Self-Adaptive Sampling for Efficient Video Question Answering on Image–Text Models

Wei Han [†*] **Hui Chen** [†] **Min-Yen Kan** [‡] **Soujanya Poria** [†]
[†] Singapore University of Technology and Design, [‡] National University of Singapore

Abstract

Image–text models (ITMs) are the prevalent architecture to solve video question–answering tasks. ITMs requires only a few input frames, saving significant computation over against video–language models. However, we find existing ITM video question–answering either 1) adopts simplistic and unintentional sampling strategies, which may miss key frames that offer answer clues; or 2) samples a large number of frames into divided groups, which computational sources can not accommodate. We develop an efficient sampling method for the few-frame scenario. We first summarize a family of prior sampling methods based on question–frame correlation into a unified one, dubbed *Most Implied Frames* (MIF). Through analysis, we form a hypothesis that question-aware sampling is not necessary, from which we further propose the second method *Most Dominant Frames* (MDF). Results on four public datasets and three ITMs demonstrate that MIF and MDF boost the performance for image–text pretrained models, and have a wide application over both model architectures and datasets. Code is available at https://github.com/declare-lab/Sealing.

1 Introduction

With the advancement in computer vision technology, we are witnessing an explosive surge of visual data. Together, research in vision–language understanding has progressed significantly in the past decade, challenging a wide variety multimodal application tasks (Wang et al., 2021; Radford et al., 2021; Jia et al., 2021; Alayrac et al., 2022; Li et al., 2023), such as image captioning, visual question answering and multimodal retrieval. With the continuing improvement in computation, researchers have extended conventional image–text models (ITMs) to video–text ones, mainly by substituting image encoders with their video counterparts (Yang et al., 2021, 2022; Zellers et al., 2021; Fu et al., 2021). This learning paradigm achieves decent performance on numerous video–text tasks, as it incorporates temporal features into modeling. Nevertheless, 3D convolution, the core technique adopted in these video–text pretrained models, demands tremendous computational power in terms of both time and memory, limiting models' deployment on consumer-level devices.

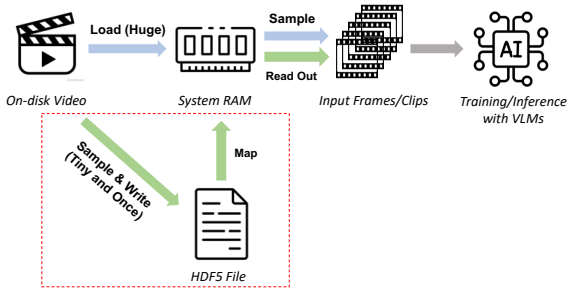


Figure 1: Comparison between conventional I/O (online sampling) and ours. The blue and green arrows distinguish the dataflow between online sampling methods and ours until the end of preprocessing. The red box highlights the process we alter from conventional routines.

A straightforward solution to reduce overhead is to extract solely those *keyframes* that describe the main content or are related to the task from a given video, so that image–text models can preprocess them (Rasheed et al., 2022; Wang et al., 2022; Li et al., 2023). Contemporary auto-regressive ITMs manage to adapt themselves to video–text tasks with a few frames sampled from those videos and yield promising results (Rasheed et al., 2022; Wang et al., 2022). In this family of approaches, image frames or clips (consecutive frames, as shown in Fig. 2a) are sampled from raw videos, cut into patches, and then encoded through a visual encoder (e.g., ResNet (He et al., 2016) and ViT (Dosovitskiy et al., 2020)). X-CLIP (Ni et al., 2022) further

*Corresponding authors: wei_han@mymail.sutd.edu.sg, hui_chen@mymail.sutd.edu.sg

Findings of the Association for Computational Linguistics: NAACL 2024, pages 2522–2534

inserts cross-frame communication modules to construct connections across timestamps.

Despite performing well, we observe that the sampling strategies employed in these models are simplistic: they are blind to the video and question and only base on statistical probability distributions (Fig. 2a). These data-agnostic approaches inevitably limit the performance when finetuning and inferring on these ITMs, since they may cause keyframe omission (Fig. 3).

On the other hand, recent works (Li et al., 2022b,c; Wei et al., 2023) introduce learning-based sampling methods. Assisted by the Gumbel-Softmax trick (Jang et al., 2016), they build a parametric sampling network and concatenate that to the backbone. Then, as an auxiliary module, the parametric sampling strategy is jointly optimized with the main video–QA task. Although these frameworks gain competitive performance, they have the following drawbacks. First, they sacrifice efficiency owing to the additional overhead and the slow convergence speed caused by the devised sampling network, compared to direct few-frame finetuning on ITMs (from less than 10 epochs to more than 50 epochs) (Li et al., 2022c; Wei et al., 2023). Secondly, it also undermines flexibility—the intervention touches the preprocessing stage in these works (Li et al., 2022c; Wei et al., 2023). They encode the pre-sampled clips with customized pretrained video encoders, like 3D ResNet101 (Hara et al., 2018) or CLIP (Radford et al., 2021), leading to incompatibility with ITMs which only accept raw images as input. Additionally, the sampling network must be optimized along with the backbones on such clip features, which deters them from being directly applied to ITMs.

To address these issues, we first explore the correlation between model's performance and the frames output from captioning-based samplers. Specifically, we propose a learning-free sampling method, dubbed *Most Implied Frames* (MIF), which we show is an simplified and unification of previous V(isual) Q(uestion)-aware methods. It utilizes lightweight pretrained models to annotate frames and grades each of them with a caption–question score. The selected frames are those with highest scores, or the best captions that *imply* the answer. Then, we conclude from empirical studies on MIF that capturing the most question-related frames is not a prerequisite for better accuracy. Based on our analysis, we hypothesize that question-aware sampling is not necessary and propose another self-adaptive sampling strategy—*Most Dominant Frames* (MDF). The underlying logic is to diversify the input frames to minimize the *dominant* scenes in that video, because most of the answers can be answered from *static scenes* instead of *dynamic segments*. To this end, we first define a goal function that measures the dynamics in videos whose input is the visual feature encoded by the backbone model's inherent image encoder. Then we devise a search algorithm to quickly locate the frames where features move slowest in that video. Since question content no longer participates in the sampling process, MDF is a V-aware Q-agnostic method. In implementation, both MIF and MDF are executed in an offline fashion Figure 1, enhancing the training efficiency compared to those online sampling algorithms. We further conduct experiments on three ITMs (CLIP (Radford et al., 2021), GIT (Wang et al., 2022) and All-in-one (Wang et al., 2023)) using four widely tested video QA datasets. The results show that both methods are feasible solutions towards Video–QA tasks on ITMs, among which MDF can provide better efficiency, and indirectly substantiating the correctness of our hypothesis.

