Diagnosing and Addressing Temporal Reasoning Limitations in Video-LLaVA

Arne Eichholtz [1] **Caspar de Jong** [1] **Daniel Uyterlinde** [1] **Emma Kasteleyn** [1] **Freek Byrman** [1] **Jutte Vijverberg** [1]

GitHub: https://github.com/Emmakast/Modification-on-Video-LLaVA

Abstract

Video Large Language Models (Vid-LLMs) have demonstrated impressive capabilities in general video understanding, yet genuine temporal reasoning remains a persistent challenge. In this paper, we systematically investigate the temporal limitations of Video-LLaVA using the TempCompass benchmark. Additionally, we introduce several interventions aimed at improving temporal reasoning: (1) we integrate optical flow to capture motion dynamics, (2) we explicitly encode timestamp embeddings to provide temporal context, (3) we implement tailored prompt engineering to guide temporal interpretation, and (4) we propose three novel query-aware frame sampling strategies. Our empirical findings show the interventions yield limited and inconsistent gains, highlighting the need for deeper architectural changes in how time is represented, encoded, and processed.

1. Introduction

Video Large Language Models (Vid-LLMs) represent a significant leap in bridging visual content with natural language understanding, demonstrating impressive capabilities in tasks such as video captioning and question answering (QA). Video-LLaVA (Lin et al., 2024), a widely-used open-source Vid-LLM, extends the success of image-language models by integrating a CLIP-based visual encoder (Radford et al., 2021) with a Large Language Model (LLM), processing videos using eight uniformly sampled frames.

Despite these achievements, Vid-LLMs continue to struggle with temporal reasoning—the ability to interpret motion, understand event sequences, and reason about changes over time. Prior work has shown that many Vid-LLMs rely heavily on superficial cues, such as language priors or static visual information, rather than truly engaging with the temporal structure of videos (Huang et al., 2018; Buch et al., 2022; Sevilla-Lara et al., 2019; Lei et al., 2022; Girdhar & Ramanan, 2020). This limitation is particularly evident when evaluated on benchmarks like TempCompass (Liu et al., 2024), which systematically tests temporal understanding across diverse tasks, including event sequencing, speed perception, and action reasoning. Evaluations on TempCompass have exposed significant weaknesses in Video-LLaVA's temporal understanding, indicating that it struggles to grasp the flow of events over time.

In this paper, we aim to diagnose the sources of this temporal reasoning deficit in Video-LLaVA. We hypothesize that the model's shortcomings stem from four interrelated bottlenecks: (1) limited motion sensitivity, (2) ineffective frame sampling strategies that miss critical transitions, (3) limited temporal encoding, and (4) the LLM's tendency to exploit language biases and shortcuts over temporal reasoning.

To investigate these hypotheses, we propose and evaluate four targeted interventions, each designed to address a specific limitation in the Video-LLaVA pipeline: (1) adding optical flow as motion cues, (2) implementing query-aware frame sampling to better represent temporal transitions, (3) introducing explicit timestamp embeddings to encode frame order, and (4) applying temporal-focused prompt engineering to guide the LLM's reasoning. Using the TempCompass benchmark, we conduct a systematic evaluation to isolate the contributions of each component to temporal reasoning failures. While these interventions are theoretically well-motivated, our results reveal that their practical impact is limited and often inconsistent. This suggests that the underlying challenges in temporal understanding are more deeply rooted in the architecture and reasoning patterns of Vid-LLMs. Our analysis provides new insights into the nature of these limitations and offers concrete directions for future work on temporal modeling, bias mitigation, and potential architectural changes in Vid-LLMs.

2. Related work

2.1. Video Large Language Models

Vid-LLMs extend image-language models by integrating temporal video representations with LLMs, enabling tasks such as video captioning, question answering (QA), and cross-modal retrieval (Tang et al., 2025; Maaz et al., 2024; Lin et al., 2024).

[1]University of Amsterdam, Netherlands..

UvA Foundation Models Course 2025, Project Paper Submission.

Developing models for video understanding introduces challenges beyond static image modeling, particularly the need to capture and reason about temporal dynamics. Earlier approaches used convolutional architectures, including simple Convolutional Neural Networks (CNNs) (Karpathy et al., 2014; Ji et al., 2013), two-stream networks (Feichtenhofer et al., 2016), and 3D CNNs (Carreira & Zisserman, 2017). The introduction of Vision Transformers (ViT) (Dosovitskiy et al., 2021) enabled spatiotemporal modeling in architectures like TimeSformer (Bertasius et al., 2021) and MViT (Fan et al., 2021). Self-supervised methods such as VideoBERT (Sun et al., 2019) and VideoMAE (Tong et al., 2022) further improved label efficiency for video modeling. Building on these advances, recent Vid-LLMs such as Video-ChatGPT (Maaz et al., 2024) and Video-LLaVA (Lin et al., 2024) incorporate LLMs through in-context learning, allowing for more flexible and general-purpose reasoning across video-language tasks.

2.2. Temporal Reasoning in Vid-LLMs

Prior research attributes temporal reasoning failures in Vid-LLMs to multiple causes, including both architectural, representational, and task bottlenecks. Some studies specifically identify the LLM component as the bottleneck, suggesting that current Vid-LLMs struggle to process and reason over temporal structures, even when provided with temporally rich visual embeddings (Li et al., 2024b), while others have highlighted how visual features may be inadequately encoded or poorly aligned with temporal semantics (Fateh et al., 2025; Hu et al., 2024). Others have emphasized the inherent complexity of temporal reasoning, which involves both perception of visual change and logical interpretation of event order and causality (Wang et al., 2024).

2.3. Temporal Reasoning Benchmarks

Temporal reasoning in video understanding spans multiple areas, including action recognition, speed perception, direction understanding, attribute change, and event ordering (Liu et al., 2024). However, existing benchmarks tend to overlook the crucial differences among these various temporal aspects (e.g., nuances in action type, speed, and direction). As a result, they do not capture the intricacies required to assess temporal perception. In addition, earlier benchmarks provide means to test reasoning abilities on these temporal aspects (Chen et al., 2024b; Li et al., 2024a), but make use of only a single question type, such as Multi-choice format. This limits their usefulness in evaluating Vid-LLMs that are expected to perform well across a range of temporal aspects and question types.

TempCompass (Liu et al., 2024) addresses these issues by offering a benchmark that systematically evaluates five primary temporal aspects (action, speed, direction, attribute change, and event order) and 10 fine-grained sub-aspects. Furthermore, it contains four question types: Multi-choice QA, Yes/No QA, Caption Matching, and Caption Generation. The benchmark leverages videos from the Shutterstock[1] platform and deliberately constructs conflicting pairs or triplets of videos. In the conflicting sets, while the static content remains consistent, specific temporal aspects vary. It thus tests if Vid-LLMs exploit single-frame bias or pre-existing language knowledge. In total, 7,540 task instructions have been developed through a combination of human-annotated meta-information and LLM-generated content. The evaluation process relies on carefully designed prompts to verify the model's temporal reasoning abilities.

3. Video-LLaVA and Problem Statement

While the limitations of Vid-LLMs' temporal reasoning are increasingly recognized, there remains a lack of systematic investigation into why specific models struggle and how these deficiencies can be addressed. In this work, we focus on Video-LLaVA, an open-source Vid-LLM that combines a CLIP-based visual encoder with an LLM. Its accessibility and transparency make it a strong candidate for probing the architectural and representational causes of temporal reasoning failures.

3.1. Video-LLaVA Architecture and Pipeline

Video-LLaVA extends the image-based LLaVA model to video understanding by sampling frames uniformly from a video. At its core, Video-LLaVA consists of three primary components: the vision encoder, the shared projection layer, and the LLM. Given a video with T frames $\{x_1, x_2, ..., x_T\}$, Video-LLaVA uniformly samples $N = 8$ frames that represent the video content:

$$\mathbf{X}_\mathrm{V} = \{x_{\lfloor i \times T/N \rfloor} | i \in [0, N-1]\}. \tag{1}$$

Each sampled frame is independently processed by a pre-trained OpenCLIP-L/14 (Ilharco et al., 2021) visual encoder from LanguageBind (Zhu et al., 2024), $f_\mathbf{V}$, which produces a sequence of visual tokens $f_\mathbf{V}(\mathbf{X}_\mathrm{V})$. These frame embeddings are then processed by the shared projection layer $f_\mathbf{P}$ that adapts them to the dimensionality expected by the LLM:

$$\mathbf{Z}_\mathrm{V} = f_\mathbf{P}(f_\mathbf{V}(\mathbf{X}_\mathrm{V})). \tag{2}$$

Crucially, this encoding process treats frames independently, without explicitly modeling relationships between them. The projected frame embeddings are presented to the language model together with the embedded textual tokens $\mathbf{Z}_\mathrm{T} = f_\mathbf{T}(\mathbf{X}_\mathrm{T})$. To train the LLM, $f_\mathbf{L}$ (Vicuna-7B v1.5) (Chiang et al., 2023), Video-LLaVA maximizes the likelihood

[1] https://www.shutterstock.com

probability of generating an answer sequence $\mathbf{X}_\mathrm{A}$:

$$p(\mathbf{X}_\mathrm{A}|\mathbf{X}_\mathrm{V}, \mathbf{X}_\mathrm{T}) = \prod_{i=1}^{L} p_\theta\left(\mathbf{X}_\mathrm{A}^{[i]}|\mathbf{Z}_\mathrm{V}, \mathbf{Z}_\mathrm{T}^{[1:i-1]}\right). \quad (3)$$

In this context, L denotes the length of $\mathbf{X}_\mathrm{A}$, and θ represents a parameter that is learned during training.

