DynFocus: Dynamic Cooperative Network Empowers LLMs with Video Understanding

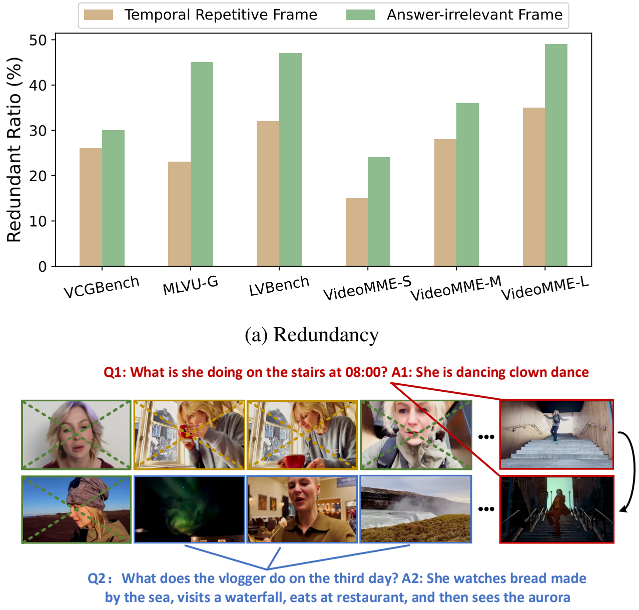
Yudong Han[1◊], Qingpei Guo[2†], Liyuan Pan[1†], Liu Liu[3], Yu Guan[4], Ming Yang[2]
[1]Beijing Institute of Technology, [2]Ant Group,
[3]KooMap Dept., Huawei, [4]University of Warwick
hanyudong.sdu@gmail.com, qingpei.gqp@antgroup.com, liyuan.pan@bit.edu.cn
liuliu33@huawei.com, Yu.Guan@warwick.ac.uk, m.yang@antgroup.com

Abstract

The challenge in LLM-based video understanding lies in preserving visual and semantic information in long videos while maintaining a memory-affordable token count. However, redundancy and correspondence in videos have hindered the performance potential of existing methods. Through statistical learning on current datasets, we observe that redundancy occurs in both repeated and answer-irrelevant frames, and the corresponding frames vary with different questions. This suggests the possibility of adopting dynamic encoding to balance detailed video information preservation with token budget reduction. To this end, we propose a dynamic cooperative network, DynFocus, for memory-efficient video encoding in this paper. Specifically, i) a Dynamic Event Prototype Estimation (DPE) module to dynamically select meaningful frames for question answering; (ii) a Compact Cooperative Encoding (CCE) module that encodes meaningful frames with detailed visual appearance and the remaining frames with sketchy perception separately. We evaluate our method on five publicly available benchmarks, and experimental results consistently demonstrate that our method achieves competitive performance. Code is available at https://github.com/Simon98-AI/DynFocus

1. Introduction

Large Language Models (LLMs) have shown their ability on general AI [24]. Vision Language Models (VLMs) extend the capabilities of LLMs to process visual data, demonstrating proficiency in tasks such as image captioning and visual question answering. However, challenges arise in video understanding, especially with long-term videos, where representing consecutive video frames requires an excessive number of tokens, leading to high memory usage.



(a) Redundancy

(b) Correspondence

Figure 1. Concept of redundancy and correspondence in our pipeline. (a) The proportion of redundancy for video datasets[1]. Redundancy includes both repeated and answer-irrelevant frames. Repeatance gauges the redundancy between consecutive frames, while answer-irrelevance refers to frames with a marginal contribution to question answering. (b) An example of correspondence. Given a video, we highlight the corresponding question/answer pairs and frames using red and blue boxes, respectively.

Recent attempts use average pooling, attention, or dynamic masking to reduce video tokens spatially [23, 30, 37, 39, 58]. However, redundant frames lead to the neglect of key visual details. Several works [17, 57] capture visual appearance with memory banks to preserve key details. However, the key details vary in correspondence to questions, which can easily result in the loss of keyframes from long

[1] The implementation is provided in supplementary materials.

† Corresponding authors. ◊ Work done during internship at Ant Group.

videos and increase the overhead of the memory banks.

In Fig. 1, we illustrate examples of redundancy and correspondence in videos. We observe that i) there is significant redundancy among frames, with only a few meaningful frames directly contributing to question answering. This suggests the potential for adopting a dynamic frame encoding strategy to reduce tokens based on their contribution. ii) Answering different questions generally requires focusing on different parts of the frame. Therefore, dynamically identifying meaningful frames offers better flexibility for sophisticated video content understanding.

In this paper, we propose a dynamic cooperative framework, *DynFocus*, for memory-efficient video encoding. Specifically, it consists of two key modules: Dynamic Event Prototype Estimation (DPE) and Compact Cooperative Encoding (CCE). DPE serves as the dynamic selector to accurately discern the meaningful frames, which takes both redundancy and correspondence to question answering into consideration as the selection standard. Afterwards, CCE complies with the dynamic encoding principle. The meaningful frames are encoded with fine-grained context features for detailed visual appearance, whereas those redundant frames are substantially encapsulated into a few tokens for sketchy perception, which enables LLM to capture broader temporal clues within a fixed receptive field. These two modules reconcile the nuanced visual-semantic understanding with affordable token quota.

Moreover, our CCE module draws inspiration from the cooperation of retinal ganglion cells in the primate visual system. Biological studies [20, 47] found that in these cells, *Rod* cells perceive the overall scene in a wide field of view, while *Cone* cells understand complex scenes with fine details. These cells are located at the periphery of the retina and arranged in a parallel manner, receiving the signals but activated under different conditions. Our framework is analogous in two aspects: (1) Which cell is activated depends on whether the current input frame is meaningful or not. (2) The meaningful frames are encoded with fine-grained tokens as key detailed clues, akin to *Cones*, whereas the marginal frames are condensed into low-resolution tokens, ensuring better temporal consistency, similar to *Rods*. We hope these relations will further support the design philosophy of our method and reveal its rationality.

Our contributions are summarized as follows:

- We propose a dynamic cooperative network, *DynFocus*, towards memory-efficient video encoding within LLM, inspired by the biological concept of *Cone* and *Rod* cells.
- We introduce two modules, DPE and CCE, that dynamically balance subtle visual appearance with sketchy temporal perception using affordable tokens.
- Experimentally, we achieve the competitive even SOTA performance on two publicly mainstream short video benchmarks, three long video benchmarks, and one diagnosis benchmark on video hallucination.

2. Related Work

Video-based Large Language Models. In recent years, Vision Language Models (VLMs) has emerged to extend the capabilities of LLMs [9, 10, 42, 44, 54] to handle diverse and complicated inputs with satisfactory generalization. Generally, VLMs incorporate additional connector to bridge the semantic gap between input video content and LLMs, further performing modality alignment and instruction tuning on video-based dataset. However, video understanding presents the significant challenges due to their extensive memory overhead. Several studies have dedicated to addressing these challenges with greater efficiency. Video-ChatGPT [39] adopts both spatial and temporal pooling to condense video tokens. VideoChat [27] employs a learnable Q-former [11] to aggregate the similar tokens for memory reduction. Chat-UniVi [23] develops a unfied framework for processing both image and video, which reduces spatial and temporal tokens through multi-stage token merging. Although these methods alleviate the memory usage to some extent, they often discard the abundant temporal clues by sampling parts of frames as the input. To compensate the loss of temporal clues, LLaMA-VID [30] innovates with a dual-token approach that represents each frame with context and content tokens, which allows for larger video throughput. MovieChat [52] incorporates the short-term memory and long-term memory into unified framework, strategically combining similar frames to reduce memory footprint while capture the temporal clues. Similarly, MA-LLM [18] stores past video information in a memory bank, which allows to reference historical video content for long-term analysis without exceeding memory limits. However, these methods exhibit proficiency in capturing temporal clues at the expense of discarding the visual details. In a nutshell, they struggle to jointly capture the spatial details and temporal dynamics effectively.

