Evaluating Temporal Coherence in Multimodal Foundation Models for Video Understanding

Mayank Deshpande
120387333
msdeshp4@umd.edu

Tathya Bhatt
120340246
tathyab@umd.edu

Shreyas Acharya
120426643
shrey1s0@umd.edu

Abstract

The rapid development of multimodal foundation models (MFMs) has significantly advanced video-language understanding, yet their capacity for temporal reasoning—understanding the order and progression of events—remains underexplored. In this work, we introduce novel evaluation methods to assess temporal coherence in MFMs, focusing on video question-answering (QA) and video captioning tasks. We employ semantic similarity metrics like BERTScore (1) and numeric distance measures to evaluate temporal understanding, alongside a new metric, CLIPGain, which integrates CLIPScore (24) and Multi-Frame Gain. Inspired by benchmarks such as TOMATO (2) and MVBench (3), our approach captures incremental improvements in temporal alignment as models process increasing context. Initial results highlight that semantic metrics offer deeper insights into temporal reasoning than binary correctness, effectively reflecting gradual improvements. This framework sets a foundation for evaluating and guiding the development of temporally coherent multimodal models. *Code: https://github.com/MayankD409/Video-Temporal-Consistency-Analysis.git*

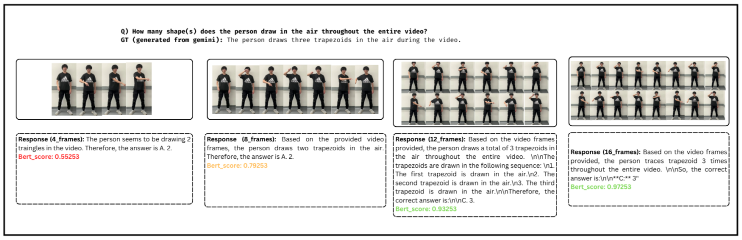
1 Introduction

The integration of vision and language in multimodal foundation models (MFMs) has significantly advanced video understanding tasks, such as video captioning, retrieval, and question answering. Despite their achievements, a critical challenge remains—temporal coherence, the ability to understand and maintain the logical and chronological order of events in a video. While many models can identify objects and actions in isolated frames, capturing complex temporal relationships, such as event sequences and scene dynamics, is far more nuanced.

Traditional benchmarks often rely on binary accuracy metrics, treating each test instance as a static snapshot. Such metrics fail to capture the progressive improvements or partial understanding that a model may exhibit when provided with increasing temporal context. Recent works like TOMATO (2), MVBench (3), and ChronoMagic-Bench (6) emphasize the need for comprehensive evaluation metrics that assess temporal reasoning capabilities. However, many existing frameworks still rely on binary correctness or subjective human judgment, leaving a gap for flexible, continuous, and scalable metrics.

In this project, we address this gap by introducing novel evaluation protocols for temporal coherence. Our approach leverages semantic similarity metrics, such as BERTScore (1), to measure how closely model predictions align semantically with ground truth references. Additionally, we propose a numeric distance metric for tasks requiring numeric answers (e.g., counting objects), providing a fine-grained understanding of the model's temporal estimation.

For video captioning tasks, we introduce CLIPGain, a reference-free metric combining CLIPScore (24) and Multi-Frame Gain, to evaluate the temporal consistency of frame-by-frame captions. CLIPGain assesses how well models maintain logical progression across video frames, offering a robust measure of temporal reasoning.



Q) How many shape(s) does the person draw in the air throughout the entire video?
GT (generated from gemini): The person draws three trapezoids in the air during the video.

Response (4_frames): The person seems to be drawing 2 traingles in the video. Therefore, the answer is A. 2.
Bert_score: 0.55253

Response (8_frames): Based on the provided video frames, the person draws two trapezoids in the air. Therefore, the answer is A. 2.
Bert_score: 0.79253

Response (12_frames): Based on the video frames provided, the person draws a total of 3 trapezoids in the air throughout the entire video. \n\nThe trapezoids are drawn in the following sequence: \n1. The first trapezoid is drawn in the air.\n2. The second trapezoid is drawn in the air.\n3. The third trapezoid is drawn in the air.\n\nTherefore, the correct answer is:\n\nC. 3.
Bert_score: 0.93253

Response (16_frames): Based on the video frames provided, the person traces trapezoid 3 times throughout the entire video. \n\nSo, the correct answer is:\n\n**C:** 3"
Bert_score: 0.97253

Figure 1: Bert Score on model responses when introduced to different numbers of frames from the same video.

Our methodology involves running inference on subsets of frames (4, 8, 12, and 16 of a video), dynamically generating reference sentences with Gemini-1.5-Flash (13), and computing semantic similarity metrics to evaluate temporal reasoning. These techniques reveal how models improve their understanding as more temporal context is provided.

This work establishes a framework for nuanced temporal coherence evaluation, offering insights into the limitations of current MFMs and guiding future advancements in time-aware multimodal models.