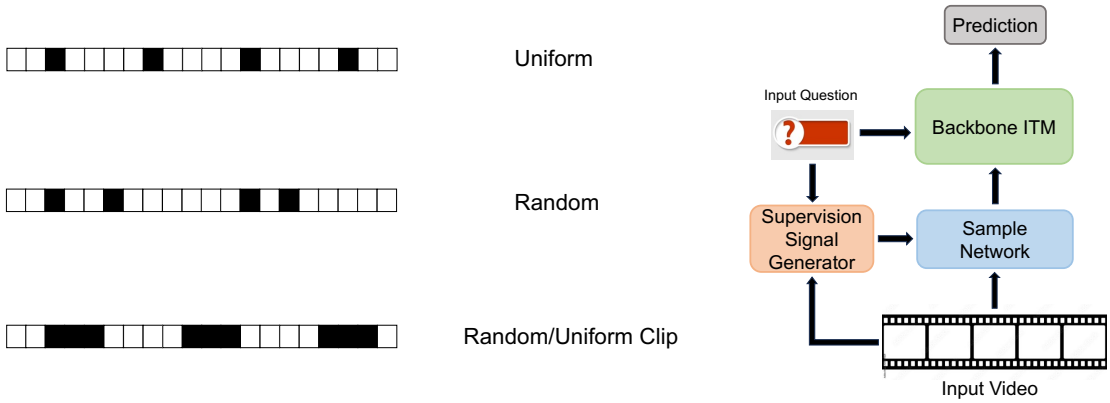
The contributions in our paper are as follows:

- We propose MIF, an offline question-aware sampling method for video question answering, which leverages two backbone models as captioner and scorer respectively.
- Based on the analysis of the MIF experimental results, we hypothesize that question-aware is redundant and propose a more efficient question-agnostic sampling method, MDF.
- We conduct comprehensive evaluation on a large variety of datasets and models. MDF yields competitive results with MIF, and both methods exceeds strong baselines, which also substantiates our hypothesis.

2 Related Work

2.1 Visual Language Models

Since the remarkable success of vision language models (VLMs) like CLIP (Radford et al., 2021) and ALIGN (Jia et al., 2021) in the field of zero-shot multimodal learning, there is a growing trend in training large VLMs through minimizing image–text contrastive loss (Li et al., 2020;



(a) Heuristic Sampling (b) Learning Based Sampling

Figure 2: Existing sample strategies for video–question answering tasks. In heuristic sampling, the black boxes indicate selected frames.

Figure 3: Randomly sampled video frames from the msrvtt-qa dataset and two questions. The bracketed timestamps indicate cues for corresponding answers from the video. The QA pair in the red box cannot be grounded from the four sampled frames.

Kim et al., 2021; Zhang et al., 2021; Yu et al., 2022) to achieve cross-modality semantic alignment. Early VLMs for multi-task purposes frequently adopt a bi-encoder architecture (Radford et al., 2021; Li et al., 2021, 2022a), where visual and textual modality are separately encoded in their individual encoders and finally combined to complete downstream tasks. Recent achievements resort to the more efficient GPT-style (Brown et al., 2020) architecture, which takes the output sequences from visual encoders as the visual prefixes and jointly tunes the decoder and visual encoder (Tsimpoukelli et al., 2021; Alayrac et al., 2022; Li et al., 2023). When confronted with video data, a common practice (Seo et al., 2020; Yang et al., 2021) replaces image encoders in these ITMs with video encoders that can capture temporal correlations, like S3D (Xie et al., 2017) and video Swin-Transformer (Liu et al., 2021b).

2.2 Sampling Techniques in Video Question–Answering Tasks

To apply ITMs on video understanding tasks, sampling is demanded to convert streaming data into discrete frames. Most of current sampling algorithms are online algorithms, i.e., sampling happens after loading the streaming-in video data into the memory. The heuristic sampling methods (Fig. 2a) are prevalent in default ITM implementations (Lei et al., 2021b; Fu et al., 2021; Wang et al., 2022, 2023), since these algorithms are learning-free and convenient to adjust. However, Buch et al. (2022) points out that for most video understanding tasks, understanding of event temporality is often not necessary to achieve strong or state-of-the-art performance. Therefore, recent works turns to integrate the sampling module into the entire learning frameworks. As shown in Fig. 2b, this kind of architectures usually has a parameterized sampler, which is trained with pseudo labels generated from a question-guided indices generator and then jointly optimized with the predictions of the main task (Li et al., 2022b,c; Wei et al., 2023). Based on the causal theory (Pearl et al., 2016), Li et al. (2022b) separate the clips into causal and complement ones;while Li et al. (2022c) and Wei et al. (2023) consider invariant/transient and positive/negative scenes. Distinct to these online sampling algorithms, our proposed methods are totally offline and learning-free, but sufficiently utilizes the inherent knowledge learned by these ITMs during pretraining. Finally, the sampled frames are saved into HDF5 files for fast loading during fine-tuning, which greatly cut off the training time.

3 Method

In this section, we first briefly recap the background of the video-QA task on ITMs. Then we introduce the *Most Implied Frames* (MIF), a generalization to previous question–aware sampling approaches. We report some primary results and describe our key findings to the statistics. Finally, based on these discoveries we introduce the more efficient *Most Dominant Frames* (MDF).

3.1 Problem Definition

Given a short video $V = \{v_1, v_2, ..., v_T\}$ of T frames and a literal question $Q = \{q_1, q_2, ..., q_l\}$ of l tokens, an ITM $\mathcal{M}$ is expected to generate an answer $\hat{A} = \{\hat{a}_i\}_{i=1}^{n}$ (generative setting, $n \geq 1$) or the answer index (multiple choice setting, $n = 1$) to match a reference answer which serves as a valid response to the given question.

$$\hat{A} = \mathcal{M}(V', Q) \tag{1}$$

where $V' \subset V$ is the set of sampled frames.