Dynamic Networks. Dynamic networks, adjusting the encoding strategy according to specific input, have recently garnered the burgeoning interest across various domains. Early methods mainly focus on traditional image classification by channel pruning or layer skipping. For example, BlockDrop [59] designed an auxiliary policy network to determine whether skip or execute convolutional blocks via reinforcement optimization. Based on dynamic mechanism, a series of research efforts are devoted to better adapting to the various dynamic scenes. Specifically, Dynamic [29] proposes a routing network with soft conditional gate to adaptively search data-dependent scale transformation paths for semantic segmentation. In the field of image question answering, SUPER [16] develops a semantic-aware modular routing framework to recursively handle different complexity of visual scene. In this work, we marks the first attempt to reveal the substantial potential of dynamic encoding strategies when understanding the complicated long-term video.

3. Dynamic Cooperative Network

As shown in Fig 2, the overall framework is comprised of three parts. **Part 1:** visual and text encoder are adopted to produce the corresponding features; **Part 2:** the proposed dynamic cooperative network serves as the connector to compress the video content for LLM, which consists of two modules, DPE module and CCE module; **Part 3:** the foundational LLM receives the token sequence outputted from Part 2 to generate the language response.

3.1. Visual and Text Encoder

Given a T-frame video, we extract the frame-wise features $\mathbf{V} = \{\mathbf{v}_t\}_{t=1}^{T}$, using a pre-trained visual encoder, where $\mathbf{v}_t \in \mathbb{R}^{N \times d}$ denotes the feature of the t-th frame. Here, N is the number of image patches and d is the feature dimension. For the text encoder, user instruction is fed to the pre-trained text encoder to generate the text features $\mathbf{Q} = \{\mathbf{q}_r\}_{r=1}^{R}$, where $\mathbf{q}_r \in \mathbb{R}^{d'}$ denotes the feature embedding of each token in user instruction. d' is the feature dimension and R is the total number of tokens.

3.2. Dynamic Event Prototype Estimation

Given frame-wise features $\mathbf{V}$, we aim to estimate event prototypes, *i.e*., discriminative features that are most relevant to ground-truth answer.

For each frame feature $\mathbf{v}_t$, we first perform local average pooling spatially. The pooling process reduces the number of features from N to P for efficiency, resulting in $\mathbf{f}_t = \mathrm{Pool}(\mathbf{v}_t), \mathbf{f}_t \in \mathbb{R}^{P \times d}$. Redundancy exists in T-frame features $\mathbf{F} = \{\mathbf{f}_t\}_{t=1}^{T}$. To remove redundancy and identify representative features from $\mathbf{F}$, we perform clustering on $\mathbf{F}$, and use cluster centers to estimate event prototypes.

Following [23], we borrow components from the traditional DPC-KNN [13] algorithm for obtaining cluster centers. For self-contain purposes, we briefly summarize two important variables (local density ρ_t and distance indicator δ_t) in DPC-KNN below.

For clustering, the local density ρ_t measures the mean distance to C nearest neighbors of the t-th frame, and is given by,

$$\rho_t = \exp\left(-\frac{1}{C}\sum_{t' \in \mathcal{N}(t)} \frac{1}{P}\|\mathbf{f}_t - \mathbf{f}_{t'}\|_F^2\right), \tag{1}$$

where $t' \in \mathcal{N}(t)$ denotes that $\mathbf{f}_{t'}$ is in the neighborhood of $\mathbf{f}_t$. $\|\cdot\|_F$ denotes the Frobenius norm, and it is also used to perform C nearest neighbor search for each $\mathbf{f}_t$.

The distance indicator δ_t measures the possibility of t-th frame to be cluster center by calculating the minimum distance between $\mathbf{f}_t$ and any other frames with higher density, and is given by,

$$\delta_t = \begin{cases} \min\limits_{t'} \frac{1}{P}\left\|\mathbf{f}_t - \mathbf{f}_{t'}\right\|_F^2, & \text{if } \exists t' \text{ s.t. } \rho_{t'} > \rho_t \\ \max\limits_{t'} \frac{1}{P}\left\|\mathbf{f}_t - \mathbf{f}_{t'}\right\|_F^2, & \text{otherwise}. \end{cases} \tag{2}$$

We use the product of ρ_t and δ_t to measure the importance of each frame. Frames with high scores are more likely to be informative. We first sort scores $\{\rho_t \times \delta_t\}_{t=1}^{T}$ in the decreasing order and then take the Top-L frame features from $\mathbf{F}$ with high scores as representative features. Finally, we normalize representative features using their importance scores and estimate event prototypes as follows,

$$\mathbf{m}_l = \frac{\sum_{t' \in \mathcal{N}(l)} \exp(\rho_{t'} \cdot \delta_{t'})\mathbf{f}_{t'}}{\sum_{t' \in \mathcal{N}(l)} \exp(\rho_{t'} \cdot \delta_{t'})}, l \in [1, L], \tag{3}$$

where $\exp(\rho_{t'} \cdot \delta_{t'})$ denotes the importance weight of $\mathbf{f}_{t'}$.

Note that the obtained event prototypes $\mathbf{M} = \{\mathbf{m}_l\}_{l=1}^{L}$ from Eq. (3) are estimated only based on frame features, and are not aligned to the ground-truth answer. In other words, $\mathbf{m}_l \in \mathbb{R}^{P \times d}$ may contain visual redundancy, which would disturb useful clues for sophisticated video understanding. In light of this, we aim to further select answer-relevant event prototypes from $\mathbf{M}$. Specifically, we use a multi-layer perceptron (MLP) network $\mathcal{U}(\cdot)$ to regress frame-wise scores. Note that $\mathcal{U}(\cdot)$ is learned end-to-end with the supervision from LLM, thus is aligned to ground-truth answer implicitly. The regression is given by,

$$s_l = \mathcal{U}\left(\mathrm{Max}(\mathbf{m}_l) || \mathrm{Avg}(\mathbf{m}_l)\right), \tag{4}$$

where $\mathrm{Max}(\mathbf{m}_l) \in \mathbb{R}^d$ denotes the row feature with maximum feature norm (L_2) across P rows. $\mathrm{Avg}(\mathbf{m}_l) \in \mathbb{R}^d$ denotes the averaged row feature across P rows. $\cdot||\cdot$ denotes concatenation. After collecting all scores into a score vector $\mathbf{s} = \{\mathbf{s}_l\}_{l=1}^{L}$, we perform min-max normalization to normalize score values to be within $[0, 1]$. Finally, we sort scores in the decreasing order and take the Top-K event prototypes from $\mathbf{M}$ with high scores, indicated by the index vector $\mathbf{p}$, as filtered event prototypes. The filtered event prototypes and the index vector are denoted as $\mathbf{H} = \{\mathbf{h}_k\}_{k=1}^{K}$ and $\mathbf{p} = \mathrm{topk}(\mathbf{s}) \in \mathbb{N}^K$, respectively.

We also retrieve indices of filtered event prototypes in the T-frame video, obtaining the binary index mask $\mathbf{b} = \{b_t\}_{t=1}^{T}$. Notably, $b_t = 1$ indicates important frame, while $b_t = 0$ signifies redundant frame.