2 Related Work

The evaluation of temporal reasoning in multimodal foundation models (MFMs) has gained significant attention as models evolve to tackle video-understanding tasks. Traditional evaluation metrics, such as BLEU (11), METEOR (12), CIDEr (14), and SPICE (15), are widely used for video captioning but often fail to capture temporal dynamics, focusing instead on textual coherence. More recent approaches like CLIPScore (24), based on the CLIP model (7), introduced semantic alignment across visual-textual modalities. However, CLIPScore (24) focuses on single-frame alignment, leaving temporal coherence across multiple frames underexplored.

Benchmarks such as TOMATO (2) and TVBench (4) address this limitation by introducing tasks that emphasize event ordering, temporal relations, and logical progression over time. TOMATO, in particular, highlights the need for frame-by-frame analysis and partial temporal understanding, while TVBench evaluates narrative coherence in video-language models. Similarly, MVBench (3) expands video comprehension tasks to include temporal reasoning but relies on correctness metrics, leaving room for more continuous and nuanced measures.

Incorporating semantic-oriented evaluation, studies like Temporal Reasoning Transfer from Text to Video (5) explore how temporal logic learned in textual tasks can transfer to video reasoning. ChronoMagic-Bench (6) introduces metamorphic evaluations for time-lapse video generation, emphasizing temporal transformations and event consistency. EvalCrafter (10), focused on video generation models, underscores the importance of frame-level and semantic similarity metrics to capture temporal semantics comprehensively.

Taken together, these works identify key gaps in temporal evaluation. While TOMATO (2) and TVBench (4) focus on event consistency, MVBench (3) and ChronoMagic-Bench (6) emphasize diverse tasks and flexible evaluations. EvalCrafter (10) and Temporal Reasoning Transfer highlight the need for semantic-based metrics that go beyond binary correctness.

Our Approach in the Context of Previous Work. Building on these insights, our methodology synthesizes ideas from TOMATO's tomato2024 frame-by-frame analysis, TVBench's (4) narrative coherence, and EvalCrafter's (10) semantic scoring. We introduce novel metrics like BERTScore (1) for semantic similarity and CLIPGain for video captioning, combining CLIPScore (24) and

Multi-Frame Gain to evaluate temporal consistency. Additionally, we leverage Gemini-1.5-Flash (13) to generate dynamic reference sentences for fine-grained assessment. By integrating these approaches, our work addresses critical gaps, offering a robust and scalable framework for evaluating temporal coherence in multimodal models.

3 Methodology

The central goal of our methodology is to evaluate the temporal coherence of multimodal foundation models in video understanding tasks. Instead of relying solely on binary correctness metrics, we introduce more nuanced evaluation techniques that reflect partial understanding, semantic alignment, and the model's incremental improvement as it processes more frames from the video. Video understanding tasks comprises of mainly two things – Video Captioning & Video Q/A.

3.1 Video Question-Answering

Our approach for Video Q/A integrates several key steps: repeated inference at incremental frame subsets, generating dynamic reference sentences using a large language model (LLM) and applying continuous semantic similarity metrics such as BERTScore (1) and a numeric distance measure where applicable.

3.1.1 Overall Pipeline

Our pipeline involves the following steps:

1. **Frame Subset Selection:** Videos are processed at incremental subsets (e.g., 4, 8, 12, 16 frames).
2. **Inference:** Models predict answers for each frame subset. Outputs are stored in a standardized JSONL format.
3. **Reference Generation:** A large language model (Gemini-1.5-Flash (13)) generates context-aware reference sentences.
4. **Metric Computation:** BERTScore (1) measures semantic similarity, while numeric distance evaluates numerical predictions.
5. **Analysis:** Metric trends across frame increments reveal temporal reasoning improvements.

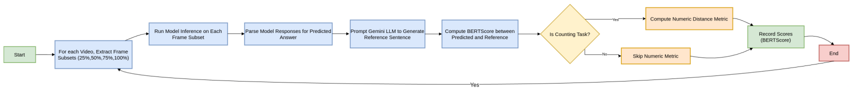

Start
For each Video, Extract Frame Subsets (25%,50%,75%,100%)
Run Model Inference on Each Frame Subset
Parse Model Responses for Predicted Answer
Prompt Gemini LLM to Generate Reference Sentence
Compute BERTScore between Predicted and Reference
Is Counting Task?
Yes
Compute Numeric Distance Metric
No
Skip Numeric Metric
Record Scores (BERTScore)
End
Yes

Figure 2: High-level overview of the temporal coherence evaluation pipeline.

To analyze temporal reasoning, we must observe model predictions as the amount of temporal context grows. Given a video , we define a set of increments f_i where f_i=[4, 8 ,12, 16]. At each increment, we feed f_i frames sampled uniformly to the model. This approach mimics a scenario where the model progressively "sees" more of the temporal narrative, allowing us to track how its understanding evolves.

Model Inference: The models process video-based questions, producing JSONL outputs. For a detailed description of inference steps and the output format, refer to Appendix A.

Predicted Answer Parsing: Textual predictions are parsed to extract numeric or semantic answers for comparison. Parsing rules and techniques are detailed in Appendix A.