3.2. Temporal Reasoning Bottlenecks

Although Video-LLaVA demonstrates strong performance on general video understanding benchmarks (Lin et al., 2024), its evaluation on the TempCompass benchmark reveals substantial shortcomings in temporal reasoning (Liu et al., 2024). Based on Video-LLaVA's architecture and the literature, we formulate four primary hypotheses regarding the model's temporal reasoning limitations:

- **Limited Motion Sensitivity**. Video-LLaVA relies on a CLIP-based visual encoder pretrained on static images. As a result, the frame representations may lack motion-related information, which is critical for understanding changes over time. Without explicit modeling of temporal transitions, the encoder may be insensitive to motion cues.

- **Ineffective Frame Sampling**. The default uniform frame sampling used by Video-LLaVA strategy may miss temporally salient segments, especially when key events occur in short time intervals. This can lead to visual inputs that underrepresent the meaningful temporal structure of the video.

- **Limited Temporal Encoding**. Video-LLaVA's visual encoder models temporal structure between frames by injecting learnable position embeddings that encode each frame's order (Zhu et al., 2024). However, this way of temporal encoding is rather limited as it only encodes the order of frames, omitting any notion of absolute timing.

- **Limited Prompt-Based Guidance**. Even when provided with temporally ordered information, the LLM itself may lack the experience necessary to reason over time-based events. This can lead to over-reliance on static visual patterns or language priors instead of actual temporal cues.

In the following sections, we systematically investigate these hypotheses through targeted experiments and interventions. Our objective is to isolate the key contributors to temporal reasoning failures and to evaluate the effectiveness of these interventions.

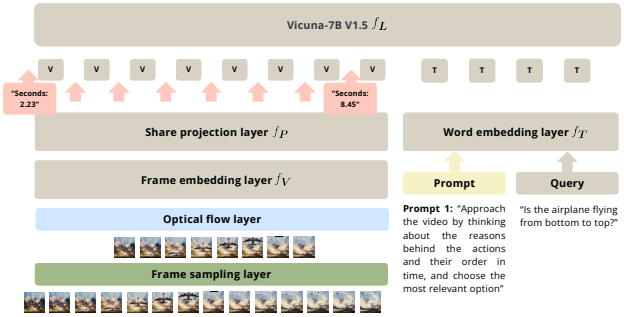

Figure 1. The Video-LLaVA architecture with the four interventions: Optical Flow (Blue), Frame Sampling (Green), Timestamps (Pink), and Prompt Engineering (Yellow). Note that these interventions are applied separately.

4. Methodology

4.1. Experimental Setup

For our experiments, we use the Video-LLaVA model (Lin et al., 2024) described in Section 3.1. All experiments utilize the pre-trained model weights provided by the authors, without architectural modifications unless explicitly noted as an intervention. We use an NVIDIA A100 GPU for all our experiments.

We evaluate on TempCompass, focusing on three question types: Multi-choice QA (1,580 questions), Yes/No QA (2,453 questions), and Caption Matching (1,503 questions). The Caption Generation is excluded due to its dependence on external ChatGPT access. Performance is reported using accuracy for each temporal category, and an average accuracy is calculated across all questions within each question type.

4.2. Interventions

For each of the the hypothesis discussed in Section 3.2, we design a targetted intervention aimed at improving Video-LLaVa.

4.2.1. Limited Motion Sensitivity

To improve motion capturing, we draw inspiration from computer vision techniques that apply visual cues to images (Cai et al., 2024). Specifically, we focus on optical flow to enhance performance on direction-related questions. Until now, optical flow has seen limited integration into Multimodal Large Language Models (MLLMs) due to its high computational cost state-of-the-art methods such as RAFT (Teed & Deng, 2020) are resource-intensive. As a result most models rely on learned video encoders for temporal representations.

Inspired by Cai et al. (2024), where visual prompts,

such as arrows, are overlaid directly onto input images prior to CLIP processing, we apply a similar technique by drawing optical flow arrows as visual prompts on video frames. Specifically, we compute optical flow between consecutive frames for each of the eight uniformly sampled frames in the video. To reduce clutter, we visualize flow vectors at every 40th pixel on a grid, including only those with displacements exceeding 25 pixels in either direction. The arrows are drawn using OpenCV's `arrowedLine` function (Bradski, 2000), with a thickness of 3 for enhanced visibility. An example visualization is provided in Appendix A. To ensure that the LLM incorporates this motion information, we use as prompt the following: *"The arrows show the direction of motion. Please take the direction of these arrows into account when answering the question."*

4.2.2. Ineffective Frame Sampling

We evaluate five frame sampling strategies beyond standard uniform sampling to assess their impact on temporal understanding. For methods 2 through 5, to guide frame selection, we construct a cosine similarity signal between each frame embedding and the query embedding using a CLIP-L/14 encoder. The first method, Randomized Frame Permutation, serves as a non-query-aware baseline. While BOLT is an existing query-aware baseline, the remaining three methods are novel query-aware strategies introduced in this work. These proposed methods account not only for frame relevance and diversity but also explicitly capture temporal transitions by considering changes between adjacent frames. Below, we describe each sampling technique in detail.

1. **Randomized Frame Permutation**. Frames are uniformly sampled, but their order is randomly permuted before being fed into the LLM. This probes the model's reliance on strict chronological order versus unordered frames.

2. **BOLT (Boost Large VLMs without additional Training)** (Liu et al., 2025). This method selects N frames most pertinent to the input query. It calculates cosine similarity scores between the query and all video frames, normalizes these scores to form a probability distribution, and then samples N frames via inverse transform sampling. Following the recommendation by Liu et al. (2025) for a value between 2.5 and 3.5, we set BOLT's single hyperparameter to $\alpha = 3$.

3. **Gradient-based sampling**. This method identifies frames corresponding to the four highest-gradient magnitudes in the smoothed time series of query-frame cosine similarity, where smoothing is applied using a moving average. We ensure the selected gradient points are at least 15 frames apart to maintain relevance. We then sample 2 frames at a 5-frame radius around each of these four gradient points to capture the key transition moments.

4. **Breakpoint Sampling**. This method identifies frames at the boundaries of semantically meaningful scenes by detecting statistically significant change-points in the query-frame similarity signal. A significant change is defined as a shift in the similarity signal that results in a statistically distinct segment, i.e., one whose average similarity differs meaningfully from that of its neighbors. We employ a dynamic programming algorithm (Truong et al., 2020) to find an optimal segmentation by minimizing a global cost function, which is the sum of absolute deviations from the mean within each segment. The four resulting change-points typically correspond to transitions in relevance (e.g., from irrelevant to relevant scenes). To summarize these transitions, we sample the local minima and maxima in similarity around each change-point.

5. **Query-Aware Diversity Sampling (QuADS)**. Lastly, we propose QuADS, a novel frame sampling method that promotes visual diversity by using the Maximum Marginal Relevance (MMR) algorithm (Carbonell & Goldstein, 1998), originally developed for Information Retrieval tasks. It iteratively selects frames from an oversampled candidate pool to optimize the trade-off between relevance to the query and dissimilarity to already selected frames, thereby reducing redundancy. As shown in Equation 4, the trade-off is governed by the hyperparameter λ, which balances relevance (measured as cosine similarity between a frame f_i and the query q) against diversity (cosine similarity between pairs of frames). F denotes the initial set of oversampled frames, while S is the collection of already sampled ones. The initial candidate frames are obtained using inverse transform sampling, following the approach in BOLT (Liu et al., 2025). We set $\lambda = 0.5$ to achieve an equal balance between diversity and relevance, and use an oversampling factor of 3.

$$\text{MMR}(f_i) = \underset{f_i \in F \setminus S}{\text{argmax}} \left[\lambda \cdot \text{sim}(f_i, q) - (1 - \lambda) \cdot \max_{f_j \in S} \text{sim}(f_i, f_j) \right] \tag{4}$$

To highlight the main issue with uniform sampling, we refer to Figure 2, in which we display the frames selected by uniform sampling, and by one of our methods, QuADS sampling. More examples, including sampled frames for all other methods, can be found in Appendix B.