Training $\mathcal{U}(\cdot)$. Note that the Top-K operation is not differentiable and thus stops the gradient propagation from LLM to update our score net $\mathcal{U}(\cdot)$. This limitation restricts our $\mathcal{U}(\cdot)$ to dynamically estimate the redundancy and flexibly capture correspondence without auxiliary loss supervision. To address this issue, we transform the Top-K operation into

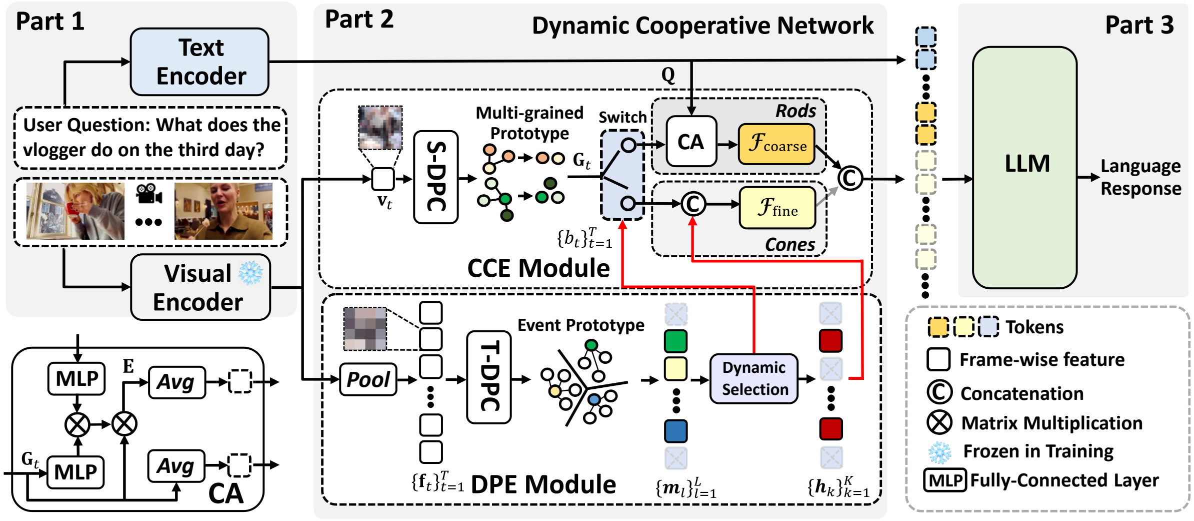


Figure 2. Schematic Illustration of *DynFocus*. Our method takes the user instruction and video frames as input, and yields the compact video tokens from CCE module for LLM. Specifically, DPE module serves as the selector to identify the prototypes that contribute greatly to answer, providing CCE module with event prototype $\{\mathbf{h}_k\}_{k=1}^K$ and the binary mask $\{b_t\}_{t=1}^T$, which is marked with two red arrows. Benefited from this, CCE module dynamically encode the critical prototypes with more tokens, and encapsulate the marginal prototypes with few tokens. T-DPC and S-DPC represent the DPC-KNN clustering temporally and spatially, respectively.

solving a linear programming problem to make our network end-to-end trainable. Specifically, we convert the index vector $\mathbf{p} = [p_1, ..., p_K]$ into a stack of L one-hot vector with K elements, denoted as $\mathbf{P} = [\mathbf{I}_{p_1}, ..., \mathbf{I}_{p_K}] \in \{0,1\}^{L\times K}$. Here, $\mathbf{I}_{p_1}$ denotes the one-hot vector where only the p_1-th element is set to 1. As a result, the filtered event prototypes with top-K scores can be summarized as $\mathbf{H} = \mathbf{P}^\top\mathbf{M}$. Afterwards, we resort to the perturbed maximum method [5] to construct a differentiable operator. In theory, selecting top-K prototypes via subspace projection matrix $\mathbf{P}$ equals to solving a linear programming problem ,

$$\text{argmax}_{\mathbf{P}\in\mathcal{C}} \left\langle \mathbf{P}, \mathbf{s}\mathbf{1}^\top \right\rangle, \tag{5}$$

where $\mathbf{s}\mathbf{1}^\top \in \mathbb{R}^{K\times L}$ denotes the score vector s replicated L times, $\langle\rangle$ denotes the flatten operation followed by dot product. $\mathcal{C}$ is the orthodox convex polytope constrain set $\mathcal{C} = \{\mathbf{P} \in \mathbb{R}^{K\times L} : \mathbf{P}_{k,l} \geq 0, \mathbf{1}^\top\mathbf{P} = \mathbf{1}\}$. We follow [5] to perform forward and backward operations to solve $\mathbf{P}$. Specifically, solving Eqn. 5 could be achieved by taking the expectation of random perturbations,

$$\mathbf{P}_\sigma = \mathbb{E}_P\left[\text{argmax}_{\mathbf{P}\in\mathcal{C}} \left\langle \mathbf{P}, \mathbf{s}\mathbf{1}^\top + \sigma\mathbf{Z} \right\rangle\right], \tag{6}$$

where $\mathbf{Z}$ is a perturbed vector sampled from the uniform Gaussian distribution and σ serves as the hyper-parameter to control the variance of perturbation. Following [1], the Jacobian associated with Eqn. 6 can be simplified as,

$$\frac{\partial \mathbf{P}_\sigma}{\partial \mathbf{s}} = \mathbb{E}_P\left[\text{argmax}_{\mathbf{P}\in\mathcal{C}} \left\langle \mathbf{P}, \mathbf{s}\mathbf{1}^\top + \sigma\mathbf{Z} \right\rangle \mathbf{Z}/\sigma\right], \tag{7}$$

By means of Eqn. 7, the gradient from autoregressive loss in LLM would update the distribution of representation $\mathbf{H}$ and matrix $\mathbf{P}$, thereby updating our score network $\mathcal{U}(\cdot)$ via $\frac{\partial \mathbf{P}_\sigma}{\partial \mathbf{s}}$ according to the chain rule. As a result, our DPE module can be trained together with LLM in an end-to-end fashion, which effectively mitigates the answer-irrelevant visual nuisance in video while achieving dynamic selection in accordance with answer and question.

3.3. Compact Cooperative Encoding

Given frame-wise features $\mathbf{V} = \{\mathbf{v}_t\}_{t=1}^T$, event prototypes $\mathbf{H} = \{\mathbf{h}_k\}_{k=1}^K$, and their corresponding index mask $\mathbf{b} = \{b_t\}_{t=1}^T$, we perform cooperative encoding for memory-efficient video understanding. Frames that contribute greatly to answer (*i.e.*, $b_t = 1$) will be encoded with more tokens than marginal frames (*i.e.*, $b_t = 0$), to capture intricate spatial details.

For each frame feature $\mathbf{v}_t$, we use the same clustering pipeline (Eq. (1),(2),(3)) in Sec. 3.2 to estimate spatial object prototypes $\mathbf{Z}_t = \{\mathbf{z}_{t,i}\}_{i=1}^I$. The only difference is that spatial clustering is performed on N patch features within single frame feature $\mathbf{v}_t$, aggregating N patch features into I prototypes. In contrast, temporal clustering in Sec. 3.2 is performed on T frame features in the whole video.

We find that semantic abstraction of $\mathbf{v}_t$ can be achieved when spatial clustering is performed multiple times. For example, concepts such as "person" and "dog" are progressively formed from the low-level attribute or color information. We thus use multiple clustering layers to capture more abundant visual details. The output spatial object prototypes of layer j is fed to layer $j+1$ for abstraction, which means that number of prototypes participating in subsequent layers reduce progressively.

Collecting all layer outputs results in our multi-grained spatial object prototypes $\mathbf{G}_t = \{\mathbf{Z}_t^{(j)}\}_{j=1}^{J}$. Here, J is the total number of clustering layers.

***Cones* Encoding.** We mimic cones to focus on fine visual appearance. Specifically, frames with $b_t = 1$ are encoded with the combination of event and its corresponding multi-grained spatial prototypes,

$$\mathbf{U}_{t,b_t=1} = \mathcal{F}_{\text{fine}}\left(\mathbf{h}_t || \mathbf{G}_t\right), \tag{8}$$

where $\mathcal{F}_{\text{fine}}$ is a simple MLP network. Note that no feature pooling operation is performed to capture delicate details, and the number of tokens in $\mathbf{U}_{t,b_t=1}$ equals the number of summation of event and multi-grained spatial prototypes.