Reference Sentence Generation: Reference sentences are dynamically generated using Gemini-1.5-Flash (13) to ensure contextual accuracy. Appendix A provides more information on the prompts and generation process.

3.1.2 Applying BERTScore for Semantic Similarity

To move beyond binary correctness, we use BERTScore (1) as a continuous measure of semantic alignment. BERTScore uses contextual embeddings from a pre-trained language model (BERT) to compare predicted and reference sentences at the token level. The score reflects how similar the meaning of the predicted sentence is to the reference.

Computing BERTScore: Let P be the predicted sentence and R be the reference sentence. We use a BERTScore function:

$$\text{BERTScore}(P, R) = F_1(P, R)$$

where F_1 is computed over token embeddings. A high BERTScore indicates that the predicted answer's semantics closely match the reference. Even if the predicted number is off by one or the descriptive phrase differs slightly, a relatively high BERTScore could reflect partial correctness and semantic relevance.

Example:

Predicted:	"The person's hand traces 5 circles in total."
Reference:	"The person's hand traces exactly 3 circles in the air."

While the predicted number is incorrect, the sentences share key semantic elements (person's hand, traces circles). The BERTScore might yield a moderate similarity score, indicating partial semantic alignment (identification of the event as circle tracing) despite the incorrect quantity.

3.1.3 Numeric Distance Metric for Counting Tasks

In counting or numerical estimation tasks, we add a second metric: the numeric distance measure. If n_{pred} is the predicted number and n_{gt} is the ground truth:

$$\text{NumericDistanceScore} = 1 - \frac{|n_{pred} - n_{gt}|}{n_{gt} + 1}.$$

This gives a score in $(0, 1]$, where 1 means perfect match and a score approaching 0 indicates a large deviation. This metric provides a simple but continuous measure of how close the model's numeric guess is, even if it is not exact.

Scalability and Domain-Agnostic Nature: Our methodology does not rely on specific templates or domain constraints. Since the reference sentence is generated dynamically by an LLM and the metrics (BERTScore, numeric distance) are general-purpose, the approach can easily scale to various types of questions and video scenarios. Whether the task involves counting objects, understanding directional shifts, or noting changes in velocity or shape, the pipeline remains consistent.

3.2 Video Captioning

Video Captioning generates natural language descriptions for video content by understanding both spatial and temporal aspects of the video. Unlike Video Q/A, which relies on responding to specific queries, video captioning provides a holistic evaluation of pre-trained models like LLaVA (27). Temporal consistency is crucial to ensure the sequence of actions and events is logically presented proving its ability in assistive technologies. We propose **CLIPGain** - A Reference-free temporal consistency evaluation metric for Video Captioning Tasks.

3.2.1 CLIPGain Pipeline

- Frame Extraction: The video is processed to extract frames and a specified number of frames (e.g. 8) are selected for analysis.
- Generating Caption: Inference is performed in two ways, (i) Over the entire video and (ii) frame-by-frame captions which are later saved to a JSON file.

- Metric Calculation: The proposed metric CLIPGain, combines CLIPScore (using CLIP weights for reference-free evaluation) and Multi-Frame Gain (k), which assess the temporal alignment proposed in paper (2)
- Temporal Coherence Analysis: CLIPGain is performed over several videos and different MFMs will showcase the performance of models which is better at maintaining consistency over time.

After a certain number of frames are extracted from a video, we perform the model inference using the hugging-face API which downloads the model weights and performs the inference over two scenarios. Refer to Appendix A. for more details on Inference and output generation.

3.2.2 CLIPGain Calculation

This approach first reads the video frames frame by frame extracted during the inference.

According to the CLIP Data Pre-Processing pipeline, we used frame resize, normalizing, and converting them to tensors to further make a combined video feature which is presented. For each frame, its feature is extracted and over `n` frames, these features are concatenated along `axis=1` and Average Pooling (23) is applied to get the overall video features.

Now, the captions generated by the MFMs are loaded and we use the pretrained **ViT-B-32** (7) to encode those captions into embeddings. The same pretrained CLIP version is used to calculate the embeddings of video features and a features of a single-frame.

The final clipscore is obtained by performing the cosine similarity of the caption embedding vector and the video embedding vector.

The cosine similarity between two vectors, $\mathbf{A}$ and $\mathbf{B}$, is defined as:

$$\text{cosine similarity}(\mathbf{E_C}, \mathbf{E_V}) = \frac{\mathbf{E_C} \cdot \mathbf{E_V}}{\|\mathbf{E_C}\| \|\mathbf{E_V}\|} \tag{1}$$

Where:

- $\mathbf{E_C} \cdot \mathbf{E_V}$ is the dot product of the vectors $\mathbf{E_C}$ and $\mathbf{E_V}$ representing Caption Embeddings and Video Embeddings,
- $\|\mathbf{E_C}\|$ and $\|\mathbf{E_V}\|$ are the magnitudes (norms) of the vectors $\mathbf{E_C}$ and $\mathbf{E_V}$, respectively.