4.2.3. Limited Temporal Encoding

We incorporate explicit temporal order tokens to enhance temporal awareness, relying on two methods. The Frame

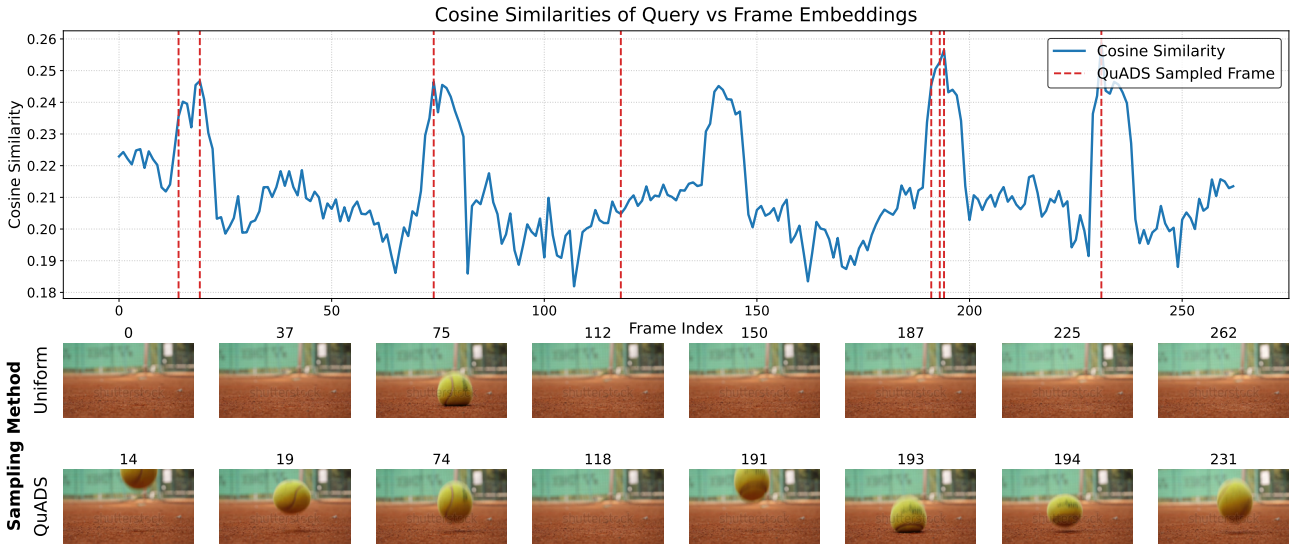

Figure 2. Sampled frames for two different sampling methods (Uniform and QuADS) for the question: *"Which description is a more suitable match for the video? Option 1: The tennis ball is bouncing up and down. Option 2: The tennis ball is spinning clockwise and anticlockwise."*. The uniform sampling approach often fails to provide a sufficient set of informative frames, making it difficult to infer the correct answer.

Order method encodes each frame's relative position within the sequence, mirroring Video-LLaVA's existing positional embeddings, but further reinforcing this ordering signal. The Timestamps method encodes the absolute time in the video (in seconds) for each frame, rather than the order index. This gives the model more information about the interval between frames and the overall video length. We implement these methods as follows:

1. **Frame Order**. We create strings of the form: "Frame i", where $i \in \{1, ..., 8\}$.
2. **Timestamps**. We create strings of the form: "Seconds: t_i" with $i \in \{1, ..., 8\}$ and t_i the time of frame i in seconds, rounded down to 2 decimals, following the implementation of Chen et al. (2024a).

For each method, the timestamp strings are tokenized and embedded using the same text encoder as the query. The resulting embedding is then inserted immediately before the visual tokens of its corresponding frame. For the Timestamps method, the visual input sequence $\mathbf{Z}'_\mathbf{V}$ takes the form:

$$\mathbf{Z}'_\mathbf{V} = \{f_\mathbf{T}(\text{"Seconds: } t_0\text{"}) \| \mathbf{Z}_{\mathbf{V}_0} \| f_\mathbf{T}(\text{"Seconds: } t_1\text{"}) \| \mathbf{Z}_{\mathbf{V}_1} \| \dots\} \tag{5}$$

where $\|$ denotes concatenation, $f_\mathbf{T}$ is the embedding function, and $\mathbf{Z}_{\mathbf{V}_i}$ represents the embedded visual tokens of frame i. For the Frame Order method, "Second t_i" is substituted with "Frame i".

4.2.4. Limited Prompt-Based Guidance

To mitigate the LLM's potential over-reliance on static cues, we implement a prompt engineering strategy for explicit temporal reasoning. This approach was refined iteratively based on observed performance across different task formats.

Initially, building on (Hu et al., 2024), we introduce a base prompt, **Prompt 1:** *"Approach the video by thinking about the reasons behind the actions and their order in time, and choose the most relevant option."* This was applied to all question types, except for Yes/No QA, where the ending *"and choose ..."* is replaced by *"and please answer with yes or no."* This is **Prompt 2**.

We develop additional task-specific prompts drawing on Chain-of-Thought techniques (Wei et al., 2023). For Yes/No QA, we test **Prompt 3:** *"Does the video show this event happening? Answer yes or no, focusing on timing:"*, and **Prompt 4:** *"Analyze the video frame-by-frame for this event, answer yes or no:"*. For Caption Matching, we introduce **Prompt 5:** *"Choose the option that best matches the visual content of the video."*, and **Prompt 6:** *"Consider the beginning, middle, and end of the video. Which caption best summarizes the overall temporal narrative?"*. These prompts aim to direct the LLM towards chronological and causal event processing.

5. Results

Table 1 presents the results of our optical flow and temporal encoding interventions, compared against the baseline performance of vanilla Video-LLaVA. For the evaluation of the various frame sampling techniques, we refer to Table 2. Results for all prompt variations are provided in Table 3.

Question type	Method	Action	Direction	Speed	Order	Attr. Change	Avg
Multi-choice	Vanilla	76.13 ± 0.34	35.52 ± 0.52	37.22 ± 0.63	38.19 ± 0.51	41.20 ± 0.40	45.65 ± 0.25
	Optical Flow	78.30 ± 0.75*	**36.30 ± 1.83**	**38.77 ± 1.14**	40.50 ± 1.93	40.37 ± 1.07	47.37 ± 0.75*
	No Flip	78.90 ± 0.46**	35.50 ± 0.52	38.57 ± 1.44	**42.70 ± 1.00****	40.27 ± 0.91	**47.67 ± 0.55***
	Frame Order	73.20 ± 1.65	32.30 ± 1.35*	38.30 ± 0.46	34.67 ± 0.75**	38.63 ± 1.21	43.83 ± 0.42**
	Timestamps	**79.50 ± 0.62****	36.20 ± 0.62	36.63 ± 1.10	41.63 ± 1.08*	**42.13 ± 0.91**	**47.67 ± 0.29*****
Yes/No	Vanilla	73.59 ± 0.80	51.82 ± 0.61	50.71 ± 0.11	50.09 ± 0.40	51.19 ± 0.13	55.48 ± 0.23
	Optical Flow	73.40 ± 0.36	50.87 ± 0.42	49.77 ± 0.42	**51.63 ± 0.29****	51.50 ± 0.36	56.30 ± 0.20*
	Frame Order	71.73 ± 0.81*	51.53 ± 0.76	49.77 ± 0.42	50.43 ± 1.72	51.20 ± 0.70	55.80 ± 0.26
	Timestamps	**77.50 ± 0.85****	**52.87 ± 0.76**	**51.67 ± 1.23**	51.03 ± 0.46	**52.97 ± 0.31****	**58.30 ± 0.36*****
Caption Matching	Vanilla	**87.88 ± 0.58**	54.13 ± 0.61	**58.65 ± 1.05**	**56.78 ± 2.55**	57.18 ± 0.72	**62.92 ± 0.56**
	Optical Flow	83.03 ± 1.08**	**55.83 ± 2.55**	52.23 ± 2.66*	55.33 ± 1.19	58.23 ± 1.46	60.87 ± 0.75*
	Frame Order	80.27 ± 1.69**	53.60 ± 0.96	54.43 ± 1.63*	52.90 ± 3.27	**59.63 ± 1.62**	60.03 ± 0.93*
	Timestamps	84.40 ± 0.72**	55.47 ± 0.68	54.33 ± 1.23*	54.23 ± 1.08	58.93 ± 1.62	61.37 ± 0.50*

Table 1. Accuracy (mean ± std) across 3 runs for each question type and attribute on the TempCompass benchmark for optical flow and temporal order tokens. Highest value in bold face. No Flip refers to omitting the horizontal flip in the processing video function (`processing_video.py`). Significance of difference compared to the Vanilla model is tested using an independent t-test, where *, **, *** indicate a p-value below 0.05, 0.01, and 0.001, respectively.

