Come in. Go wash your hands.","start_time": 3.5,"end_time": 5.5},
{"text": "That's a lot of side dishes.","start_time": 5.5,"end_time": 7.0},
{"text": "Good because I'm starving.","start_time": 7.0,"end_time": 9.5},
{"text": "Wait a second, I have some big news for you.","start_time": 9.5,"end_time": 11.5},
{"text": "Oh, really? - Let's eat.","start_time": 11.5,"end_time": 12.0},
"occlusion_level": "none"
}],

"caption": "0.0s–1.5s: In a warmly lit dining room, three students in white polo shirts with dark pants enter through a wooden door. A man in a black button-up shirt over a white tee stands near a wooden dining table, smiling and gesturing. Subtitles read: 'Hello. Dad, we're home.' A calendar with the Chinese character '福' (fortune) is visible on the wall. A bottle of orange juice sits on the table. The room features wooden furniture, a shelf with decorative items, and a window with blinds. 1.5s–3.5s: The man, now facing the students, gestures welcomingly. Subtitles read: 'Our senior high students.' and 'Welcome high students.' A patterned curtain with large green leaves hangs near a doorway. 3.5s–5.5s: The man gestures toward the kitchen area. Subtitles read: 'Come in. Go wash your hands.' The students, one carrying a backpack, move toward the kitchen. 5.5s–7.0s: The man, now seated, gestures toward the table laden with food. Subtitles read: 'That's a lot of side dishes.' 7.0s–9.5s: The man, seated, leans forward with a hungry expression. Subtitles read: 'Good because I'm starving.' 9.5s–11.5s: The teenagers move around the room, with the girl turning to face the man. Subtitles read: 'Wait a second, I have some big news for you.' 11.5s–12.0s: The girl turns to the man, and the boy behind her looks on. Subtitles read: 'Oh, really? - Let's eat.' The table is set with various dishes, and the man remains seated, smiling."}

Figure 11: Selected text-rich example. Struct-VidOCR separates a burned-in TV logo, scene text, and dialogue subtitles, using `sequential_change` to record the subtitle updates at the sampled temporal resolution.

Qualitative Case Study



Qwen3-VL-8B-Instruct: The video displays a mathematical limit problem written in white text on a black background. The expression is \"lim [x² + 5x - 4]\" with \"x→3\" underneath the limit symbol. Below this, the substitution process begins with \"= 3² + 5(3) - 4\" appearing at 0:03. The text remains static throughout the frames, with no additional visual elements or actions occurring. The scene is solely focused on the mathematical expression and its evaluation, with no people, objects, or other visual components present.",

Qwen3-VL-235B-A22B-Instruct: From 0.0s to 0.4s, white handwritten text appears on a black background, showing a limit problem: \"lim [x² + 5x - 4]\" with \"x→3\" beneath it, followed by the step \"= 3² + 5(3) - 4\" being written progressively. At 0.0s, only the limit expression is visible. By 0.1s, \"= 3² + 5(\" is written. At 0.2s, it becomes \"= 3² + 5(3) -\". At 0.3s, the full substitution \"= 3² + 5(3) - 4\" is complete. At 0.4s, a new line with \"=\" appears below, indicating the next step in the calculation. No people, objects, or scene elements other than the handwritten mathematical text are visible.",

Struct-VidOCR


```
{
 "ocr_segments": [
   {
    "id":  "1",
    "position": "top_left",
    "content_type": "mathematical_expression",
    "source_type": "burned_in_graphic",
    "movement_pattern": "static",
    "persistence_pattern": "continuous",
    "text": "lim [x² + 5x - 4]",
    "start_time": 0.0,
    "end_time": 5.5,
    "occlusion_level": "none"
   },
   {
    "id":  "2",
    "position": "middle_left",
    "content_type": "mathematical_expression",
    "source_type": "burned_in_graphic",
    "movement_pattern": "static",
    "persistence_pattern": "continuous",
    "text": "x→3",
    "start_time": 0.0,
    "end_time": 5.5,
    "occlusion_level": "none"
   },
   {
    "id": "3",
    "position": "center-left",
    "content_type": "mathematical_equation",
    "source_type": "burned_in_graphic",
    "movement_pattern": "static",
    "persistence_pattern": "sequential_change",
    "variants": [
    {"text": "= 3² +", "start_time": 0.0, "end_time": 1.0},
    {"text": "= 3² + 5", "start_time": 1.0, "end_time": 2.0},
    {"text": "= 3² + 5(3", "start_time": 2.0, "end_time": 3.0},
    {"text": "= 3² + 5(3) -", "start_time": 3.0, "end_time": 4.0},
    {"text": "= 3² + 5(3) - 4", "start_time": 4.0, "end_time": 5.5}],
    "occlusion_level": "none"
   },
   {
    "id": "4",
    "position": "bottom-center",
    "content_type": "mathematical_expression",
    "source_type": "burned_in_graphic",
    "movement_pattern": "static",
    "persistence_pattern": "continuous",
    "text": "=",
    "start_time": 4.5,
    "end_time": 5.5,
    "occlusion_level": "none" }],
```


"caption": "From 0.0s to 5.5s, a mathematical limit problem is displayed on a black background. At the top-left, the expression 'lim [x² + 5x - 4]' is shown, with 'x->3' directly below it, indicating the limit as x approaches 3. Below this, the solution is written sequentially, starting with " 3² +' and ending with"= 3² + 5(3) - 4\", appearing gradually from 1.0s onward. At 4.5s, a new line of calculation "=" appears below this expression, indicating the next step in the problem. No people, objects, or scene layout changes are visible beyond the text on the black background." }

Figure 12: Selected incremental-text example. A mathematical limit problem is solved step by step. Struct-VidOCR separates the static formula and represents incremental substitutions with `sequential_change` variants over the sampled 5.5-second span.