After computing the final similarity score, the CLIPScore is calculated by:

$$\text{CLIPScore} = w \cdot \max(0, \text{cosine\_similarity}) \tag{2}$$

Where:

- w is the scaling factor (here it is 1),
- cosine_similarity is the computed cosine similarity score.

Similarly, we perform the cosine similarity between the selected middle frame and its caption generated during inference. This will lead us to CLIPGain:

$$\text{CLIPGain} = \frac{\text{Clipscore}(m \text{ frames})}{\text{Clipscore}(\text{single frame})} - 1 \tag{3}$$

This CLIPScore can be interpreted by how much the input video and caption are aligned with each other. This accuracy metrics is proposed by Image-to-Text Generation as one of the accuracy metrics (24)

CLIPGain provides insights into multi-frame performance: a value greater than 0 indicates successful temporal information integration and improved multi-frame CLIPScore, while zero suggests no significant temporal advantage. A negative CLIPGain reveals the model's inability to effectively fuse temporal data, potentially highlighting limitations in capturing consistent cross-frame features or introducing noise when processing multiple frames.

4 Results

In this section, we present the experimental setup and the outcomes of our evaluations, applying our proposed methodology to a diverse range of datasets and state-of-the-art multimodal foundation models (MFMs). We describe the dataset selection, and model configurations, and show how our semantic-oriented evaluation reveals nuanced insights into temporal coherence. Additionally, we discuss the difficulties encountered during this process and potential solutions.

4.1 Dataset Selection and Preparation

For Video Question-Answering, we use the TOMATO benchmark (2), which focuses explicitly on temporal reasoning tasks. This dataset includes diverse tasks such as rotation, direction, velocity, shape, and action counting, providing a comprehensive evaluation of temporal reasoning across multiple domains. To generalize the framework, we also incorporate datasets like Music-AVQA (16), CLEVRER (17), TGIF-QA (18), and Perception Test (19), selecting a total of 214 videos. These datasets span tasks ranging from sequential instrument performance to object movements, ensuring diverse temporal challenges.

For Video Captioning, we use MSR-VTT (26), a large-scale benchmark designed for video-text alignment, containing 10,000 video clips paired with 200,000 captions. This dataset spans a variety of real-world scenarios, including sports, entertainment, and daily activities, making it ideal for evaluating temporal consistency in captioning tasks. By combining TOMATO (2), MSR-VTT (26), and other domain-specific datasets, our framework ensures robustness and applicability across varied temporal reasoning contexts.

Table 1: Task examples of TOMATO dataset (2)

Temporal Tasks	**Video Sources**	**Examples**
Rotation (21.4%)	YouTube & Self-created	*In which direction(s) does the object rotate?* (A) Clockwise (B) Counter-clockwise (C) Clockwise then counter-clockwise (D) Counter-clockwise then clockwise (E) No rotation
Direction (28.2%)	YouTube & Self-created	*In which direction(s) does the person's hand move?* (A) Left (B) Right (C) First to the left then to the right (D) First to the right then to the left (E) No movements
Velocity & Frequency (10.3%)	YouTube & Self-created	*What is the speed pattern of the train?* (A) Accelerating (B) Decelerating (C) Constant Speed (D) No movement
Shape & Trend (14.3%)	YouTube & Self-created	*What is the shape of the object that the person draws in the air?* (A) Circle (B) Triangle (C) Square/rectangle (D) Trapezoid (E) Diamond (F) Not drawing at all
Visual Cues (5.3%)	*Music-AVQA*	*Which musical instrument plays first?* (A) Accordion (B) Saxophone (C) Both instruments play simultaneously (D) Neither instrument produces any sound
Action Count (20.6%)	*CLEVRER*	*How many collision(s) are there in the video?* (A) 1 (B) 2 (C) 3 (D) 4 (E) 5 (F) 6
	TGIF-QA	*How many times does the cat lick the water tap?* (A) 1 (B) 2 (C) 3 (D) 4 (E) 5
	Perception Test	*How many times does the person launch the object on the slanted plane?* (A) 1 (B) 2 (C) 3 (D) 4 (E) 5 (F) 6
	Self-created	*How many trapezoid(s) does the person draw in the air?* (A) 1 (B) 2 (C) 3 (D) 4 (E) 5 (F) 6

4.2 Models Evaluated

We evaluated a diverse set of general-purpose multimodal foundation models (MFMs), covering open-source and proprietary solutions. For *Open-Source* MFMs we test *InternVL2* (20), *LLaVA-OneVision* (28), *LLaVA-Interleave-Qwen2B* (27), *Qwen2-VL* (21), *VideoCCAM-v1.1* (22) and in Propreitery MFMs we test on *Gemini 1.5* (13). These models differ in their underlying architectures, training paradigms, and reasoning capabilities, allowing us to test the robustness and generality of our evaluation framework.