In evaluation, we use item-wise accuracy as the performance metric, defined as:

$$acc = \frac{1}{|\mathbf{Q}|} \sum_{i=1}^{|\mathbf{Q}|} \mathbf{1}(\hat{A}_i = A_i) \tag{2}$$

where $\mathbf{Q}$ is the entire set of questions in the dataset, $\mathbf{1}(\cdot)$ is the indicator function that equals 1 only if the expression is true. The predictions can be either generated through direct generation (generative setting) or classification (multiple choice setting). See Appendix C for more details.

3.2 Most Implied Frames (MIF)

MIF uses a caption model $\mathcal{M}_c$ and a set of grading models $\mathcal{M}_g$ to select the best frame candidates, as illustrated in Fig. 4. Given a question, MIF could also be termed "cue frame retrieval". Before starting the process, following previous work (Buch et al., 2022; Li et al., 2022c), we reduce the computational cost by uniformly sampling T' ($T' << T$) frames from the original video, with indices as $\{t_1, t_2, ..., t_{T'}\} \subset \{1, 2, ..., T\}$. The caption model $\mathcal{M}_c$ takes all downsampled frames as input and generates a description C. Then $\mathcal{M}_g$ computes the matching score s between the question Q and the generated description ($s = \mathcal{M}_g(Q, C)$). We presume that the matching score s indicates the possibility that each frame can serve as a cue to answer the given question. Therefore, we rank all frames by score, selecting the highest N frames as the sample (indicated by indices):

$$i_1, i_2, ..., i_N = \arg\operatorname*{topk}_t(\{s_{t_1}, s_{t_2}, ..., s_{t_{T'}}\}, N) \tag{3}$$

where s_t is the matching score for frame v_t. Notably, MIF is a QA–aware algorithm. For questions posted under the same video, MIF usually generates different sets of sampling results.

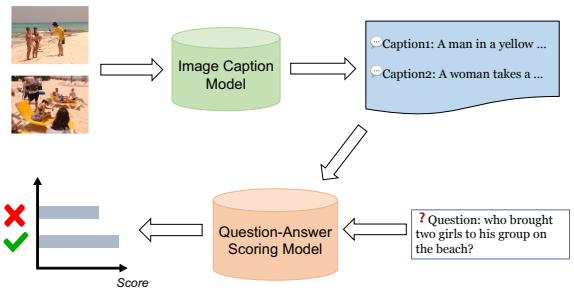


Figure 4: MIF workflow. Here we just show an example of how it selects one frame out of two frames.

3.3 Primary Results on MIF

The main results by MIF can be found in Table 2, Table 3 and Table 4. All experiments leverage the base version of GIT (consistent with target model) to generate captions and BERT[1] fine-tuned on many prevalent textual question answering datasets (SQuAD (Rajpurkar et al., 2018), RACE (Lai et al., 2017), CoQA (Reddy et al., 2019) and MSMARCO (Nguyen et al., 2016)) as the grader to calculate question–caption correlation score. The increment of accuracy is significant on all backbone models and datasets compared to state-of-the-art baselines, showing that MIF is a promising solution when performing video understanding tasks on ITMs.

Upon the decent performance, we are curious about the correlation between accuracy and captioner/grader model sizes in MIF—for which we form our first research question below.

RQ1: Are stronger captioning or scoring models bound to bring better results?

To provide a potential response, we systematically study MIF by testing frames picked via two general types of samplers on GIT-Base: i) two separate models for captioning and grading; ii) BLIP-2 pretrained on QVHighlights (Lei et al., 2021a) as

[1]https://huggingface.co/iarfmoose/bert-base-cased-qa-evaluator

a unified model for question-aware key-frame extraction (Yu et al., 2024).

$\mathcal{M}_c$	$\mathcal{M}_g$	MSVD	MSRVTT
Separate Model			
GIT-S	BERT-S	46.5	42.3
GIT-B	BERT-B	46.7	42.4
GIT-L	BERT-L	46.9	42.1
Unified Model			
BLIP2-T5-XL		46.6	42.0
BLIP2-T5-XXL		46.2	42.2

Table 1: Results of MSVD-QA and MSRVTT-QA on GIT using frames sampled from different captioner-grader combinations. The number of input frames are fixed at 6. "GIT-B" and "Bert-B" is the default implementation in later sections.

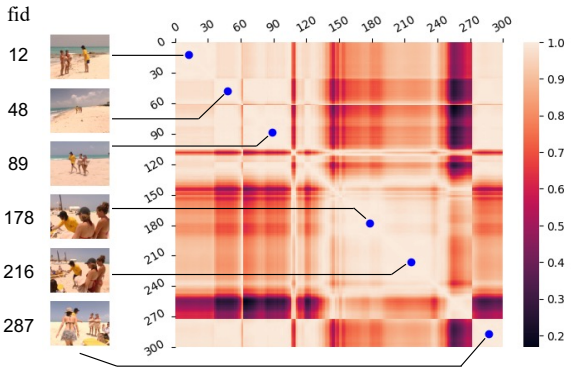


Figure 5: Sample MDF processing (6 frames). The heatmap visualizes the calculated frame similarity matrix as the cosine value between pairs of frame vectors. The entry at i^{th} row j^{th} column represents the similarity between frames i and j. Blue points indicate the frames eventually extracted.

Among these results, we find that there is no significant correlation between the size of caption-grading system and the accuracy of Video–QA task, though larger models may produce more informative and accurate captions and scores overall. Now that question-guided sampler has reached its upper bound, we make a bold hypothesis:

Hypothesis: *Question-agnostic sampling methods can perform as well as question-aware ones.*

RQ2: Can we design a question-agnostic sampler? To provide a possible solution, we propose another method, *Most Dominant Frames* (MDF), in the following section, powered by the inherent vision-encoder of ITMs.

3.4 Most Dominant Frames (MDF)

It has been pointed out in early video sampling works (Shahraray, 1995; Nam and Tewfik, 1999) that the sampling rate in each temporal region should be proportional to the object motion speed. Besides, because the frame lengths are usually fixed in ITMs (3 or 6 in our experiments), if the sampled frames are temporally closed, at a large chance they will share analogous contents and some key frames may be missing.