Beyond reporting accuracy scores, we analyze the answer distributions for Multi-choice and Yes/No QA in Figures 7 and 8 (Appendix D) for the vanilla Video-LLaVA model. These distributions reveal a pronounced bias: Video-LLaVA tends to overselect answer *A* in Multi-choice QA and *Yes* in Yes/No QA. Further analysis in Table 4 (Appendix C) and Figures 10 and 11 (Appendix E) shows that this bias persists even when answer options are randomly shuffled, suggesting an answer selection bias in the model.

5.1. Lack of Motion Sensitivity

Analysis of Table 1 reveals that incorporating optical flow produces mixed results. While it significantly improves average scores for Multi-choice and Yes/No questions, performance declines significantly for the Caption Matching questions, driven primarily by drops in the Action and Speed subcategories. Surprisingly, adding optical flow did not yield a statistically significant improvement on the Direction attribute for any of the three question types. This is despite Direction being the most intuitively relevant aspect for optical flow, as the direction of motion is explicitly drawn on the video frames.

An additional experiment shed more light on the underlying reasons. We found that the default configuration in the TempCompass benchmark repository horizontally flips half of the videos during processing. This is critical for the Direction questions, with typical questions being, e.g., *Which direction is the person moving?*, and answer options being *right to left* and *left to right*. Flipped frames now show the opposite of the correct answer. While half of the videos are flipped, the corresponding Direction questions are not adapted accordingly, meaning that half of the Direction questions are invalid. However, as is visible in Table 1, there is no significant difference in the Direction questions between the Vanilla set up and the configuration without flipping videos, No Fip (where optical flow is still added).

5.2. Ineffective Frame Sampling

Empirical observations on specific examples suggest that our frame sampling strategies provide more informative frames for answering questions (see Appendix B). Prior work has shown that frame selection alone can enhance performance in long-video understanding without retraining (Liu et al., 2025). However, we observe only limited overall performance differences across the various sampling techniques. Notably, the random permutation method yields significant gains over Vanilla (uniform) sampling for the average of the Yes/No questions, while Vanilla sampling significantly outperforms random permutation in the average of the Caption Matching questions. These inconsistencies suggest that Video-LLaVA is fundamentally unable to perform robust temporal reasoning, even when provided with more relevant or better-distributed frames.

Across the other proposed frame sampling methods, we find that all query-aware sampling methods outperform Vanilla sampling on the Yes/No questions. However, for the other question types, no overall statistically significant differences are observed.

5.3. Limited Temporal Encoding

To assess the impact of temporal tokens, we compare the Frame Order and Timestamps methods against the Vanilla model in Table 1. On average, Frame Order does not perform significantly different on the Yes/No questions, and is significantly worse for the Multi-choice and Caption Matching question types. This suggests adding temporal tokens that encode order is of limited benefit. In contrast, the Timestamps method yield a significant improvement for the average of the Multi-choice and Yes/No questions. More specifically, the Action and Order aspects show a significant improvement for Multi-choice. For Yes/No, the Action and Attribute change show a significant improvement; the gains

Question type	Method	Action	Direction	Speed	Order	Attr. Change	Avg
Multi-choice	Vanilla	76.13 ± 0.34	**35.52 ± 0.52**	**37.22 ± 0.63**	**38.19 ± 0.51**	**41.20 ± 0.40**	**45.65 ± 0.25**
	Randomized frame permutation	76.50 ± 0.17	34.70 ± 0.46	36.50 ± 0.17	38.10 ± 1.30	36.93 ± 0.51***	45.07 ± 0.45
	BOLT ($\alpha = 3$)	76.00 ± 0.52	34.90 ± 0.52	36.17 ± 0.96	37.30 ± 0.35	38.53 ± 0.64**	45.07 ± 0.57
	Breakpoint (Ours)	77.00 ± 0.62	34.10 ± 0.17*	35.83 ± 0.40*	37.73 ± 1.42	39.47 ± 0.91	45.33 ± 0.57
	Gradient-based (Ours)	75.50 ± 0.37	34.60 ± 0.65	36.17 ± 0.66	36.67 ± 0.66*	38.17 ± 0.47**	44.70 ± 0.00**
	QuADS (Ours)	**76.80 ± 0.17**	35.40 ± 0.87	36.67 ± 0.92	37.30 ± 0.17	38.17 ± 0.58**	45.40 ± 0.36
Yes/No	Vanilla	73.59 ± 0.80	**51.82 ± 0.61**	50.71 ± 0.11	50.09 ± 0.40	51.19 ± 0.13	55.48 ± 0.23
	Randomized frame permutation	74.37 ± 0.29	51.53 ± 0.90	**51.07 ± 1.20**	49.53 ± 0.29	50.73 ± 0.47	56.47 ± 0.25**
	BOLT ($\alpha = 3$)	74.13 ± 0.47	51.73 ± 0.31	50.97 ± 0.25	**50.73 ± 0.75**	51.57 ± 1.35	**56.80 ± 0.20****
	Breakpoint (Ours)	**74.93 ± 0.40**	51.40 ± 0.53	50.67 ± 0.55	49.90 ± 0.35	**51.60 ± 0.52**	56.73 ± 0.06 **
	Gradient-based (Ours)	74.20 ± 0.54	51.27 ± 0.25	50.43 ±0.19	50.17 ± 0.34	50.20 ± 0.16**	56.23 ± 0.17*
	QuADS (Ours)	73.53 ± 0.83	51.33 ± 0.31	50.90 ± 0.85	50.10 ± 0.96	51.50 ± 1.65	56.43 ± 0.31*
Caption Matching	Vanilla	**87.88 ± 0.58**	54.13 ± 0.61	58.65 ± 1.05	**56.78 ± 2.55**	57.18 ± 0.72	**62.92 ± 0.56**
	Randomized frame permutation	86.67 ± 1.08	54.13 ± 1.43	59.13 ± 0.91	54.43 ± 2.38	55.00 ± 0.56*	61.73 ± 0.38*
	BOLT ($\alpha = 3$)	87.67 ± 1.29	**55.97 ± 1.10**	**60.37 ± 2.78**	54.53 ± 1.66	56.13 ± 1.29	62.80 ± 0.62
	Breakpoint (Ours)	87.33 ± 0.51	54.50 ± 0.62	**60.37 ± 1.12**	55.33 ± 1.46	56.47 ± 0.74	62.67 ± 0.59
	Gradient-based (Ours)	86.20 ± 0.73*	54.63 ± 0.75	57.87 ± 1.05	54.23 ± 1.84	56.83 ± 0.66	61.83 ± 0.76
	QuADS (Ours)	86.87 ± 0.85	54.77 ± 2.16	59.23 ± 2.36	55.53 ± 0.40	**57.53 ± 2.06**	62.63 ± 0.67

Table 2. Accuracy (mean ± std) across 3 runs for each question type and attribute on the TempCompass benchmark using different frame sampling strategies. Highest value in bold face. Significance of difference compared to the Vanilla model is tested using an independent t-test, where *, **, *** indicate a p-value below 0.05, 0.01, and 0.001, respectively.

Question type	Prompt	Action	Direction	Speed	Order	Attr. Change	Avg
Multi-choice	Vanilla	76.13	35.52	37.22	38.19	41.20	45.65
	Prompt 1	**84.00*****	**37.40***	**38.9***	**38.52**	**46.63****	**49.62*****
Yes/No	Vanilla	**73.59**	**51.82**	**50.71**	50.09	**51.19**	**55.48**
	Prompt 1	66.72	49.89	49.90	49.04	50.07	53.83
	Prompt 2	67.11	50.88	49.72	50.59	49.83	54.34
	Prompt 3	57.30	50.03	49.55	**50.89**	50.03	51.87
	Prompt 4	70.63***	50.72*	50.29*	49.84	51.08	55.30
Caption Matching	Vanilla	**87.88**	54.13	58.65	**56.78**	57.18	62.92
	Prompt 1	83.20	57.82	57.03	49.31	**58.06**	61.05
	Prompt 5	87.54	**59.07****	**58.88**	54.29	56.64	**63.26**
	Prompt 6	74.43	52.92	50.93	25.36	46.55	50.08

Table 3. Accuracy for each question type and attribute on the TempCompass benchmark for different prompts. Results are shown for a single run, except the configurations with the prompt in blue, which were run three times to test significance. These prompts had the highest average accuracy of the non-Vanilla prompts. Highest value in bold face. Significance of difference compared to the Vanilla prompt is tested using an independent t-test, where *, **, *** indicate a p-value below 0.05, 0.01, and 0.001, respectively.

for the Direction and Speed aspects are positive but not statistically significant. There is no meaningful improvement for the Caption Matching questions.