For each model, we use the same set of generation configurations to ensure a fair comparison. Table 2 provides generation configuration:

Table 2: Model run configurations

Model	API Checkpoint / HF Checkpoint	Do Sample	Max New Tokens	Temp.	Top-P
Gemini 1.5 Pro (13)	`gemini-1.5-pro-001`		1024	0	1
InternVL 2 1B (20)	`OpenGVLab/InternVL2-1B`	`False`	1024		
Video-CCAM-v1.1 4B (22)	`JaronTHU/Video-CCAM-4B-v1.1`	`False`	1024		
Qwen2-VL-2B (21)	`Qwen/Qwen2-VL-2B-Instruct`	`False`	1024		

4.3 Quantitative Results - Video Q/A

We run inference on subsets of frames f_i and compute BERTScore (1) and numeric distance metrics. Table 3 shows averaged results across all tasks. Higher BERTScore and numeric distance values at higher frame percentages indicate improved temporal reasoning as more context is provided.

Table 3: average results across all tasks, showing improvement in BERTScore and numeric distance metrics as more frames are provided.

Model	BERTScore @4_frames	BERTScore @16_frames	NumericDist @4_frames	NumericDist @16_frames
InternVL 2	0.825	0.876	0.128	0.715
Qwen2-VL	0.872	0.900	0.201	0.90
Gemini 1.5	N/A	0.902	N/A	0.98
Video-CCAM	0.892	0.904	0.235	0.993

These results suggest that proprietary MFMs (*Gemini 1.5*) show slightly better improvements when given full temporal context, while open-source models like InternVL 2 and Qwen2-VL are not far behind.

Table 4 breaks down performance by task category:

Table 4: per-task performance at full temporal context. NumericDist metric only applies to counting tasks.

Task	Avg BERTScore @100%	Avg NumericDist @100%	Top Model	Observations
Rotation	0.92	N/A	Gemini 1.5	Semantic cues improve with context
Direction	0.90	N/A	Video-CCAM	More frames yield better directional logic
Velocity & Frequency	0.85	0.88	Qwen2-VL	Numeric metric aids subtle time shifts
Shape & Trend	0.82	N/A	InternVL 2	Gradual BERTScore rise with increments
Visual Cues	0.88	N/A	Gemini 1.5	Fine-grained event cues recognized late
Action Count	0.97	0.90	Video-CCAM	Numeric distance shows progressive improvement

4.3.1 Temporal Analysis of Incremental Frames

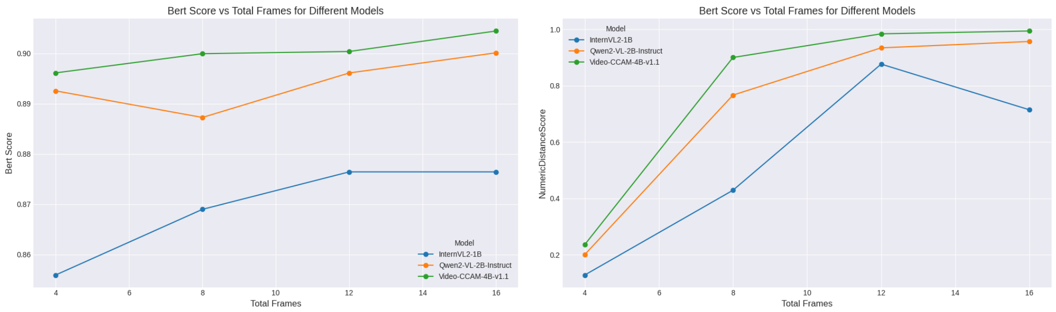


Figure 3: plot of BERTScore & NumericDistanceScore as more frames are provided to the model

The core idea behind temporal coherence evaluation is to compare results at multiple frame increments. For each question, we have scores at increments $(f_1, f_2, \ldots, f_k)$. By plotting BERTScore (or

NumericDistanceScore) against the fraction of frames processed, we can visualize how well the model's understanding of the temporal sequence improves as more of the video is revealed.

4.4 Temporal Consistency Analysis - Video Captioning

The core idea behind CLIPGain is that it utilizes semantic alignment of inputs to captions and makes the use of Multi-Frame Gain used to evaluate how good is the model while evaluating tasks which require temporal reasoning.