To this end, we construct our solution based on the ITM's cognition towards the frames from its own vision module. The first intuition comes from the theory and experience of representation learning from large pretrained models (Bengio et al., 2013; Devlin et al., 2018; Dosovitskiy et al., 2020), where learned representation output from well-tuned large models embed meaningful semantic information. We harness the inherent vision encoder of the VLM (if it has one) to acquire visual embeddings $E = \{e_1, e_2, ..., e_T\}$. To quantify the invariance in each frame, we define the following metric $dom(t)$ (the abbreviation of dominant) for frame v_t at timestamp t.

$$dom(t) = \sum_{t'=t-W}^{t+W} \mathbf{sim}(e_t, e'_t) \tag{4}$$

The problem then can be formulated as seeking N local minima of $dom(t)$ with respect to time $\tau = \{t_1, t_2, ..., t_N\} \subset \{1, 2, ..., T\}$, subject to $|\tau_i - \tau_{i+1}| \geq W$.

The details of the algorithm is given in Algorithm 1. Considering the disparity in the lengths of videos, instead of keeping a constant W, we set W automatically in an self-adaptive way:

$$W_V = L_V/(\lambda \cdot N) \tag{5}$$

where L_V is the length of video V in terms of frame numbers, λ is the constant width-adjusting rate that controls the scope to search in every steps. Fig. 5 visualizes an example of searching results on the similarity map.

4 Experiments

Datasets. To evaluate our proposed methods, we conduct extensive experiments on the following 4 frequently tested datasets:

Algorithm 1: Most Dominant Frames (MDF)

Input: Video frames $V = \{v_1, v_2, ..., v_T\}$, vision model $\mathcal{M}$, width-adjusting rate λ
Output: Visual prefix $F = \{f_1, f_2, ..., f_N\}$

Encode frames using the vision model $E = \mathcal{M}(V) = \{e_1, e_2, ..., e_T\}$
Compute *dom* score for all frames and set W, according to Eq. 4 and Eq. 5.
Init $F = \{f_{\arg\max_t dom(t)}\}$, index set $I = \{0, 1, ..., i - W, i + W, ..., T\}$
while $|F| < N$ *and* $I \neq \oslash$ **do**
    $t' \leftarrow \arg\max_t dom(t)$
    $F \leftarrow F \cup \{f_{t'}\}$
    $I \leftarrow I \setminus \{t''\}_{t''-t'<W}$
if $|F| < N$ **then**
    $\tau \leftarrow \texttt{argtop}_N(\{dom(t)\}_{t \in T})$
    return $F \cup \{f'_t\}_{t' \in \tau}$
else
    return F

MSVD-QA and MSRVTT-QA. These two datasets (Xu et al., 2016a) are adapted from corresponding video captioning datasets—Microsoft Research Video Description Corpus (Chen and Dolan, 2011) and Microsoft Research Video to Text (Xu et al., 2016b). Both datasets provide same five types of questions—*what, where, who, when, how*. The answers to the questions are all single words.

TGIF-QA. The TGIF-QA (Jang et al., 2019) dataset contains 165K QA pairs for the animated GIFs from the TGIF dataset (Li et al., 2016). Its question–answer pairs are annotated via crowdsourcing with a carefully designed user interface to ensure quality. TGIF-QA offers three question types: frame, transition, and (repetition) count. We folllow previous common benchmarking work (Fu et al., 2021; Wang et al., 2022; Xiao et al., 2022) and test only on the frame-QA task.

NExT–QA. The NExT-QA dataset (Xiao et al., 2022) targets at reasoning from causal and temporal relationships between actions. There are three question types in NExT–QA: descriptive, temporal and causal reasoning, which respectively targets at evaluating model's different aspects of capability. There are two versions for the composition of questions and answers: open-ended and multiple choice (MC). We test our methods on the MC setting following the most common practice.

4.1 Backbone Models

CLIP CLIP (Rasheed et al., 2022) is the first ITM that focuses on zero-shot transfer onto diverse multimodal downstream tasks. It is composed of two modality-specific encoders to process input modality signals separately. In our experiments, we also modify its structure by adding a single-layer transformer decoder on the top of the two encoders (dubbed "CLIP-dec" but we still use "CLIP" to denote it for simplicity). We decode for only one step to get the answer, not alike other generative ITMs that predict the whole sequence containing both the question and answer words.

GIT (Wang et al., 2022) is one of the state-of-the-art ITMs for video question answering tasks, released by Microsoft Research. It adopts ViT-B-16 (Radford et al., 2021) as its visual encoder and a GPT-style decoder that receives both the encoded image patches (as visual prefix) and textual embeddings to generate the output text. Currently the GIT family consists of four versions[2]. In our experiments, we tune GIT-Base on these three datasets (denoted as GIT in later context for simplicity).

All-in-one (AIO) (Wang et al., 2023) is another family of ITMs which follows the philosophy of *learning-by-fusion*. The model is composed of stacked multimodal attention layers called a unified transformer that takes concatenated video–text input as the basic fusion modules. Similar to the previous two ITMs, it can adapted to employ output embeddings to solve many downstream video–language tasks. Particularly, we use All-in-one(-Base) in all our experiments.

In what follows, by default "CLIP" and "AI" respectively denote CLIP-ViT-base-patch16[3] with a decoder and All-in-one-Base[4]. For GIT-related models, we follow (Wang et al., 2022) to finetune the pretrained GIT-Base[5] on four datasets).

4.2 Baselines

Direct Finetuning We first consider directly finetuning each backbone model, which can be categorized into online learning-free sampling. Since the exact sampling strategy adopted by GIT is unknown, we examine the results using uniform sampling and find that they are closed to the reported numbers on three datasets. Hence, we treat uniform sampling as baseline for GIT and CLIP-series (because there is not open-sourced implementation provided for CLIP on these datasets as well). As

[2]GIT-Base, GIT-Large, GIT and GIT2, as of July 2023
[3]https://huggingface.co/openai/clip-vit-base-patch16
[4]https://github.com/showlab/all-in-one
[5]https://huggingface.co/microsoft/git-base

AIO provides public code, inclusive of sampling strategy, we report such baselines results direct using their code (inclusive of their hyperparameter settings) for both training and testing.

Learning-based Sampler We compare with two advanced learning-based samplers, IGV (Li et al., 2022c) and VCSR (Wei et al., 2023). Both methods construct two or more complement segment groups with contrastive property and jointly optimize the main network and sampler by minimizing auxiliary losses. In original implementation, both IGV and VCSR sample much more frames than the default input lengths of backbone ITMs ($|V|= 16$ in IGV and $|V|$=frames/clip$\times$clip $= 6 \times 4 = 24$ in VCSR) to the same value (1×3 for VCSR). Because enlarging the input size improves accuracy (see Section 5.1), for fair comparison we reset the sampling size when implementing the two methods on each backbone model.