Overall, the Timestamps method, where absolute time is explicitly encoded, helps with tasks requiring temporal reasoning, but can hinder tasks more dependent on static visual-semantic alignment.

5.4. Limited Prompt-Based Guidance

As we show in Table 3, Prompt 1 leads to a significant improvement in average performance of nearly four percentage points. This is driven primarily by a substantial gain in the Action aspect. This indicates that explicitly directing the model to focus on temporal reasoning can be effective in formats requiring reasoning across multiple options.

In contrast, for the Yes/No questions, none of the alternative prompts outperformed the Vanilla prompt. The small declines for all prompts suggest that additional temporal instructions may have introduced unnecessary complexity or ambiguity in simpler binary decision formats.

For Caption Matching, the effect of prompting was mixed. Prompt 5 yields a small (yet insignificant) improvement, while Prompt 6 resulted in a large average performance drop, especially for the Order aspect.

Overall, instructing the model to engage in temporal reasoning via the prompt shows promise for Multi-choice tasks, but its benefits do not generalize to other question types. Our findings suggests that not all forms of temporal prompts are beneficial and the benefit may depend heavily on the structure of the task.

6. Discussion

6.1. Synthesis of Key Findings

Our investigation into Video-LLaVA's temporal reasoning capabilities reveals a nuanced and, at times, contradictory set of behaviors.

First, our interventions achieved no consistent improved

performance of Video-LLava. Adding optical flow as motion cues had a significant positive impact for only two of the three question types. Various frame sampling techniques show only minor differences in overall performance. While incorporating temporal tokens through timestamps improved two question types, it failed to provide a general performance gain. A new prompt for Multi-choice questions, designed to encourage temporal reasoning, yielded the largest average gain (nearly four percentage points) across all question types and intervention, while a narrative-focused prompt harmed Caption Matching performance.

Furthermore, we observed three notable behavioral patterns of the model. A key finding is that we found no effect of horizontally flipping half of the videos on Multi-choice questions, despite questions specifically asking for a direction of motion in the horizontal direction. This suggests that video frames do not play a critical role in the model's answers to the Direction questions, and by extension, to the other questions as well. Another key finding is the model's surprising indifference to the chronological order of frames. Randomly shuffling frames before they were fed to the LLM resulted in a negligible impact on performance and, in some cases, even a marginal improvement. This strongly implies that Video-LLaVA is not processing the video as a coherent temporal sequence. Instead, it appears to treat the input as a "bag of frames", relying on aggregated static cues from individual frames rather than the narrative flow between them. Finally, the underlying LLM (Vicuna-7B v1.5) shows a strong preference for certain responses, such as answering *A* or *Yes* for Multi-choice and Yes/No questions. These biases can obscure the model's actual reasoning abilities. Furthermore, it raises concerns about how accurately current video-language benchmarks reflect true understanding.

Models such as Video-LLaVA represent a significant step forward in bridging the gap between vision and language. However, our findings underscore the substantial challenges that remain before such systems can perform coherent temporal reasoning or approximate human-level understanding of video content.

6.2. Limitations and Future Research

It is essential to acknowledge the limitations of this study, which in turn highlight promising directions for future work. First, our method for integrating motion features via optical flow overlays was rudimentary and may not sufficiently capture complex temporal dynamics. Second, while this work successfully diagnosed significant answer selection biases in the LLM, it did not extend to implementing mitigation strategies; therefore, the performance improvements from our interventions might be masked. Finally, our reliance on manual prompt engineering, guided by human intuition, means the evaluated prompts may be suboptimal compared to those discoverable through automated methods. Based on these limitations, we propose several concrete directions for future research:

- **Addressing Foundational LLM Biases**. A critical next step is to implement and evaluate established debiasing techniques, such as PriDE (Zheng et al., 2024), to disentangle reasoning failures from inherent selection biases and obtain a more accurate measure of the model's temporal reasoning capabilities.
- **Developing Sophisticated Temporal Encoders**. Future work should focus on integrating dedicated temporal modules capable of extracting richer, more abstract motion representations. This could involve better optical flow representations, such as those based on RAFT (Teed & Deng, 2020), and specialized representation encoders inspired by recent work (Koroglu et al., 2024).
- **Evaluating on Advanced Vid-LLM Architectures**. The effectiveness of the interventions explored here should be tested on next-generation Vid-LLMs that feature more advanced mechanisms for fusing visual and linguistic information, moving beyond simple feature concatenation to foster more genuine temporal reasoning. This will determine if our interventions can scale and contribute to models with more sophisticated temporal fusion capabilities.
- **Automating Prompt Engineering**. Systematically exploring automated prompt optimization, for instance by leveraging an auxiliary LLM to generate and refine prompts (Du et al., 2024), could unlock more effective and robust temporal guidance without manual tuning.
- **Broadening Dataset Diversity**. To ensure the generalizability of these findings, investigations should be extended to more diverse datasets, particularly those featuring longer videos and more complex, multi-stage temporal dependencies than those found in TempCompass.

References

Bertasius, G., Wang, H., and Torresani, L. Is space-time attention all you need for video understanding?, 2021. URL https://arxiv.org/abs/2102.05095.

Bradski, G. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

Buch, S., Eyzaguirre, C., Gaidon, A., Wu, J., Fei-Fei, L., and Niebles, J. C. Revisiting the "video" in video-language understanding. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2907–2917,

2022. URL https://api.semanticscholar.org/CorpusID:249375461.

Cai, M., Liu, H., Mustikovela, S. K., Meyer, G. P., Chai, Y., Park, D., and Lee, Y. J. Vip-llava: Making large multimodal models understand arbitrary visual prompts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12914–12923, 2024.

Carbonell, J. G. and Goldstein, J. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 335–336. ACM, 1998. doi: 10.1145/290941.291025. URL https://dl.acm.org/doi/10.1145/290941.291025.

Carreira, J. and Zisserman, A. Quo vadis, action recognition? a new model and the kinetics dataset. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4724–4733, 2017. doi: 10.1109/CVPR.2017.502.

Chen, S., Lan, X., Yuan, Y., Jie, Z., and Ma, L. Timemarker: A versatile video-llm for long and short video understanding with superior temporal localization ability, 2024a. URL https://arxiv.org/abs/2411.18211.

Chen, X., Lin, Y., Zhang, Y., and Huang, W. Autoeval-video: An automatic benchmark for assessing large vision language models in open-ended video question answering, 2024b. URL https://arxiv.org/abs/2311.14906.

Chiang, W.-L., Li, Z., Lin, Z., Sheng, Y., Wu, Z., Zhang, H., Zheng, L., Zhuang, S., Zhuang, Y., Gonzalez, J. E., Stoica, I., and Xing, E. P. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, March 2023. URL https://lmsys.org/blog/2023-03-30-vicuna/.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. An image is worth 16x16 words: Transformers for image recognition at scale, 2021. URL https://arxiv.org/abs/2010.11929.

Du, Y., Sun, W., and Snoek, C. G. M. Ipo: Interpretable prompt optimization for vision-language models, 2024. URL https://arxiv.org/abs/2410.15397.

Fan, H., Xiong, B., Mangalam, K., Li, Y., Yan, Z., Malik, J., and Feichtenhofer, C. Multiscale vision transformers, 2021. URL https://arxiv.org/abs/2104.11227.

Fateh, F. J., Ahmed, U., Khan, H., Zia, M. Z., and Tran, Q.-H. Video llms for temporal reasoning in long videos, 2025. URL https://arxiv.org/abs/2412.02930.

Feichtenhofer, C., Pinz, A., and Zisserman, A. Convolutional two-stream network fusion for video action recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1933–1941, 2016. doi: 10.1109/CVPR.2016.213.

Girdhar, R. and Ramanan, D. Cater: A diagnostic dataset for compositional actions and temporal reasoning, 2020. URL https://arxiv.org/abs/1910.04744.

Hu, Z.-Y., Zhong, Y., Huang, S., Lyu, M. R., and Wang, L. Enhancing temporal modeling of video llms via time gating, 2024. URL https://arxiv.org/abs/2410.05714.