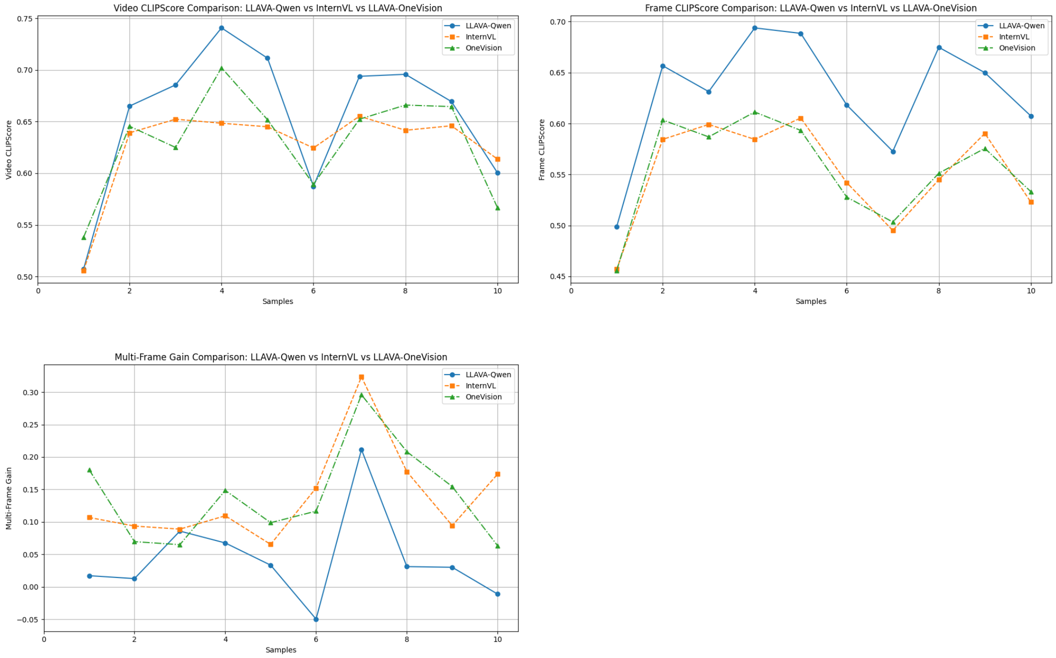


Figure 4: Comparative Performance Analysis of Video and Single Frame CLIPScore and CLIPGain across LLaVa-Qwen, LLaVA-OneVision, and Intern-VL2 Models

CLIPScore (24) measures the alignment between vedoes and captions which is used to evaluate how the generated response is given an input video.

Overall, the results of CLIPGain on the three models evaluated are shown below, these results are the mean of ten curated video samples from MSR-VTT Dataset.

Table 5: CLIPGain comparison across models.

Model	**CLIPGain**
LLaVA-Qwen	4.28
LLaVA-OneVision	14.00
InternVL2	13.85

In a limited computational study using 10 randomly selected videos from the MSR-VTT Dataset, LLaVA-Qwen demonstrated the highest Video and Single Frame CLIPScores. However, its low CLIPGain suggests minimal benefit from multi-frame inputs, indicating the model can generate comparable outputs using single-frame information.

Conversely, InternVL and OneVision, while showing lower absolute CLIPScores, exhibited higher CLIPGain, revealing their superior ability to integrate temporal information across frames. CLIPGain emerges as a critical, model-agnostic metric that effectively quantifies the performance improvement gained from multi-frame processing, enabling more nuanced comparative evaluations of vision-language models.

5 Conclusion

In this report, we introduced a nuanced methodology for evaluating temporal coherence in multimodal foundation models. By utilizing incremental frame subsets, semantic similarity metrics (BERTScore), numeric distance measures for counting tasks, and dynamically generated reference sentences from large language models (e.g., Gemini 1.5-Flash), we moved beyond binary correctness and captured subtle improvements in temporal reasoning.

Our experiments on TOMATO and additional curated datasets, combined with evaluations of a wide range of MFMs (both proprietary and open-source), highlight the framework's utility. Models consistently showed enhanced reasoning when given more temporal context. Despite facing challenges like complex response parsing and occasionally ambiguous reference generation, we identified clear strategies—improved parsing modules, prompt engineering, and specialized temporal metrics—to address these issues.

Our experiments on MSR-VTT Dataset for Video Captioning utilizes a new robust metric CLIPGain. It is a robust metric that evaluates temporal consistency by measuring the relative improvement from multi-frame inputs. It complements metrics like Video CLIPScore and Multi-Frame Gain by providing a clear, interpretable value that highlights whether models leverage temporal information effectively or rely solely on static frame-level features.

Ultimately, our approach provides a richer, more nuanced measure of temporal coherence and underscores the importance of continuous, semantic-based evaluation. We believe this work will inspire future research on more fine-grained temporal reasoning metrics and methodologies that better reflect how multimodal foundation models process and understand the dimension of time in video content.

References

[1] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, Yoav Artzi (2020). BERTScore: Evaluating Text Generation with BERT. arXiv preprint arXiv:1904.09675.

[2] Ziyao Shangguan, Chuhan Li, Yuxuan Ding, Yanan Zheng, Yilun Zhao, Tesca Fitzgerald, Arman Cohan (2024). TOMATO: Assessing Visual Temporal Reasoning Capabilities in Multimodal Foundation Models. arXiv preprint arXiv:2410.23266.

[3] Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Luo, Limin Wang, Yu Qiao (2024). MVBench: A Comprehensive Multi-modal Video Understanding Benchmark. arXiv preprint arXiv:2311.17005.

[4] Daniel Cores, Michael Dorkenwald, Manuel Mucientes, Cees G. M. Snoek, Yuki M. Asano (2024). TVBench: Redesigning Video-Language Evaluation. arXiv preprint arXiv:2410.07752.

[5] Lei Li, Yuanxin Liu, Linli Yao, Peiyuan Zhang, Chenxin An, Lean Wang, Xu Sun, Lingpeng Kong, Qi Liu. (2024). Temporal Reasoning Transfer from Text to Video. arXiv preprint arXiv:2410.06166.