4.3 Implementation Details

The details of MIF have been introduced in Section 3.2. In MDF, we use each model's inherent vision encoder to encode the sampled frames, and then calculate the cosine values between these vectors as the measure of frame similarity. A special case is that AIO does not have an independent visual encoder. Hence, we use ViT-B-16 (the same visual encoder as CLIP and GIT) as the "pseudo visual encoder", and following the same procedure to obtain the sampled frames in each video.

Model	MSVD	MSRVTT	TGIF
Base (Radford et al., 2021)	33.8	33.7	59.9
IGV (Li et al., 2022c)	34.8	34.1	61.9
VCSR (Wei et al., 2023)	34.6	34.5	61.6
MIF (Ours)	35.0	**35.4**	62.5
MDF (Ours)	**35.1**	35.2	**63.2**

Table 2: Experimental results on CLIP ($|V|= 3$) backbone and three datasets.

4.4 Results

Results on CLIP Table 2 shows the results over the three datasets. Both MIF and MDF acquire achieves significant improvement over original CLIP implementations (1.2$\sim$3.3%) and baselines that incorporate learning-based sampling methods. However, the performance gap between the sampling strategies is insignificant on both MSVD-QA and MSRVTT-QA, indicating that question awareness is unnecessary for performance.

Model	MSVD	MSRVTT	TGIF
GIT Backbone			
Base (Wang et al., 2022)	52.2	41.1	67.5
IGV (Li et al., 2022c)	53.2	41.5	68.1
VCSR (Wei et al., 2023)	52.7	41.6	68.6
MIF	54.5	**42.3**	69.9
MDF	**55.3**	42.0	**70.0**
AIO Backbone			
Base (Wang et al., 2023)	46.1	42.7	64.0
IGV (Li et al., 2022c)	46.3	43.3	64.7
VCSR (Wei et al., 2023)	46.4	43.0	64.5
MIF	46.7	**44.0**	65.9
MDF	**46.9**	43.8	**66.2**

Table 3: Test set results on MSVD, MSRVTT and TGIF. Best scores are bolded.

Model	Val	Test
Base (Wang et al., 2023)	47.1	45.9
IGV (Li et al., 2022c)	48.3	47.1
VCSR (Wei et al., 2023)	48.0	47.4
MIF (Ours)	48.5	**48.2**
MDF (Ours)	**48.8**	48.0

Table 4: Results on validation and test of the multi-choice NExT-QA dataset (5-choices per question).

Results on GIT and AIO. Table 3 and Table 4 display the results of GIT and AIO on four datasets. There are three key points to worth concerning. Firstly, compared to the original implementation results, both MIF and MDF can enhance the accuracy on all four datasets regardless of model architectures. This appearance matches the trend on CLIP, which demonstrates our proposed methods are broadly applicable to diverse datasets and models. Secondly, the increment in accuracy is higher on models with more sampled frames (6 in GIT and 3 in AIO), which implies that our proposed methods are possibly more effective when the input frames is longer. Lastly, we notice that the improvement on TGIF-Frame by MIF and MDF over VCSR is more drastic than the other two datasets. This outcome is somehow counter-intuitive since videos in TGIF-Frame are much shorter with fewer chance in switching scenes—by intuition the dataset should be more insensitive to the sampling variants.

5 Analysis

5.1 Impact of Input Frame Length

Recall that we fix all baselines' input frame lengths in all experiments. However, intuitively the number (length) of input frames should be regarded as a potential factor to the accuracy, since increasing the input frames equals to exposing larger amount of training data to the model. To see how this factor affects backbone models' performance and whether our proposed sampling methods can consistently enhance the accuracy when sampling more or fewer frames, we continue to fine-tune GIT on the MSRVTT–QA dataset with distinct frame lengths. The results of this set of experiments are plotted in Figure 6a. From the figure we firstly discover that as expected, after increasing the number of input frames, the accuracy scores become higher. Moreover, the accuracy of the proposed two sampling strategies MDF and MIF consistently surpasses the VCSR baseline, indicating that they can really locate those key frames in videos even after changing the input length.

5.2 Auto-generated Captions in MIF

In MIF, we invoke a captioning model and anticipate it to provide precise and informative annotations to each frame. Since intuitively, the question–answering matching judgement model can not probably differentiate nuance in two sentences if their pattern looks quite similar. However, the actual results are opposite to our expectation. Take our randomly selected video from MSVD-QA in Table 5 as an example, where Q1 and Q2 denote two questions "what does a small dog wildly play with?" and "what wildly plays with a ball?". First, it can be observed that the captions generated by the VLM looks similar to each other, in the format of " [*noun*] [*verb*] [*prep. phrase*]", suggesting that the captioner model tends to generate descriptions in a nearly fixed pattern. This outcome can be viewed as a syntactic bias during generation. Moreover, the sentence similarity among these captions confuse the scorer model—although Q1 and Q2 describe nearly the same scenario and thus should share some cue frames, the most essential frame (the $12th$ frame) is successfully captured for Q1 but discarded for Q2, as well as the second most important frame (the $3rd$ frame). Therefore, we believe that a captioning model that can provide diversified output and a robust scoring model that offers objective and fair ratings to question–answer pairs is necessary to guarantee sampling effectiveness, itself vulnerable to noise.

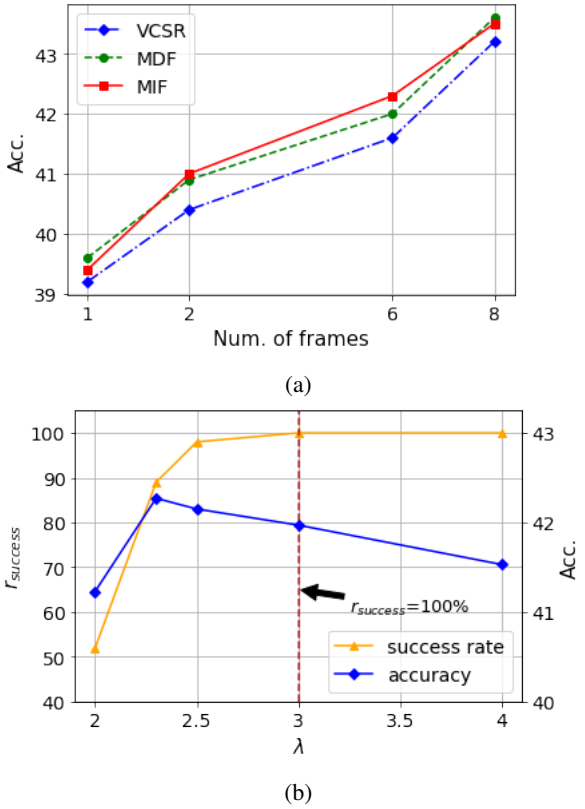


Figure 6: Performance compared to VCSR (Wei et al., 2023) under (a) different input lengths of frames in both MDF and MIF (b) varied separation factor λ in MDF on the MSRVTT-QA dataset by GIT.