Huang, D.-A., Ramanathan, V., Mahajan, D., Torresani, L., Paluri, M., Fei-Fei, L., and Niebles, J. C. What makes a video a video: Analyzing temporal information in video understanding models and datasets. pp. 7366–7375, 06 2018. doi: 10.1109/CVPR.2018.00769.

Ilharco, G., Wortsman, M., Wightman, R., Gordon, C., Carlini, N., Taori, R., Dave, A., Shankar, V., Namkoong, H., Miller, J., Hajishirzi, H., Farhadi, A., and Schmidt, L. Openclip, July 2021. URL https://doi.org/10.5281/zenodo.5143773.

Ji, S., Xu, W., Yang, M., and Yu, K. 3d convolutional neural networks for human action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35 (1):221–231, 2013. doi: 10.1109/TPAMI.2012.59.

Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., and Fei-Fei, L. Large-scale video classification with convolutional neural networks. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1725–1732, 2014. doi: 10.1109/CVPR.2014.223.

Koroglu, M., Caselles-Dupré, H., Sanmiguel, G. J., and Cord, M. Onlyflow: Optical flow based motion conditioning for video diffusion models. *arXiv preprint arXiv:2411.10501*, 2024.

Lei, J., Berg, T. L., and Bansal, M. Revealing single frame bias for video-and-language learning, 2022. URL https://arxiv.org/abs/2206.03428.

Li, K., Wang, Y., He, Y., Li, Y., Wang, Y., Liu, Y., Wang, Z., Xu, J., Chen, G., Luo, P., Wang, L., and Qiao, Y. Mvbench: A comprehensive multi-modal video understanding benchmark, 2024a. URL https://arxiv.org/abs/2311.17005.

Li, L., Liu, Y., Yao, L., Zhang, P., An, C., Wang, L., Sun, X., Kong, L., and Liu, Q. Temporal reasoning transfer from text to video, 2024b. URL https://arxiv.org/abs/2410.06166.

Lin, B., Ye, Y., Zhu, B., Cui, J., Ning, M., Jin, P., and Yuan, L. Video-llava: Learning united visual representation by alignment before projection, 2024. URL https://arxiv.org/abs/2311.10122.

Liu, S., Zhao, C., Xu, T., and Ghanem, B. BOLT: Boost Large Vision-Language Model Without Training for Long-form Video Understanding, 2025. URL https://arxiv.org/abs/2503.21483.

Liu, Y., Li, S., Liu, Y., Wang, Y., Ren, S., Li, L., Chen, S., Sun, X., and Hou, L. Tempcompass: Do video llms really understand videos?, 2024. URL https://arxiv.org/abs/2403.00476.

Maaz, M., Rasheed, H., Khan, S., and Khan, F. S. Video-chatgpt: Towards detailed video understanding via large vision and language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL 2024)*, 2024.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., and Sutskever, I. Learning transferable visual models from natural language supervision, 2021. URL https://arxiv.org/abs/2103.00020.

Sevilla-Lara, L., Zha, S., Yan, Z., Goswami, V., Feiszli, M., and Torresani, L. Only time can tell: Discovering temporal data for temporal modeling. *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 535–544, 2019. URL https://api.semanticscholar.org/CorpusID:197935119.

Sun, C., Myers, A., Vondrick, C., Murphy, K., and Schmid, C. Videobert: A joint model for video and language representation learning, 2019. URL https://arxiv.org/abs/1904.01766.

Tang, Y., Bi, J., Xu, S., Song, L., Liang, S., Wang, T., Zhang, D., An, J., Lin, J., Zhu, R., Vosoughi, A., Huang, C., Zhang, Z., Liu, P., Feng, M., Zheng, F., Zhang, J., Luo, P., Luo, J., and Xu, C. Video understanding with large language models: A survey. *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 1–1, 2025. doi: 10.1109/TCSVT.2025.3566695.

Teed, Z. and Deng, J. Raft: Recurrent all-pairs field transforms for optical flow. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pp. 402–419. Springer, 2020.

Tong, Z., Song, Y., Wang, J., and Wang, L. Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training, 2022. URL https://arxiv.org/abs/2203.12602.

Truong, C., Oudre, L., and Vayatis, N. Selective review of offline change point detection methods. *Signal Processing*, 167:107299, February 2020. ISSN 0165-1684. doi: 10.1016/j.sigpro.2019.107299. URL http://dx.doi.org/10.1016/j.sigpro.2019.107299.

Wang, H., Xu, Z., Cheng, Y., Diao, S., Zhou, Y., Cao, Y., Wang, Q., Ge, W., and Huang, L. Grounded-videollm: Sharpening fine-grained temporal grounding in video large language models. *arXiv preprint arXiv:2410.03290*, 2024.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E., Le, Q., and Zhou, D. Chain-of-thought prompting elicits reasoning in large language models, 2023. URL https://arxiv.org/abs/2201.11903.

Zheng, C., Zhou, H., Meng, F., Zhou, J., and Huang, M. Large language models are not robust multiple choice selectors, 2024. URL https://arxiv.org/abs/2309.03882.

Zhu, B., Lin, B., Ning, M., Yan, Y., Cui, J., Wang, H., Pang, Y., Jiang, W., Zhang, J., Li, Z., Zhang, W., Li, Z., Liu, W., and Yuan, L. Languagebind: Extending video-language pretraining to n-modality by language-based semantic alignment, 2024. URL https://arxiv.org/abs/2310.01852.

A. Optical Flow Example

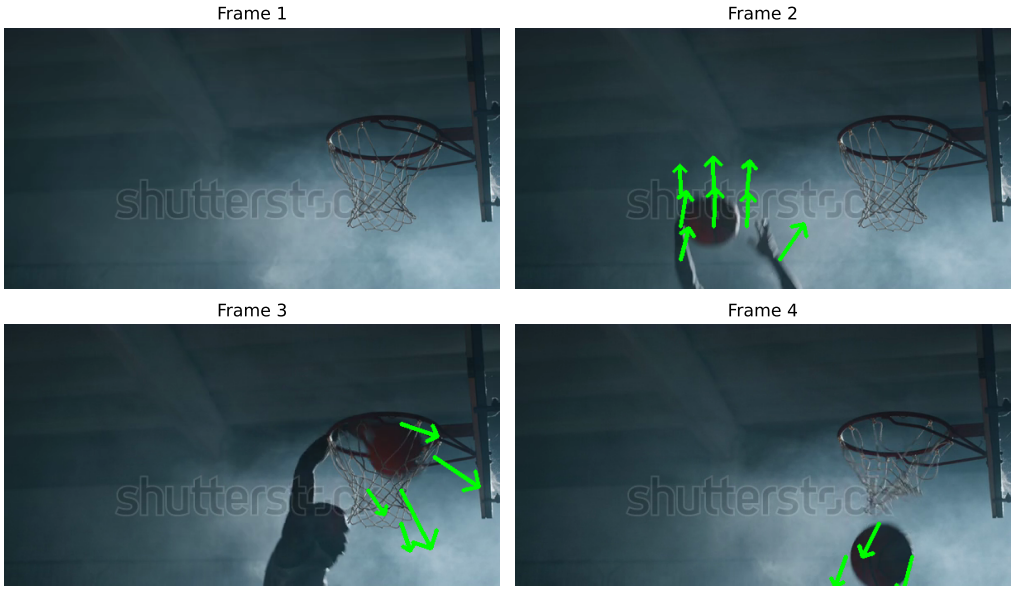

Figure 3. Optical flow between video frames for the first four frames out of the eight uniformly sampled frames from a video in the TempCompass dataset.