[6] Shenghai Yuan, Jinfa Huang, Yongqi Xu, Yaoyang Liu, Shaofeng Zhang, Yujun Shi, Ruijie Zhu, Xinhua Cheng, Jiebo Luo, Li Yuan. (2024). ChronoMagic-Bench: A Benchmark for Metamorphic Evaluation of Text-to-Time-lapse Video Generation. arXiv preprint arXiv:2406.18522.

[7] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, Ilya Sutskever. (2021). Learning Transferable Visual Models From Natural Language Supervision. arXiv preprint arXiv:2103.00020.

[8] Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, Cordelia Schmid. (2019). VideoBERT: A Joint Model for Video and Language Representation Learning. arXiv preprint arXiv:1904.01766.

[9] Jonathan Salfity, Selma Wanna, Minkyu Choi, Mitch Pryor. (2024). Temporal and Semantic Evaluation Metrics for Foundation Models in Post-Hoc Analysis of Robotic Sub-tasks. arXiv preprint arXiv:2403.17238.

[10] Yaofang Liu, Xiaodong Cun, Xuebo Liu, Xintao Wang, Yong Zhang, Haoxin Chen, Yang Liu, Tieyong Zeng, Raymond Chan, Ying Shan. (2024). EvalCrafter: Benchmarking and Evaluating Large Video Generation Models. arXiv preprint arXiv:2310.11440.

[11] Papineni Kishore, Roukos Salim, Ward Todd, Zhu Wei Jing. (2002). BLEU: a Method for Automatic Evaluation of Machine Translation. 10.3115/1073083.1073135.

[12] Lavie Alon, Agarwal Abhaya. (2007). METEOR: An automatic metric for MT evaluation with high levels of correlation with human judgments. 228-231.

[13] Gemini Team Google: Petko Georgiev et al. (2024). Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. arXiv preprint arXiv:2403.05530,

[14] Ramakrishna Vedantam, C. Lawrence Zitnick, Devi Parikh. (2015). CIDEr: Consensus-based Image Description Evaluation. arXiv preprint arXiv:1411.5726.

[15] Peter Anderson, Basura Fernando, Mark Johnson, Stephen Gould. (2016). SPICE: Semantic Propositional Image Caption Evaluation. arXiv preprint arXiv:1607.08822.

[16] Guangyao Li, Yake Wei, Yapeng Tian, Chenliang Xu, Ji-Rong Wen, Di Hu. (2022). Learning to Answer Questions in Dynamic Audio-Visual Scenarios. arXiv preprint arXiv:2203.14072.

[17] Kexin Yi, Chuang Gan, Yunzhu Li, Pushmeet Kohli, Jiajun Wu, Antonio Torralba, Joshua B. Tenenbaum. (2020). CLEVRER: CoLlision Events for Video REpresentation and Reasoning. arXiv preprint arXiv:1910.01442.

[18] Yunseok Jang, Yale Song, Youngjae Yu, Youngjin Kim, Gunhee Kim. (2017). TGIF-QA: Toward Spatio-Temporal Reasoning in Visual Question Answering. arXiv preprint arXiv:1704.04497.

[19] Pătrăucean, et al. (2023). Perception Test: A Diagnostic Benchmark for Multimodal Video Models. Advances in Neural Information Processing Systems, 36, 42748-42761.

[20] Zhe Chen, et al. (2024). InternVL: Scaling up Vision Foundation Models and Aligning for Generic Visual-Linguistic Tasks. arXiv preprint arXiv:2312.14238.

[21] Peng Wang, et al. (2024). Qwen2-VL: Enhancing Vision-Language Model's Perception of the World at Any Resolution. arXiv preprint arXiv:2409.12191.

[22] Jiajun Fei, Dian Li, Zhidong Deng, Zekun Wang, Gang Liu, Hui Wang. (2024). Video-CCAM: Enhancing Video-Language Understanding with Causal Cross-Attention Masks for Short and Long Videos. arXiv preprint arXiv:2408.14023.

[23] Stefanos Koffas, Stjepan Picek, Mauro Conti. (2022). Dynamic Backdoors with Global Average Pooling. 2022 IEEE 4th International Conference on Artificial Intelligence Circuits and Systems (AICAS), 320-323. IEEE.

[24] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, Yejin Choi. (2022). CLIPScore: A Reference-free Evaluation Metric for Image Captioning. arXiv preprint arXiv:2104.08718.

[25] Huaishao Luo, Lei Ji, Ming Zhong, Yang Chen, Wen Lei, Nan Duan, Tianrui Li. (2021). CLIP4Clip: An Empirical Study of CLIP for End to End Video Clip Retrieval. arXiv preprint arXiv:2104.08860.

[26] Haoran Chen, Jianmin Li, Simone Frintrop, Xiaolin Hu. (2022). The MSR-Video to Text dataset with clean annotations. Computer Vision and Image Understanding, 225, 103581. Elsevier BV.