FID	Caption	Q1	Q2
1	a puppy playing with toys.		
2	a white puppy playing with a toy.		
3	a white puppy with black eyes and a blue ball.	✓	
4	a puppy that is laying down on the floor.		
5	a puppy playing with a blue ball.		
6	a puppy that was found in a house.		✓
7	a puppy that is laying down on the floor.		
8	a puppy that is sitting on the floor.		✓
9	a puppy is sitting on the floor.	✓	✓
10	a white puppy sitting on a table.		✓
11	a white puppy laying on the floor.	✓	✓
12	a puppy playing with a blue ball.	✓	
13	a white dog standing on top of a floor.	✓	✓
14	a white dog walking on the floor.	✓	
15	a small white dog playing with a ball.		
16	a dog chewing on a toy in a cage.		

Table 5: Example frame captions and sampling results. "✓" marks frames chosen to constitute the input frame set along with the question in that column.

5.3 Sampling Interval in MDF

In MDF, we prevent the sampling frames from being excessively close by setting a hyperparameter λ and thus the search interval $W = L/(\lambda \cdot N)$. However, decreasing λ (enlarging the interval W) incurs more frequent failure for MDF to sample enough frames, and in this case some of the sampled frames may get too closed to degrade the target model's performance. In our experiments, we surprisingly found that such situations do not always happen. To delve into this phenomenon, we define the outcome where the collected K frames satisfy the interval requirements as "success" and otherwise as "failure". We test and plot the curve of success rate ($r_{success} = n_{success}/n_{total}$) and accuracy against λ on three datasets produced by GIT, as shown in Figure 6b. The horizontal axis denotes the hyperparameter λ that controls the minimal sampling interval. The figure shows that there is a critical point that failure will never happen if continuing to increase λ—we do not know the precise value but only to mark the minimal value among these settings that we can earn 100% success. Moreover, there is no strong correlation between the success rate and model performance, but a minimum interval should be reached to ensure a promising performance. The performance peak is achieved under a hybrid sampling strategy ($\lambda = 2.3, r_{success} = 79.1\%$).

6 Conclusion

In this paper, we focus on the frame sampling issue inhering in the task of video question–answering and propose two simple and effective methods—Most Implied Frames (MIF) and Most Dominant Frames (MDF). MIF streamlines a set of sampling methods in the textual space by projecting heterogeneous inputs (question and video) to a common space through pretrained ITMs. It then identifies frames with the highest matching scores generated from a scoring model. Based on the insights and analysis derived from MIF, we further propose Most Dominant Frames (MDF), which exploits a more concise, self-adaptive formulation for sampling. The success on these sampling strategies from CLIP to All-in-one demonstrates the broad applicability of our proposed methods across a spectrum of general scenarios.

Limitations

Despite the promising results gained from our methods, on a wider horizon we still note unaddressed limitations. First, due to the restriction of computation resource, we only evaluate our proposed methods on the video question answering task, and we do not have the opportunity to test on more emerged ITMs to further substantiate our methods' efficacy. Secondly, we do not try MIF-style methods on large language models like GPT-4. These areas may serve as future directions.

Acknowledgement

This project is supported by AcRF MoE Tier-2 grant (Project no. T2MOE2008 and Grantor reference no. MOE-T2EP20220-0017).

References

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, Andy Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karen Simonyan. 2022. Flamingo: a visual language model for few-shot learning. *ArXiv*, abs/2204.14198.

Yoshua Bengio, Aaron Courville, and Pascal Vincent. 2013. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, T. J. Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeff Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *ArXiv*, abs/2005.14165.

Shyamal Buch, Cristóbal Eyzaguirre, Adrien Gaidon, Jiajun Wu, Li Fei-Fei, and Juan Carlos Niebles. 2022. Revisiting the" video" in video-language understanding. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2917–2927.

Joao Carreira and Andrew Zisserman. 2017. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference*

on Computer Vision and Pattern Recognition, pages 6299–6308.

David L. Chen and William B. Dolan. 2011. Collecting highly parallel data for paraphrase evaluation. In *Annual Meeting of the Association for Computational Linguistics*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Tsu-Jui Fu, Linjie Li, Zhe Gan, Kevin Lin, William Yang Wang, Lijuan Wang, and Zicheng Liu. 2021. Violet: End-to-end video-language transformers with masked visual-token modeling. *arXiv preprint arXiv:2111.12681*.

Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. 2018. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6546–6555.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Deng Huang, Peihao Chen, Runhao Zeng, Qing Du, Mingkui Tan, and Chuang Gan. 2020. Location-aware graph convolutional networks for video question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 11021–11028.

Eric Jang, Shixiang Gu, and Ben Poole. 2016. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*.

Yunseok Jang, Yale Song, Chris Dongjoo Kim, Youngjae Yu, Youngjin Kim, and Gunhee Kim. 2019. Video Question Answering with Spatio-Temporal Reasoning. *IJCV*.

Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. 2021. Scaling up visual and vision-language representation learning with noisy text supervision. In *International Conference on Machine Learning*, pages 4904–4916. PMLR.

Jianwen Jiang, Ziqiang Chen, Haojie Lin, Xibin Zhao, and Yue Gao. 2020. Divide and conquer: Question-guided spatio-temporal contextual attention for video question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 11101–11108.

Wonjae Kim, Bokyung Son, and Ildoo Kim. 2021. Vilt: Vision-and-language transformer without convolution or region supervision. In *International Conference on Machine Learning*.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. Race: Large-scale reading comprehension dataset from examinations. *arXiv preprint arXiv:1704.04683*.

Jie Lei, Tamara L Berg, and Mohit Bansal. 2021a. Detecting moments and highlights in videos via natural language queries. *Advances in Neural Information Processing Systems*, 34:11846–11858.