B. Frame Sampling Examples

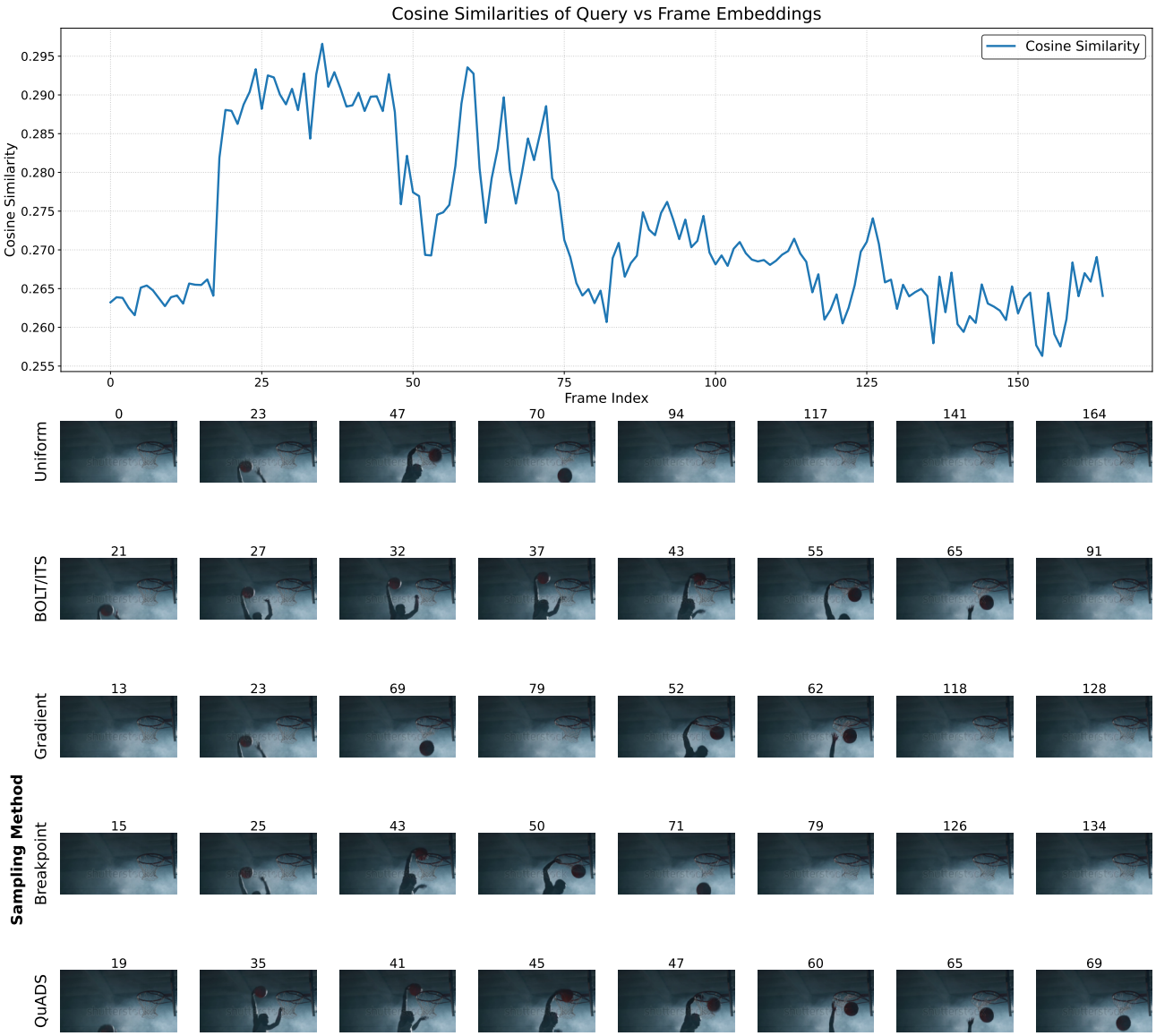

Figure 4. Frames sampled with several different sampling methods on a video from the TempCompass benchmark. Prompt: "What is the man doing in the video? A. dunking a basketball B. dribbling a basketball C. passing a basketball."

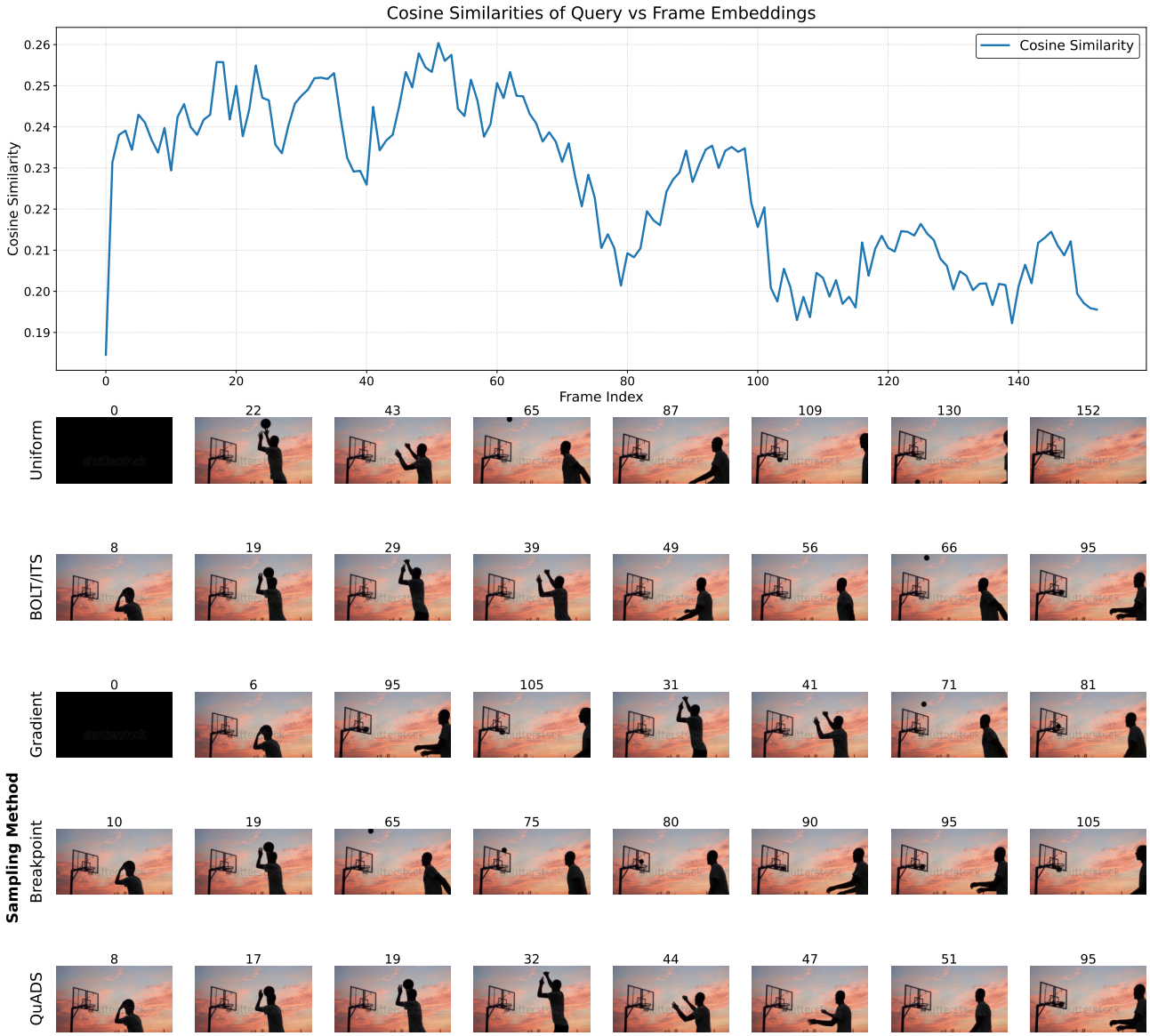

Figure 5. Frames sampled with several different sampling methods on a video from the TempCompass benchmark. Prompt: "What is the man doing in the video? A. dribbling basketball B. passing basketball C. shooting basketball D. dunking basketball."