***Rods* Encoding.** We mimic rods to focus on coarse temporal dynamics towards broader video understanding. Specifically, frames with $b_t = 0$ are encoded with the modulation of text embedding $\mathbf{Q}$, to obtain text-grounded visual clues,

$$\mathbf{E} = \text{Softmax}\left(\frac{f_q(\mathbf{G}_t)(f_k(\mathbf{Q}))^\top}{\sqrt{d}}\right)\mathbf{G}_t, \tag{9}$$

where $f_q(\cdot)$ and $f_k(\cdot)$ represent the linear projection, which map the spatial object prototypes and textual embedding into query and key, respectively.

To facilitate the memory-efficient video understanding, we condense $\mathbf{E}$ to a single token using average pooling. Combining global content token, we extract compact embedding,

$$\mathbf{U}_{t,b_t=0} = \mathcal{F}_{\text{coarse}}\left(\text{Avg}(\mathbf{E}) || \text{Avg}(\mathbf{G}_t)\right), \tag{10}$$

where $\mathcal{F}_{\text{coarse}}$ is a simple MLP network, and $\mathbf{U}_{t,b_t=0} \in \mathbb{R}^{2d'}$. Due to that $\mathbf{U}_{t,b_t=0}$ only has two tokens, it enables smooth temporal transition and improves the scene consistency for consecutive frames.

Cooperative Encoding. Given embeddings $\mathbf{U}_{t,b_t=1}$ and $\mathbf{U}_{t,b_t=0}$ from *Cones* and *Rods*, respectively, we combine them in a token-wise manner to obtain the dynamic embedding of the t-th frame,

$$\mathbf{O}_t = b_t \cdot (\mathbf{U}_{t,b_t=1} || \mathbf{U}_{t,b_t=0}) + (1 - b_t) \cdot \mathbf{U}_{t,b_t=0}. \tag{11}$$

The video embedding $\mathbf{O} = \{\mathbf{O}_t\}_{t=1}^{T}$ and the text embedding $\mathbf{Q}$ are translated into the language space in token format, which is used to generate response from LLMs.

3.4. Training Strategy

In this work, we adopt a two-stage training scheme following previous work [30].

Stage1: Vision-Language Alignment. In the first stage, we pre-train our dynamic cooperative network while freezing both the visual encoder and LLM. It is noteworthy that we only preserve the parameter of projector $\mathcal{F}_{\text{fine}}(\cdot)$ and $\mathcal{F}_{\text{coarse}}(\cdot)$ as the initialization in the second stage. Freezing LLM in the first stage is crucial to effectively align the representation space between video content and language without sacrificing any discernible performance of LLMs.

Stage2: Instruction Tuning. After the first stage, the model possesses the ability of understanding the image within the language space, but fails to flexibly generate the reasonable and coherent linguistic responses. Therefore, in the second stage, we fully fine-tune the LLM and overall parameters in DPE module and CCE module on a instruction-following dataset. This dataset is a composite of pure text QA pairs, single- or multi-turn image QA pairs, and video QA pairs presented in a conversational format. In terms of instruction formulation, different formats are adopted for different kinds of input, and input $\langle \text{prompt} \rangle$ vary with datasets. Meanwhile, the image token $\langle \text{image} \rangle$ denotes the placeholder of image or videos, which is randomly inserted at the beginning or end of user prompt or question when training.

4. Experiments

4.1. Experimental Setup

Implementation Details. We use the pre-trained ViT-G/14 from EVA-CLIP [14] as the visual encoder to extract the features of each frame in video, and it can be further changed to other clip-based video encoders. We use pre-trained Qformer weight from InstructBLIP [11] as the textual encoder. Besides, we adopt the Vicuna-7B-1.5 model [9] as our foundational LLM. Our model is trained using 8 × NVIDIA A100 80G GPUs. See more details in the supplementary material.

Training Datasets. We leverage image-video joint training following most of works to enhance the multi-modality understanding of LLMs. Specifically, we leverage the image-to-text dataset LLaVA-filter-CC3M [49] image-caption pairs for the first stage training following LLaVA-VID [30], and LLaVA-665K [15, 22, 25, 26, 34, 40, 41, 48, 50] image QA pairs and ScienceQA [38] for the second stage training, respectively. For video-to-text dataset preparation, we use WebVid-2.5M [3] video-caption pairs for the first stage, and a subset from VideoChat2 for the second stage, including VideoChatGPT-100K [39], WebVid-10M-QA [3], NExT-QA [4], and CLEVRER [64]. And all the samples are formulated as the uniform input format as LLaMA-VID [30].

4.2. Evaluation on Short Video Understanding

Zero-shot Video-question Answering Performance. In Table 1, we report the results of our *DynFocus* against a bunch of SOTA methods on three widely-used QA benchmarks: MSVD-QA [7], MSRVTT-QA [62], and ANet-QA [19]. On

Table 1. Performance comparisons on zero-shot QA benchmark, including MSVD-QA [60], MSRVTT-QA [61], and ANet-QA [6]. We empirically observe that the default version of GPT-3.5-Turbo would significantly impact evaluation performance. Thus, we also report the possible GPT-3.5 versions for evaluation.

Methods	Size	MSVD-QA		MSRVTT-QA		ANet-QA	
		Acc	Score	Acc	Score	Acc	Score
VideoLLaMA [65]	7B	51.6	2.5	29.6	1.8	12.4	1.1
LLaMA-Adapter [66]	7B	54.9	3.1	43.8	2.7	34.2	2.7
VideoChat [27]	7B	56.3	2.8	45.0	2.5	26.5	2.2
VideoChatGPT [39]	7B	64.9	3.3	49.3	2.8	35.2	2.7
BT-Adapter [CVPR 24] [36]	7B	67.5	3.7	57.0	3.2	45.7	3.2
Chat-UniVi [CVPR 24] [23]	7B	65.0	3.6	54.6	3.1	45.8	3.2
LLaMA-VID [ECCV 24] [30]	7B	69.7	3.7	57.7	3.2	47.4	3.3
LLaMA-VID [ECCV 24] [30]	13B	70.0	3.7	58.9	3.3	47.5	3.3
VideoChat2 [ECCV 24] [28]	7B	70.0	3.9	54.1	3.3	49.1	3.3
ST-LLM [ECCV 24][37]	7B	74.6	3.9	63.2	3.4	50.9	3.3
DynFocus (Turbo-16k)	7B	72.3	3.9	59.8	3.4	49.4	3.4
DynFocus (Turbo-0613)	7B	74.8	4.0	62.8	3.6	50.3	3.4

Table 2. Performance comparisons on VCG-Bench. † represents the version that first fine-tuned on all the dataset, and further post-tuning on VideoChatGPT-100K [39] with a smaller learning rate.

Methods	Size	CI	DO	CU	TU	CO	Avg.
VideoLLaMA [65]	7B	1.96	2.18	2.16	1.82	1.79	1.98
LLaMA-Adapter [CVPR 23] [66]	7B	2.03	2.32	2.30	1.98	2.15	2.16
VideoChat [27]	7B	2.23	2.50	2.53	1.94	2.24	2.29
VideoChatGPT [ACL 24] [39]	7B	2.40	2.52	2.62	1.98	2.37	2.38
BT-Adapter [CVPR 24] [36]	7B	2.68	2.69	3.27	2.34	2.46	2.69
VTimeLLM [21]	7B	2.78	3.10	3.40	2.49	2.47	2.85
Chat-UniVi [CVPR 24] [23]	7B	2.89	2.91	3.46	2.89	2.81	2.99
LLaMA-VID [ECCV 24] [30]	7B	2.96	3.00	3.53	2.46	2.51	2.89
VideoChat2 [CVPR 24] [28]	7B	3.02	2.88	3.51	2.66	2.81	2.98
PLLaVA [CVPR 24] [28]	7B	3.21	2.86	3.62	2.33	2.93	3.12
ST-LLM [ECCV 24] [37]	7B	3.23	3.05	3.74	2.93	2.81	3.15
DynFocus	7B	3.12	3.11	3.68	2.57	2.74	3.05
DynFocus†	7B	3.27	3.15	3.78	2.86	2.78	3.17

Table 3. Performance Comparisons on LV-Bench. Input shows the number of frames each model actually process when testing. † denotes the optimal results when adopting different number of L on 200 input video frames.