Jie Lei, Linjie Li, Luowei Zhou, Zhe Gan, Tamara L. Berg, Mohit Bansal, and Jingjing Liu. 2021b. Less is more: Clipbert for video-and-language learning via sparse sampling. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7327–7337.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven C. H. Hoi. 2022a. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International Conference on Machine Learning*.

Junnan Li, Ramprasaath R. Selvaraju, Akhilesh Deepak Gotmare, Shafiq R. Joty, Caiming Xiong, and Steven C. H. Hoi. 2021. Align before fuse: Vision and language representation learning with momentum distillation. *ArXiv*, abs/2107.07651.

Xiujun Li, Xi Yin, Chunyuan Li, Xiaowei Hu, Pengchuan Zhang, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, Yejin Choi, and Jianfeng Gao. 2020. Oscar: Object-semantics aligned pre-training for vision-language tasks. In *European Conference on Computer Vision*.

Yicong Li, Xiang Wang, Junbin Xiao, and Tat-Seng Chua. 2022b. Equivariant and invariant grounding for video question answering. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 4714–4722.

Yicong Li, Xiang Wang, Junbin Xiao, Wei Ji, and Tat-Seng Chua. 2022c. Invariant grounding for video question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2928–2937.

Yuncheng Li, Yale Song, Liangliang Cao, Joel Tetreault, Larry Goldberg, Alejandro Jaimes, and Jiebo Luo. 2016. Tgif: A new dataset and benchmark on animated gif description. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4641–4650.

Fei Liu, Jing Liu, Weining Wang, and Hanqing Lu. 2021a. Hair: Hierarchical visual-semantic relational reasoning for video question answering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1698–1707.

Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. 2021b. Video swin transformer. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3192–3201.

Jeho Nam and Ahmed H Tewfik. 1999. Video abstract of video. In *1999 IEEE Third Workshop on Multimedia Signal Processing (Cat. No. 99TH8451)*, pages 117–122. IEEE.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. Ms marco: A human-generated machine reading comprehension dataset. *Advances in Neural Information Processing Systems*.

Bolin Ni, Houwen Peng, Minghao Chen, Songyang Zhang, Gaofeng Meng, Jianlong Fu, Shiming Xiang, and Haibin Ling. 2022. Expanding language-image pretrained models for general video recognition. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part IV*, pages 1–18. Springer.

Judea Pearl, Madelyn Glymour, and Nicholas P Jewell. 2016. Causal inference in statistics: A primer. 2016. *Internet resource*.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for squad. *arXiv preprint arXiv:1806.03822*.

Hanoona Rasheed, Muhammad Uzair Khattak, Muhammad Maaz, Salman Khan, and Fahad Shahbaz Khan. 2022. Fine-tuned clip models are efficient video learners. *arXiv preprint arXiv:2212.03640*.

Siva Reddy, Danqi Chen, and Christopher D Manning. 2019. Coqa: A conversational question answering challenge. *Transactions of the Association for Computational Linguistics*, 7:249–266.

Paul Hongsuck Seo, Arsha Nagrani, and Cordelia Schmid. 2020. Look before you speak: Visually contextualized utterances. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16872–16882.

Behzad Shahraray. 1995. Scene change detection and content-based sampling of video sequences. In *Digital Video Compression: Algorithms and Technologies 1995*, volume 2419, pages 2–13. SPIE.

Maria Tsimpoukelli, Jacob Menick, Serkan Cabi, S. M. Ali Eslami, Oriol Vinyals, and Felix Hill. 2021. Multimodal few-shot learning with frozen language models. In *Neural Information Processing Systems*.

Alex Jinpeng Wang, Yixiao Ge, Rui Yan, Ge Yuying, Xudong Lin, Guanyu Cai, Jianping Wu, Ying Shan, Xiaohu Qie, and Mike Zheng Shou. 2023. All in one: Exploring unified video-language pre-training. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Jianfeng Wang, Zhengyuan Yang, Xiaowei Hu, Linjie Li, Kevin Lin, Zhe Gan, Zicheng Liu, Ce Liu, and Lijuan Wang. 2022. Git: A generative image-to-text transformer for vision and language. *arXiv preprint arXiv:2205.14100*.

Zirui Wang, Jiahui Yu, Adams Wei Yu, Zihang Dai, Yulia Tsvetkov, and Yuan Cao. 2021. Simvlm: Simple visual language model pretraining with weak supervision. *ArXiv*, abs/2108.10904.

Yushen Wei, Yang Liu, Hong Yan, Guanbin Li, and Liang Lin. 2023. Visual causal scene refinement for video question answering. *arXiv preprint arXiv:2305.04224*.

Junbin Xiao, Angela Yao, Zhiyuan Liu, Yicong Li, Wei Ji, and Tat-Seng Chua. 2022. Video as conditional graph hierarchy for multi-granular question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 2804–2812.

Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. 2018. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *Proceedings of the European conference on computer vision (ECCV)*, pages 305–321.

Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin P. Murphy. 2017. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *European Conference on Computer Vision*.

Jun Xu, Tao Mei, Ting Yao, and Yong Rui. 2016a. 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.

Jun Xu, Tao Mei, Ting Yao, and Yong Rui. 2016b. Msr-vtt: A large video description dataset for bridging video and language. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5288–5296.

Antoine Yang, Antoine Miech, Josef Sivic, Ivan Laptev, and Cordelia Schmid. 2021. Just ask: Learning to answer questions from millions of narrated videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1686–1697.

Antoine Yang, Antoine Miech, Josef Sivic, Ivan Laptev, and Cordelia Schmid. 2022. Learning to answer visual questions from web videos. *arXiv preprint arXiv:2205.05019*.

Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. 2022. Coca: Contrastive captioners are image-text foundation models. *ArXiv*, abs/2205.01917.

Shoubin Yu, Jaemin Cho, Prateek Yadav, and Mohit Bansal. 2024. Self-chained image-language model for video localization and question answering. *Advances in Neural Information Processing Systems*, 36.

Rowan Zellers, Ximing Lu, Jack Hessel, Youngjae Yu, Jae Sung Park, Jize Cao, Ali Farhadi, and Yejin Choi. 2021. Merlot: Multimodal neural script knowledge models. *Advances in Neural Information Processing Systems*, 34:23634–23651.

Pengchuan Zhang, Xiujun Li, Xiaowei Hu, Jianwei Yang, Lei Zhang, Lijuan Wang, Yejin Choi, and Jianfeng Gao. 2021. Vinvl: Revisiting visual representations in vision-language models. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5575–5584.