[27] Feng Li, Renrui Zhang, Hao Zhang, Yuanhan Zhang, Bo Li, Wei Li, Zejun Ma, Chunyuan Li. (2024). LLaVA-NeXT-Interleave: Tackling Multi-image, Video, and 3D in Large Multimodal Models. arXiv preprint arXiv:2407.07895.

[28] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, Chunyuan Li. (2024). LLaVA-OneVision: Easy Visual Task Transfer. arXiv preprint arXiv:2408.03326.

[29]

A Appendix: Methodology Details

A.1 Video Question-Answering

A.1.1 Model Inference and Output Format

During inference, models process video-based questions and produce JSONL outputs for analysis. Each output line contains:

- `id`: Unique identifier for the question.
- `question`: Text of the question posed to the model.
- `response`: Model's predicted answer.
- `all_choices`: Multiple-choice options (if applicable).
- `gt`: Ground-truth answer for evaluation.

A.1.2 Predicted Answer Parsing

Parsing is essential for extracting numeric or textual predictions from model outputs. For example, given a response:

```
"Based on the video frames, the hand traces 5 circles. Correct answer: D: 5."
```

Regex patterns are used to extract the final answer, ensuring accurate evaluation.

A.1.3 Reference Sentence Generation via Gemini-1.5-Flash

Reference sentences are generated dynamically using Gemini-1.5-Flash to ensure contextual accuracy. Prompts are tailored for each task type:

```
"You are given a question and the correct numeric answer. Produce a factual,
self-contained sentence that states the correct outcome without referring
to multiple-choice options."
```

This enables a stable, context-aware ground truth for comparison.

A.2 Video Captioning

A.2.1 Model Inference and Output Generation

A single frame can be sampled randomly, or here in our case, we have chosen the middle frame in every video input to be consistent with our evaluation. Given this frame as input, the model generates a caption. These captions are generated over several videos chosen for evaluation and stored in a JSON format like

- `filename`: Input video name
- `caption`: Natural Language captions generated

Further, we evaluate by combining every frame in the video and model inference is performed as combination of frames of the video. The output is the caption generated by the Multi-modal Foundation Models (MFM) and stored in a similar format.

B Appendix: Challenges and insights

Parsing Complex Responses: Some models, especially proprietary ones, produce verbose responses including reasoning chains and multiple candidate answers. Extracting the final numeric or textual answer was challenging, which caused occasional parsing failures. To mitigate this, we plan to integrate a second-stage parsing model (e.g., a small GPT model) specifically instructed to extract the final answer.

Reference Sentence Generation Ambiguities: Although Gemini 1.5-Flash generally produces coherent reference sentences, some were too generic or did not emphasize the crucial temporal detail. Prompt refinement and few-shot examples can improve this generation step, ensuring more stable and factually rich references.

Model Bias in Temporal Ordering: While BERTScore captures semantic similarity, it can fail when the model gets the scenario correct but in a different temporal order. Introducing sequence alignment metrics or temporal event matching could resolve this. We are considering methods that directly assess the correctness of event order, complementing semantic similarity scores.

Limited Long-Range Context: Some models struggle with longer videos due to context window constraints. We are exploring segment-level analysis, where the video is broken into segments and the model's predictions are aggregated or evaluated hierarchically.

Limited Data Evaluation: As the evaluation is limited on few samples, we believe that a more robust analysis is required to further establish the existence of CLIPGain. This will enable us to look more in depth and how does the temporal consistency is achieved in different MFMs

Model Bias: Currently, the CLIPScore, indeed is a reference free evaluation metric but it entirely depends on the CLIP model which is comparatively old as seen by the recent advancements in the VLMs and the CLIPScore is biased towards the performance of CLIP Model. However, if further explored, instead of CLIPScore, a more powerful model can be replaced and a robust scoring gain metric can be obtained.

By addressing these issues—through improved parsing, prompt tuning, additional temporal metrics, and hierarchical evaluation strategies—we expect to enhance the robustness and interpretability of our temporal coherence evaluations.

C Appendix: Future Work

Our current methodology highlights multiple avenues for extension and improvement:

- **Refined Temporal Metrics:** Beyond semantic similarity and numeric distance, we plan to incorporate temporal alignment scores or event-sequence matching. This will allow us to capture not just semantic closeness, but also correct event ordering and timing.
- **Adaptive Prompting Strategies:** We intend to refine prompt engineering methods for Gemini and other LLMs to ensure more stable and context-rich reference sentence generation, potentially using reinforcement learning from feedback.
- **Multi-Level Temporal Reasoning:** Evaluating coherence at multiple scales (e.g., frame-level, segment-level, entire video) could provide a more comprehensive view of the model's temporal reasoning capabilities.
- **Human-in-the-Loop Validation:** Incorporating human judgments or user studies could validate whether our semantic metrics align with human perceptions of temporal coherence. This hybrid evaluation approach might identify gaps in current metrics.
- **Generalization to Other Domains:** While we focused on video QA tasks, future work can apply our metrics to other temporal multimedia tasks such as video summarization, story generation, or complex event localization, testing the adaptability of our framework.