Method	**Size**	**Input**	**ER**	**EU**	**KIR**	**TG**	**Rea**	**Sum**	**Overall**
Short Video MLLMs									
TimeChat [CVPR 24] [46]	7B	96 f	21.9	21.7	25.9	22.7	25.0	24.1	22.3
PLLaVA [63]	34B	16 f	25.0	24.9	26.2	21.4	30.0	25.9	26.1
LLaVA-NeXT [67]	34B	32 f	30.1	31.2	34.1	31.4	35.0	27.6	32.2
GPT-4o [43]	-	10 f	26.5	23.7	28.3	21.4	28.0	32.8	27.0
Long Video MLLMs									
MovieChat [CVPR 24] [51]	7B	∼10k f	21.3	23.1	25.9	22.3	24.0	17.2	22.5
LLaMA-VID [ECCV 24] [30]	13B	∼10k f	25.4	21.7	23.4	26.4	26.5	17.2	23.9
LWM [35]	7B	∼4k f	24.7	24.8	26.5	28.6	30.5	22.4	25.5
Gemini 1.5 Pro [45]	-	∼4k f	32.1	30.9	39.3	31.8	27.0	32.8	33.1
DynFocus† ($L = 25, K/L = 0.8$)	7B	200 f	27.9	30.3	31.2	25.4	31.8	32.8	30.4
DynFocus† ($L = 50, K/L = 0.8$)	7B	200 f	28.6	31.8	32.6	27.2	35.3	34.4	31.8
DynFocus† ($L = 60, K/L = 0.6$)	7B	200 f	29.9	33.7	35.1	25.5	33.3	26.2	32.6
DynFocus† ($L = 70, K/L = 0.4$)	7B	200 f	31.8	33.5	32.6	28.7	34.8	31.3	32.9
DynFocus† ($L = 80, K/L = 0.4$)	7B	200 f	31.1	33.5	31.6	28.6	33.8	24.1	31.8

MSRVTT-QA and MSVD-QA, our model achieves comparable results than published SOTA ST-LLM [37]. For slightly longer video ANet-QA, our method achieves competitive performance using ∼25% fewer tokens than ST-LLM, exhibiting a balance between accuracy and memory efficiency. Beyond that, we empirically observe that the marginal performance gain on short video dataset gradually decreases as the dataset scale expands during instruction tuning, which can be found in the supplementary material.

VCG-Bench Performance. Table 2 presents the results on VideoChatGPT [39] in terms of Correctness of Information (CI), Detailed Orientation (DO), Contextual Understanding (CU), Temporal Understanding (TU) and Consistency (CO). Our *DynFocus* outperforms existing video MLLMs on CI, DO, and CU. Notably, it substantially surpasses VideoChat2 [28] on CI despite using fewer instructional dataset. This may be attributed to our DPE module, which supports dynamically mitigating the visual nuisance that could hamper factual correctness. ST-LLM shows slight advantages over ours on TU for two possible reasons: (1) it performs the feature alignment between masked input and unmasked video input, which explicitly emphasizes the temporal relationship. (2) The retained tokens of each frame in ST-LLM is more than ours, and more visual details could compensate for temporal clues when handling short videos.

4.3. Evaluation on Long Video Understanding

To demonstrate the advantage of our dynamic cooperative setting, we conduct experiment on three newly released long-term video benchmark. The detailed description for each benchmark are elaborated in supplementary material.

MLVU-Bench performance. The performance of individual task and the average performance of multi-choice task (M-Avg, within 0-100%) and generation task (G-Avg, within 0.0-10.0) are both reported in Table 4. We have following observations: (1) our *DynFocus* surpasses all the open-sourced video MLLMs with a clear-cut performance gain on M-Avg and G-Avg, and it nearly consistently ranks top-2 position on individual tasks. (2) For TR, AR, and VS tasks that require an thorough understanding of entire video, our method achieves the best. We attribute this to our dynamic cooperative network's ability to balance intricate spatial details with broader temporal perception without introducing external visual nuisance. (3) Most approaches find AO and AC tasks challenging due to their sensitivity to the temporal clues, which requires recalling multiple nuanced details from lengthy videos. Although not being further fine-tuned on long-term video dataset like MovieChat, our model still performs competitively. (4) However, our method struggles with ER task that needs ego-based perspectives, likely due to the requirement for ego-centric dataset like EgoQA [12] in VideoChat2.

LV-Bench Performance. We assess six core capabilities of our model on LV-Bench: Temporal Grounding (TG), Summarization (Sum), Reasoning (Rea), Entity Recognition (ER), Event Understanding (EU), and Key Information Retrieval (KIR). The average duration of each video exceeds **1 hour**. Following [55], we select several publicly evaluated methods as baselines, with results shown in Table 3. Interestingly, some methods that excel on short videos per-

Table 4. The overall performances on MLVU. Two input strategies are adopted in evaluation: Uniform Sampling (*N fr*), which evenly samples N frames from the video; Frame Rate Sampling (*N fps*), which samples N frames per second. † denotes proprietary models.

Methods	**Input**	**Holistic**			**Single Detail**				**Multi Detail**		**M-Avg**	**G-Avg**
		TR	AR	VS	NQA	ER	PQA	SSC	AO	AC		
Short Video MLLMs												
VideoChat [27]	16 f	33.0	32.0	2.31	27.0	32.1	27.6	5.01	24.3	28.6	29.2	3.66
Video-ChatGPT [ACL 24] [39]	100 f	26.9	24.0	2.31	40.3	42.0	29.9	5.48	25.1	31.1	31.3	3.90
Video-LLaMA2 [8]	16 f	54.5	41.5	2.34	39.4	33.5	35.4	5.22	18.5	25.7	35.5	3.78
VideoChat2 [CVPR 24] [28]	16 f	74.6	51.5	2.57	42.0	47.4	43.8	5.04	22.8	29.6	44.5	3.81
Video-LLaVA [31]	8 f	71.6	57.0	2.43	53.2	45.2	48.4	5.25	20.1	35.9	47.3	3.84
Long Video MLLMs												
MovieChat [CVPR 24] [51]	2048 f	29.5	25.0	2.33	24.2	24.7	25.8	3.23	28.6	22.8	25.8	2.78
Movie-LLM [53]	1 fps	30.0	29.0	2.88	29.6	24.7	24.1	5.00	20.5	24.8	26.1	3.94
TimeChat [CVPR 24] [46]	96 f	23.1	27.0	2.54	24.5	28.4	25.8	4.29	24.7	32.0	30.9	3.42
LLaMA-VID [ECCV 24] [30]	1 fps	50.8	34.5	3.22	30.1	32.7	32.5	5.22	23.9	27.8	33.2	4.22
MA-LMM [CVPR 24] [18]	1000 f	51.9	35.5	2.12	43.1	38.9	35.8	4.80	25.1	24.3	36.4	3.46
MiniGPT4-Video [2]	90 f	70.9	52.5	2.64	49.0	48.6	44.5	4.07	23.2	23.0	44.5	3.36
DynFocus ($L = 25, K/L = 0.8$)	16 f	75.4	60.5	3.36	50.6	42.3	50.5	5.34	26.2	32.6	48.3	4.35
DynFocus ($L = 25, K/L = 0.8$)	32 f	76.2	60.9	3.36	55.5	41.5	54.0	5.39	26.8	32.8	49.6	4.38
GPT-4o† [43]	0.5 fps	87.4	74.5	4.90	64.8	57.1	65.1	6.69	56.7	46.3	64.6	5.80

Table 5. Comparisons on VideoMME with short, medium, and long durations, under the settings of "without subtitles" and "with subtitles". Notably, our method adopts 224^2 frame resolution instead of using original resolution. † denotes the model with DPO tuning.