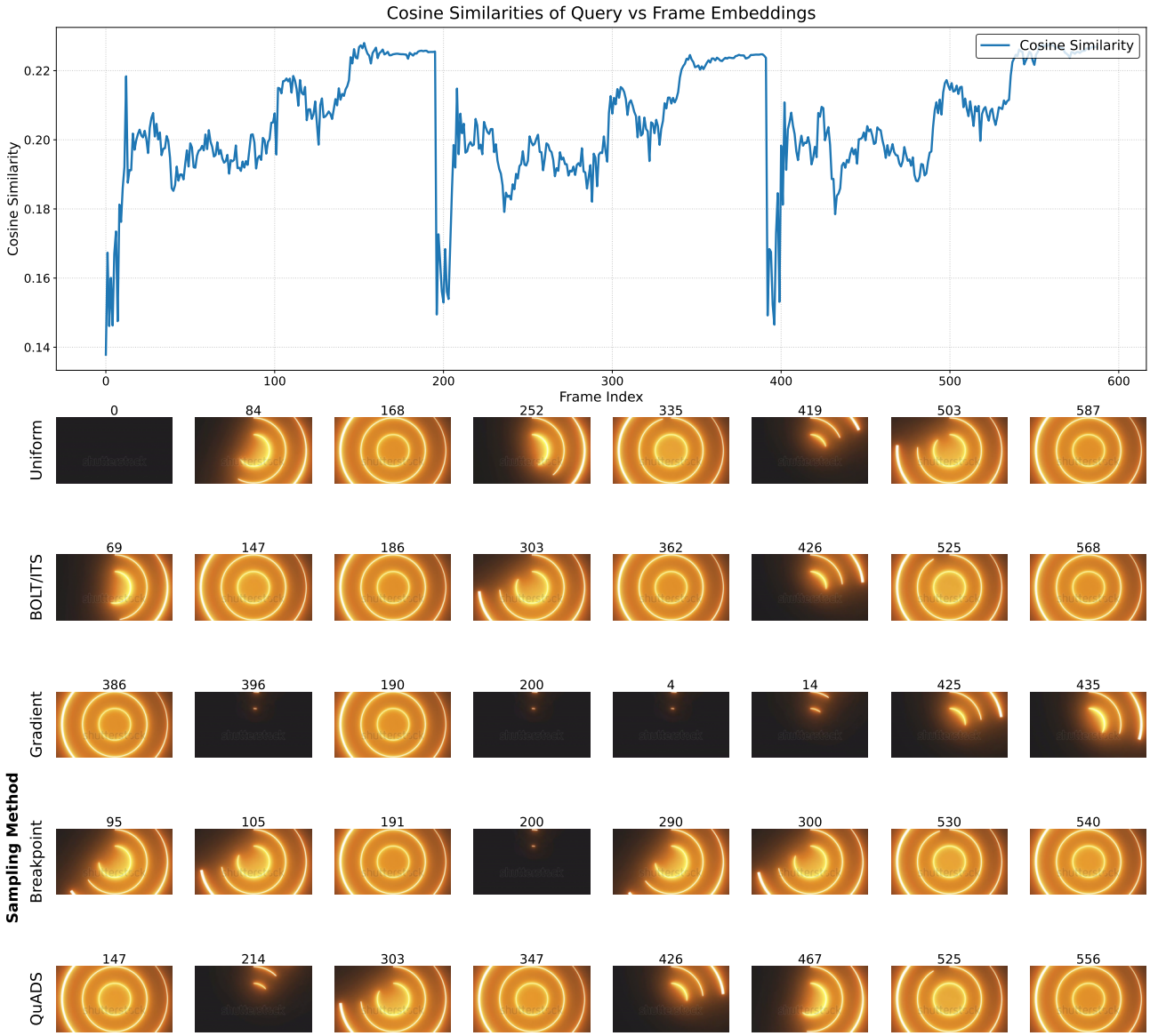

Figure 6. Frames sampled with several different sampling methods on a video from the TempCompass benchmark. Prompt: ”Which caption matches the video better? Caption A: The light is growing clockwise. Caption B: The light is growing downwards.”

C. Results on Shuffled Answers

Question type	Method	Action	Direction	Speed	Order	Attr. Change	Avg
Multi-choice	Vanilla	76.13 ±0.34	35.52 ±0.52	37.22 ±0.63	38.19 ±0.51	41.20 ±0.40	45.65 ±0.25
	Shuffled	76.70 ±0.35**	35.70 ±0.62	40.40 ±0.30**	38.50 ±0.35	39.00 ±0.17	46.57 ±0.06 **
Caption Matching	Vanilla	87.88 ±0.58	54.13 ±0.61	58.65 ±1.05	56.78 ±2.55	57.18 ±0.72	62.92 ±0.56
	Shuffled	88.43 ±0.81*	54.83 ±0.81*	59.20 ±1.39	57.67 ±1.82	55.20 ±1.40	62.93 ±0.64

Table 4. Accuracy (mean ± std) across 3 runs for the Multi-choice and Caption Matching questions. The results are shown for the original and shuffled answer order. Significance of difference between the Vanilla and Shuffled configuration is tested using an independent t-test, where * and ** indicate a p-value below 0.05 and 0.01, respectively.

D. Answer Distribution

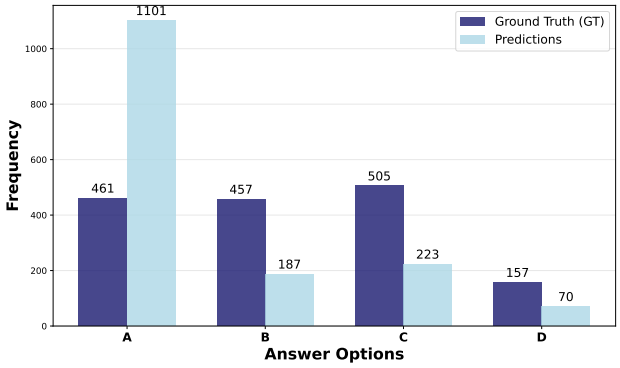

Figure 7. Answer distribution by Video-LLaVA on TempCompass MC QA. The model favors answer *A* over later options. Results are an average over 3 runs.

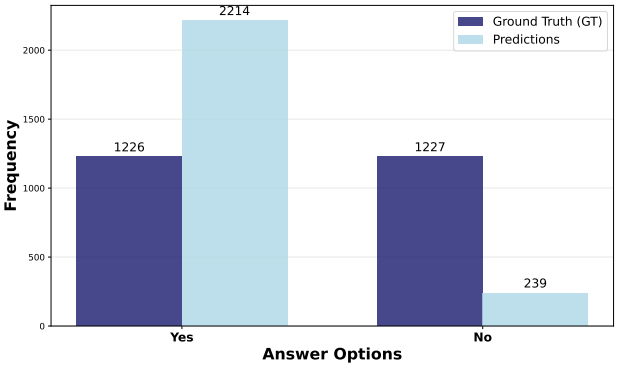

Figure 8. Answer distribution by Video-LLaVA on TempCompass Yes/No QA. The model favors *Yes* over *No* despite the uniform ground-truth distribution. Results are an average over 3 runs.

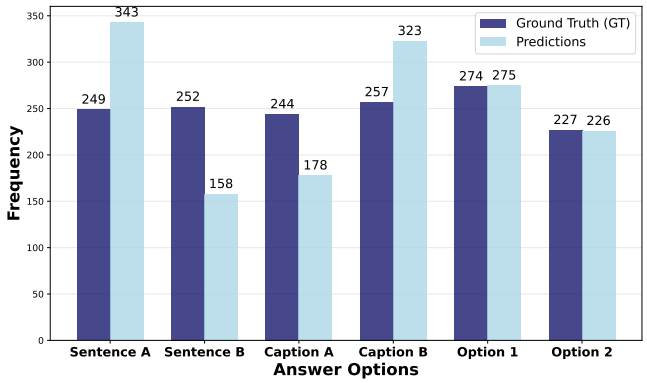

Figure 9. Answer distribution by Video-LLaVA on TempCompass Caption Matching QA. The model lightly favors *Sentence A* and *Caption B* despite the uniform ground-truth distribution. Results are an average over 3 runs.

E. Answer Distribution on Shuffled Answers

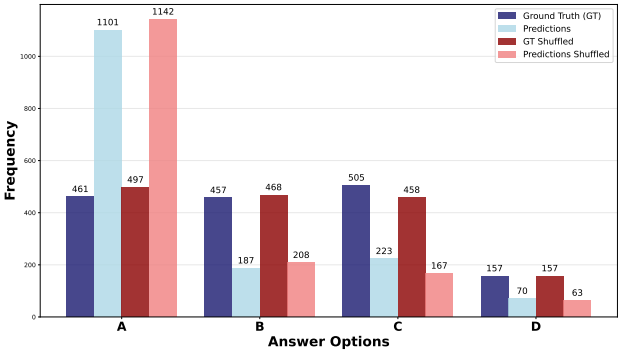

Figure 10. Answer distribution for Video-LLaVA on TempCompass MC QA with original and shuffled answer orders. The answer distribution remains nearly identical despite answer shuffling, indicating potential position bias. Results are an average over 3 runs.

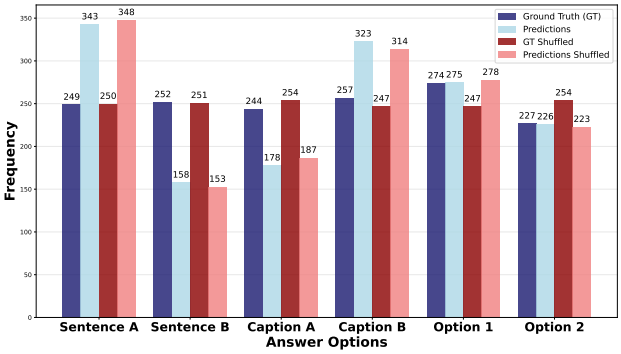

Figure 11. Answer distribution for Video-LLaVA on TempCompass caption-matching questions with original and shuffled answer orders. The answer distribution remains nearly identical despite answer shuffling, indicating potential position bias. Results are an average over 3 runs.