Models	**Input**	**LLM Size**	**Short (%)**		**Medium (%)**		**Long (%)**		**Overall (%)**	
			w/o subs	w/ subs	w/o subs	w/ subs	w/o subs	w/ subs	w/o subs	w/ subs
LLaMA-VID [ECCV 24] [30]	1 fps	7B	-	-	-	-	-	-	25.9	-
Video-LLaVA [EMNLP 24] [32]	8 f	7B	45.3	46.1	38.0	40.7	36.2	38.1	39.9	41.6
ST-LLM [ECCV 24] [36]	16 f	7B	45.7	48.4	36.8	41.4	31.3	36.9	37.9	42.3
VideoChat2 [CVPR 24][28]	16 f	7B	48.3	52.8	37.0	39.4	33.2	39.2	39.5	43.8
Chat-UniVi [CVPR 24] [23]	-	7B	45.7	51.2	40.3	44.6	35.8	41.8	40.6	45.9
DynFocus ($L = 25, K/L = 0.8$)	16 f	7B	50.9	53.7	43.7	46.0	37.7	43.6	44.1	47.8
LLaVA-NeXT† [67]	-	34B	61.7	65.1	50.1	52.2	44.3	47.2	52.0	54.9
VILA-1.5 [33]	-	34B	68.1	68.9	58.1	57.4	50.8	52.0	59.0	59.4

form almost randomly in answer selection. Remarkably, our *DynFocus* achieves the best of 32.9% among all the open-sourced 7B models, even outperforming PLLaVA [28] with 34B parameters.

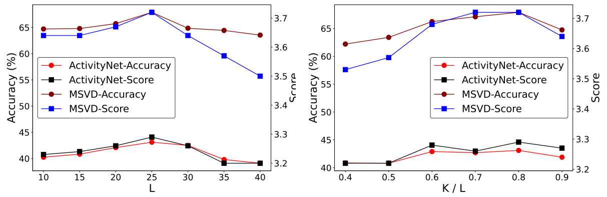


Figure 3. (a) and (b) illustrate the performance with different number of event prototypes and different ratio of filtered event prototypes, respectively.

VideoMME Benchmark Performance. VideoMME benchmark spans across three kinds of durations, VideoMME-S (∼1.3min), VideoMME-M (∼8.5min), and VideoMME-L (∼0.7h). Most videos include both subtitles and audios, which helps us investigate the performance gain from additional information sources. Table 5 compares our results with other representative video MLLMs. Notably, our *DynFocus* consistently achieves impressive advantage across different lengths of video with subtitles and without subtitles. Specifically, it exhibits the overwhelming advantage over SOTA ST-LLM and VideoChat2. Remarkably, the version of *DynFocus* without subtitles reaches an overall accuracy of 44.1%, still surpassing ST-LLM with subtitles by 1.8%.

4.4. Evaluation on Video Hallucination

Our method also achieves the competitive performance on addressing video hallucination on VideoHallucer [56]. We report the detailed results and give further analysis in the supplementary materials due to the space limitation.

4.5. Component-wise Analysis

The Effect of Number of Initial Event Prototypes. As depicted in Figure 3a, we observe that increasing L (10 $\sim$ 25) brings a consistent gain in the overall accuracy. It indicates that sufficient number of prototypes could divide the video into more fine-grained events, which would offer more abundant visual clues for accurate question answering. However, when L increases greater than 25, the performance begins to drop. This phenomenon can be explained by that increasing prototypes would hamper the intrinsic temporal structure as well as consistency. More results towards long-term video can be referred in the supplementary materials.

The Effect of Dynamic Selection. As shown in Figure 3b, a similar pattern can be observed by varying the ratio K/L,

Table 6. We report the results using different numbers of token to encode the frame with $b_t = 0$ and $b_t = 1$. Specifically, 40 tokens involves 22 multi-grained prototypes, i.e., $\mathbf{G}_t$, 16 tokens in each filtered event prototype $\mathbf{h}_t$, 1 global content token, and 1 text-guided token. 256 represents the original number of tokens without compression. $|\cdot|$ denotes the token number.

$\lvert\mathbf{U}_{b_t=0}\rvert$	$\lvert\mathbf{U}_{b_t=1}\rvert$	MSVD-QA		ANet-QA		VCG-Bench
		Acc	Score	Acc	Score	Score
0	40	63.7	3.5	41.4	3.2	2.57
0	256	65.6	3.5	42.1	3.2	2.65
2	256	68.4	3.7	44.3	3.4	2.85
2	2	62.0	3.5	40.5	3.2	2.38
2	0	58.2	3.3	38.6	2.9	2.21
2	40	67.9	3.7	43.1	3.3	2.81

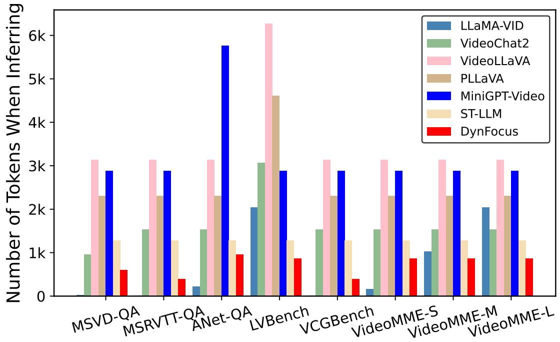
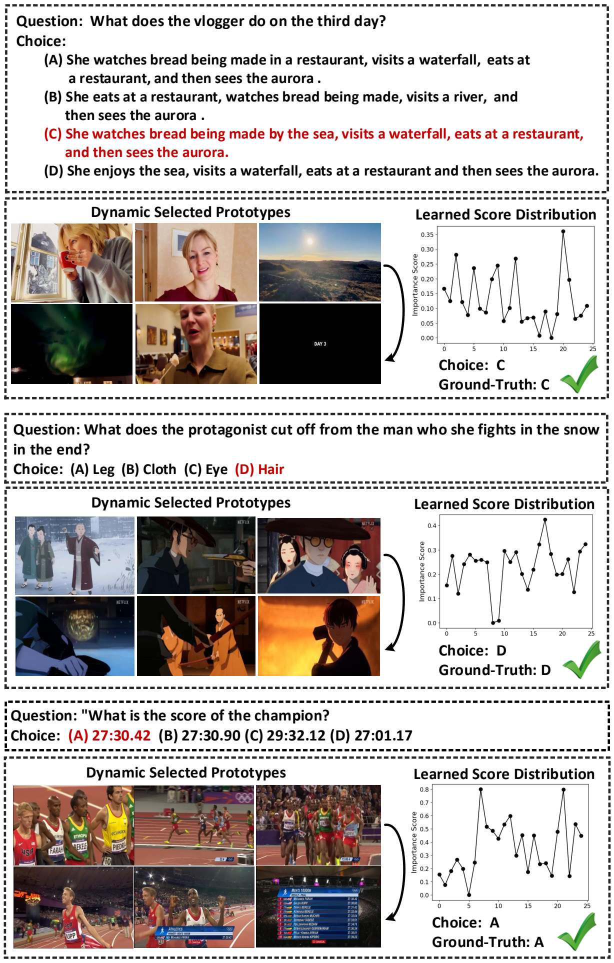


Figure 4. Token number comparison with different methods on different benchmark datasets. We calculate their token number using their released code snippet regarding loading video.

where L and K represent the number of event prototype candidates obtained by DPC-KNN clustering and filtered ones, respectively. The smaller ratio of filtered prototypes may be not enough to cover all useful visual clues, whereas the larger ratio still maintain much non-essential visual nuisance, thereby disturbing content understanding.

Visualization of Focused Frame by DPE module. As explained in DPE module, the higher score in Figure 5 indicates greater contribution to question answering. Taking the first case as an example, the question asks about the occurred events and their corresponding sequence. Although these frames with great contribution encoded by *Cones* could provide detailed visual semantics like sea, waterfall, and aurora, they may not offer sufficient temporal clues to determine the order of events. This information can be encapsulated by its complementary part, akin to *Rods*, which provides broader receptive field for capturing motion.

The Effect of Cooperation between *Cones* and *Rods*. We introduce several variants to validate the benefit of cooperation between *Cones* and *Rods*. As reported in Table 6, $|\mathbf{U}_{b_t=0}| = 0$ indicates that we discard the tokens encoded by *Rods*. We observe significant performance drop 1.7% on MSVD-QA compared with our full model. The similar pattern can be observed when completely dropping the tokens encoded by *Cones*. Moreover, although the model exhibits the best results without token compression for those important frames, it still encounter the scalability issues when extending to long-term videos, struggling to balance memory efficiency and accuracy. Figure 4 shows the comparison of total token usage, demonstrating superior advantages over existing methods.


Question: What does the vlogger do on the third day?
Choice:
(A) She watches bread being made in a restaurant, visits a waterfall, eats at a restaurant, and then sees the aurora .
(B) She eats at a restaurant, watches bread being made, visits a river, and then sees the aurora .
(C) She watches bread being made by the sea, visits a waterfall, eats at a restaurant, and then sees the aurora.
(D) She enjoys the sea, visits a waterfall, eats at a restaurant and then sees the aurora.

Question: What does the protagonist cut off from the man who she fights in the snow in the end?
Choice: (A) Leg (B) Cloth (C) Eye (D) Hair

Question: "What is the score of the champion?
Choice: (A) 27:30.42 (B) 27:30.90 (C) 29:32.12 (D) 27:01.17

Figure 5. We showcase the filtered event prototypes focused by DPE module on LV-Bench. To save space, we only showcase the prototype with top-6 score sequentially. The figure at the right-bottom corner illustrates the learned score distribution on the event prototype candidates (L=25) obtained by DPC-KNN.

5. Conclusion

In this paper, we develop a dynamic cooperative network for memory-efficient encoding. We experimentally delve into the network behavior and find that dynamic encoding could simultaneously achieves fine spatial visual appearance understanding and coarse temporal dynamics perception using affordable tokens, striking a balance between answering accuracy and memory efficiency. Our model achieves superior performance with substantially few tokens on both short and long video benchmarks. Moreover, our model also demonstrates the great potential on addressing video hallucination.

6. Acknowledgment

This work was supported by the National Natural Science Foundation of China (62302045) and the Beijing Institute of Technology Special-Zone.

References

[1] *Perturbation Techniques in Online Learning and Optimization*, pages 233–264. 2017.

[2] Kirolos Ataallah, Xiaoqian Shen, Eslam Abdelrahman, Essam Sleiman, Deyao Zhu, Jian Ding, and Mohamed Elhoseiny. Minigpt4-video: Advancing multimodal llms for video understanding with interleaved visual-textual tokens. *CoRR*, abs/2404.03413, 2024.

[3] Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. Frozen in time: A joint video and image encoder for end-to-end retrieval. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1728–1738, 2021.

[4] Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. Frozen in time: A joint video and image encoder for end-to-end retrieval. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 1708–1718. IEEE, 2021.

[5] Quentin Berthet, Mathieu Blondel, Olivier Teboul, Marco Cuturi, Jean-Philippe Vert, and Francis R. Bach. Learning with differentiable perturbed optimizers. *CoRR*, abs/2002.08676, 2020.

[6] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *Proceedings of the ieee conference on computer vision and pattern recognition*, pages 961–970, 2015.

[7] David L. Chen and William B. Dolan. Collecting highly parallel data for paraphrase evaluation. In Dekang Lin, Yuji Matsumoto, and Rada Mihalcea, editors, *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, 19-24 June, 2011, Portland, Oregon, USA*, pages 190–200. The Association for Computer Linguistics, 2011.

[8] Zesen Cheng, Sicong Leng, Hang Zhang, Yifei Xin, Xin Li, Guanzheng Chen, Yongxin Zhu, Wenqi Zhang, Ziyang Luo, Deli Zhao, and Lidong Bing. Videollama 2: Advancing spatial-temporal modeling and audio understanding in video-llms. *CoRR*, abs/2406.07476, 2024.

[9] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, March 2023.

[10] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113, 2023.

[11] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven C. H. Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.

[12] John Doe and Jane Smith. EgoQA: Egocentric question answering. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[13] Mingjing Du, Shifei Ding, and Hongjie Jia. Study on density peaks clustering based on k-nearest neighbors and principal component analysis. *Knowl. Based Syst.*, 99:135–145, 2016.

[14] Yuxin Fang, Wen Wang, Binhui Xie, Quan Sun, Ledell Wu, Xinggang Wang, Tiejun Huang, Xinlong Wang, and Yue Cao. EVA: exploring the limits of masked visual representation learning at scale. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pages 19358–19369. IEEE, 2023.

[15] Yash Goyal, Tejas Khot, Aishwarya Agrawal, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the V in VQA matter: Elevating the role of image understanding in visual question answering. *Int. J. Comput. Vis.*, 127(4):398–414, 2019.

[16] Yudong Han, Jianhua Yin, Jianlong Wu, Yinwei Wei, and Liqiang Nie. Semantic-aware modular capsule routing for visual question answering. *IEEE Trans. Image Process.*, 32:5537–5549, 2023.

[17] Bo He, Hengduo Li, Young Kyun Jang, Menglin Jia, Xuefei Cao, Ashish Shah, Abhinav Shrivastava, and Ser-Nam Lim. MA-LMM: memory-augmented large multimodal model for long-term video understanding. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pages 13504–13514. IEEE, 2024.

[18] Bo He, Hengduo Li, Young Kyun Jang, Menglin Jia, Xuefei Cao, Ashish Shah, Abhinav Shrivastava, and Ser-Nam Lim. Ma-lmm: Memory-augmented large multimodal model for long-term video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[19] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 961–970. IEEE Computer Society, 2015.

[20] D. C. Hood and M. A. Finkelstein. *Rod and cone contributions to human brightness perception*, pages 7023–7027. 1986.

[21] Bin Huang, Xin Wang, Hong Chen, Zihan Song, and Wenwu Zhu. Vtimellm: Empower llm to grasp video moments. *arXiv preprint arXiv:2311.18445*, 2023.

[22] Drew Hudson and Christopher D Manning. Gqa: A new dataset for real-world visual reasoning and compositional questions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[23] Peng Jin, Ryuichi Takanobu, Caiwan Zhang, Xiaochun Cao, and Li Yuan. Chat-univi: Unified visual representation empowers large language models with image and video understanding. *arXiv preprint arXiv:2311.08046*, 2023.

[24] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B.

Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *CoRR*, abs/2001.08361, 2020.

[25] Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara L. Berg. Referitgame: Referring to objects in photographs of natural scenes. In Alessandro Moschitti, Bo Pang, and Walter Daelemans, editors, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 787–798. ACL, 2014.

[26] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 2017.

[27] KunChang Li, Yinan He, Yi Wang, Yizhuo Li, Wenhai Wang, Ping Luo, Yali Wang, Limin Wang, and Yu Qiao. Videochat: Chat-centric video understanding. *arXiv preprint arXiv:2305.06355*, 2023.

[28] Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Luo, et al. Mvbench: A comprehensive multi-modal video understanding benchmark. *arXiv preprint arXiv:2311.17005*, 2023.

[29] Yanwei Li, Lin Song, Yukang Chen, Zeming Li, Xiangyu Zhang, Xingang Wang, and Jian Sun. Learning dynamic routing for semantic segmentation. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 8550–8559. Computer Vision Foundation / IEEE, 2020.

[30] Yanwei Li, Chengyao Wang, and Jiaya Jia. Llama-vid: An image is worth 2 tokens in large language models. *arXiv preprint arXiv:2311.17043*, 2023.

[31] Bin Lin, Yang Ye, Bin Zhu, Jiaxi Cui, Munan Ning, Peng Jin, and Li Yuan. Video-llava: Learning united visual representation by alignment before projection. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 16-16, 2024*, pages 5971–5984. Association for Computational Linguistics, 2024.

[32] Bin Lin, Bin Zhu, Yang Ye, Munan Ning, Peng Jin, and Li Yuan. Video-llava: Learning united visual representation by alignment before projection. *arXiv preprint arXiv:2311.10122*, 2023.

[33] Ji Lin, Hongxu Yin, Wei Ping, Yao Lu, Pavlo Molchanov, Andrew Tao, Huizi Mao, Jan Kautz, Mohammad Shoeybi, and Song Han. Vila: On pre-training for visual language models. *ArXiv preprint*, 2023.

[34] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning, 2023.

[35] Hao Liu, Wilson Yan, Matei Zaharia, and Pieter Abbeel. World model on million-length video and language with blockwise ringattention. *ArXiv*, abs/2402.08268, 2024.

[36] Ruyang Liu, Chen Li, Yixiao Ge, Ying Shan, Thomas H Li, and Ge Li. One for all: Video conversation is feasible without video instruction tuning. *arXiv preprint arXiv:2309.15785*, 2023.

[37] Ruyang Liu, Chen Li, Haoran Tang, Yixiao Ge, Ying Shan, and Ge Li. ST-LLM: large language models are effective temporal learners. *CoRR*, abs/2404.00308, 2024.

[38] Pan Lu, Swaroop Mishra, Tony Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. In *The 36th Conference on Neural Information Processing Systems (NeurIPS)*, 2022.

[39] Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Shahbaz Khan. Video-chatgpt: Towards detailed video understanding via large vision and language models. *arXiv preprint arXiv:2306.05424*, 2023.

[40] Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L. Yuille, and Kevin Murphy. Generation and comprehension of unambiguous object descriptions. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, 2016.

[41] Anand Mishra, Shashank Shekhar, Ajeet Kumar Singh, and Anirban Chakraborty. Ocr-vqa: Visual question answering by reading text in images. In *2019 international conference on document analysis and recognition (ICDAR)*, pages 947–952. IEEE, 2019.

[42] OpenAI. Introducing chatgpt. 2022.

[43] OpenAI. GPT-4o system card, 2024.

[44] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.

[45] Machel Reid, Nikolay Savinov, Denis Teplyashin, Dmitry Lepikhin, Timothy Lillicrap, Jean-baptiste Alayrac, Radu Soricut, Angeliki Lazaridou, Orhan Firat, Julian Schrittwieser, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *ArXiv preprint*, 2024.

[46] Shuhuai Ren, Linli Yao, Shicheng Li, Xu Sun, and Lu Hou. Timechat: A time-sensitive multimodal large language model for long video understanding. *ArXiv*, 2023.

[47] Robert W. Massof Sarah L. Pardue. Differential roles of rods and cones in visual acuity and motion detection. *Investigative Ophthalmology and Visual Science (IOVS)*, 2021.

[48] Dustin Schwenk, Apoorv Khandelwal, Christopher Clark, Kenneth Marino, and Roozbeh Mottaghi. A-okvqa: A benchmark for visual question answering using world knowledge. In *European Conference on Computer Vision*, pages 146–162. Springer, 2022.

[49] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In Iryna Gurevych and Yusuke Miyao, editors, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 2556–2565. Association for Computational Linguistics, 2018.

[50] Oleksii Sidorov, Ronghang Hu, Marcus Rohrbach, and Amanpreet Singh. Textcaps: a dataset for image captioning with reading comprehension. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*. Springer, 2020.

[51] Enxin Song, Wenhao Chai, Guanhong Wang, Yucheng Zhang,

Haoyang Zhou, Feiyang Wu, Xun Guo, Tian Ye, Yan Lu, Jenq-Neng Hwang, et al. Moviechat: From dense token to sparse memory for long video understanding. *arXiv preprint arXiv:2307.16449*, 2023.

[52] Enxin Song, Wenhao Chai, Guanhong Wang, Yucheng Zhang, Haoyang Zhou, Feiyang Wu, Xun Guo, Tianbo Ye, Yang Lu, Jenq-Neng Hwang, and Gaoang Wang. Moviechat: From dense token to sparse memory for long video understanding. *ArXiv preprint*, 2023.

[53] Zhende Song, Chenchen Wang, Jiamu Sheng, Chi Zhang, Gang Yu, Jiayuan Fan, and Tao Chen. Moviellm: Enhancing long video understanding with ai-generated movies. *CoRR*, abs/2403.01422, 2024.

[54] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[55] Weihan Wang, Zehai He, Wenyi Hong, Yean Cheng, Xiaohan Zhang, Ji Qi, Shiyu Huang, Bin Xu, Yuxiao Dong, Ming Ding, and Jie Tang. Lvbench: An extreme long video understanding benchmark. *CoRR*, abs/2406.08035, 2024.

[56] Yuxuan Wang, Yueqian Wang, Dongyan Zhao, Cihang Xie, and Zilong Zheng. Videohallucer: Evaluating intrinsic and extrinsic hallucinations in large video-language models. *CoRR*, abs/2406.16338, 2024.

[57] Yuxuan Wang, Cihang Xie, Yang Liu, and Zilong Zheng. Videollamb: Long video understanding with recurrent memory bridges. *arxiv*, 2024.

[58] Wenhao Wu. Freeva: Offline MLLM as training-free video assistant. *CoRR*, abs/2405.07798, 2024.

[59] Zuxuan Wu, Tushar Nagarajan, Abhishek Kumar, Steven Rennie, Larry S. Davis, Kristen Grauman, and Rogério Schmidt Feris. Blockdrop: Dynamic inference paths in residual networks. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 8817–8826. Computer Vision Foundation / IEEE Computer Society, 2018.

[60] Zuxuan Wu, Ting Yao, Yanwei Fu, and Yu-Gang Jiang. Deep learning for video classification and captioning. In *Frontiers of multimedia research*, pages 3–29. ACM, 2017.

[61] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. Msr-vtt: A large video description dataset for bridging video and language. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5288–5296, 2016.

[62] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. MSR-VTT: A large video description dataset for bridging video and language. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 5288–5296. IEEE Computer Society, 2016.

[63] Lin Xu, Yilin Zhao, Daquan Zhou, Zhijie Lin, See Kiong Ng, and Jiashi Feng. Pllava: Parameter-free llava extension from images to videos for video dense captioning. *arXiv preprint arXiv:2404.16994*, 2024.

[64] Kexin Yi, Chuang Gan, Yunzhu Li, Pushmeet Kohli, Jiajun Wu, Antonio Torralba, and Joshua B. Tenenbaum. CLEVRER: collision events for video representation and reasoning. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

[65] Hang Zhang, Xin Li, and Lidong Bing. Video-llama: An instruction-tuned audio-visual language model for video understanding. *arXiv preprint arXiv:2306.02858*, 2023.

[66] Renrui Zhang, Jiaming Han, Aojun Zhou, Xiangfei Hu, Shilin Yan, Pan Lu, Hongsheng Li, Peng Gao, and Yu Qiao. Llama-adapter: Efficient fine-tuning of language models with zero-init attention. In *ICLR*, 2023.

[67] Yuanhan Zhang, Bo Li, haotian Liu, Yong jae Lee, Liangke Gui, Di Fu, Jiashi Feng, Ziwei Liu, and Chunyuan Li. Llava-next: A strong zero-shot video understanding model, 2024.