A Implementation Details

To enforce a fair comparison, we run both training and testing stages for each VLM on a single NVIDIA RTX-A6000 GPU (except All-in-one because its implementation only has multi-GPU version, therefore we run it on 2 GPUs) while holding other hyperparameters and settings consistent with the default ones introduced in their original papers or codes (e.g., number of frames sampled per video, learning rate, training epoch, numerical precision in computation, etc). Gradient accumulation is applied to enable a large batch size (≥ 512) required in the fine-tuning process. To further reduce the computational complexity, all experiments are implemented with the pytorch Automatic Mixed Precision (AMP) [6] package. The checkpoints in our finetuning stage can all be found and downloaded from publicly available links.

B Baseline Models

We compare the results on the listed image–text pretrained models to other models in similar sizes that have (1) an image encoder inside but experience no or a different pretraining procedure (including the pretraining task selection and design, the goal function, datasets and annotation methods, etc) (Huang et al., 2020; Jiang et al., 2020; Liu et al., 2021a; Lei et al., 2021b). (2) a video encoder to tune during training time or merely use feature vectors extracted from pretrained video networks (I3D (Carreira and Zisserman, 2017), S3D (Xie et al., 2018)) (Xiao et al., 2022; Zellers et al., 2021; Yang et al., 2021; Fu et al., 2021). For baselines that work as our backbone network and finetuning starting point, we report our reproducing results as a more accurate benchmark, since we found many of these results are distinct from those reported in the original paper owing to the disparity in implementation environments.

Particularly, since we do not find any details introduced in the paper or official implementations online regarding the sampling strategies in GIT, and our implementation with uniform sampling in both training and testing can achieve comparable results as the reported ones (Wang et al., 2022) on 2 of 3 datasets, we treat this implementation as the reproduced results of GIT standalone.

C Evaluation Metrics

In all models, the sampled raw frames V' are resized to match the model-acceptable scales and then normalized. VLMs then take these frames as input and embed them into a sequence of vectors. Since the decoding mechanisms are different in these models, we illustrate them one by one:

In non-generative Video–LM (CLIP), the outputs from both modality encoders first pass through a transformer decoder layer and a classification layer:

$$\hat{A} = f(E_v, E_q) \tag{6}$$

In generative VLM (CLIP-Dec, GIT), the visual (from the visual encoder, like a prefix prepended to the text) and textual embeddings (from the embedding layer) constitute the input of the decoder. The decoder keeps generating the whole question and answer sequence in an auto-regressive manner:

$$P(Q, A|V, Q) = \sum_{t=1}^{n+l-1} \log P(y_{t+1}|y_1, y_2, ..., y_t, V) \tag{7}$$

In All-in-one, the model first generates answer predictions z_i for each frame. Then, these predictions are fused together by summation to form a consensus at the video level (Wang et al., 2023).

$$p = \frac{1}{S} \sum_{i=1}^{S} z_i \tag{8}$$

[6] https://pytorch.org/docs/stable/amp.html

D Speedup and Overhead Analysis

From video–text models to image–text ones. By adopting image–text VLMs (even without HDF5 as storage), we can obtain a $2.5 \sim 4\times$ acceleration during training and inference stage. Moreover the training can be completed with a single A6000 GPU (46 GB memory) for all image–text VLMs in our experiments (for all-in-one although it runs on 2 GPUs, the total memory usage can fit to a single GPU, i.e., much less than 46 GB), while video–text VLMs listed as our baselines (e.g., MERLOT (Zellers et al., 2021)) consume 4 same type of GPUs with the same batch size.

From *on-the-fly* sampling to offline sampling plus HDF5 I/O. Conventional approaches for image–encoder based VLMs to generate input frames directly read from raw videos and then sample frames among them *on-the-fly*, which consumes a large amount of storage and running time during training. As our proposed methods are *offline algorithms*, we can save all sampled frames for each video into a unified HDF5 file and meanwhile create a vid-to-id mapping file, (a.k.a. meta data) for the model to look up during its running time. HDF5 (Hierarchical Data Format) is a file format designed to store and organize large amounts of data by creating a set of "datasets", and to address current and anticipated requirements of modern systems. The contents saved in an HDF5 file can be mapped to RAM for fast loading during training, which greatly reduces the time needed for model training.

As a direct comparison, in our implementation of All-in-one, a $2.5 \sim 2.9\times$ speed-up during training stage is recorded when using HDF5 to substitute original reading from video-files and then sampling *on-the-fly*. For GIT and CLIP, this kind of comparison is infeasible since the training time can not be found neither in their papers nor replicated by our implementations (since we do not find open-sourced code for them on these video–QA datasets, the replication of their results also adopts the HDF5 I/O).

Removal of Redundant Sampling. Although the sampling process in the preprocessing stage produces additional overhead, we further highlight that the sampling process has to be run only **once per dataset** even for two different models if they consume the same number of frames as input. This feature further reduces the consumption of redundant computational power compared to those *on-the-fly* sampling methods since they need to recalculated the duplicated sample process during every tuning stages, not to mention that the HDF5 file can be shared online with potential users and researchers to download.

Case Study We take the experiment using All-in-one on TGIF-QA as an example. If using *on-the-fly* uniform sampling, the training time per epoch is 52 min and the model takes 15 epoches to converge (780 min in total). As comparison, after applying our sampling methods, the training time per epoch reduces to 18 min per epoch (270 min in total) while the additional overhead to generate the .h5 file is 3 hour (180 min). The total time combining sampling and training and is $270 + 180 = 450$ min, much shorter than the implementation with on-the-fly sampling.

E Dataset Statistics

We list the specifications of the datasets used in our evaluation process in Table 6.

Item	Split	MSVD	MSRVTT	TGIF	NExT
#Video	Train	1,200	6,513	37,089	3,870
	Dev	250	497	-	570
	Test	520	2,990	9,219	1,000
#Q&A	Train	30,933	158,581	39,392	31,173
	Dev	6,415	12,278	-	4,682
	Test	13,157	72,821	13,691	16,189

Table 6: Statistics of the four QA datasets evaluated in this paper. The split row lists the number of corresponding items in train/dev/test set. Note TGIF-QA does not have a validation set.

F Hyperparameter Search

In MDF, we run experiments on the sampled datasets with $\alpha \in \{2.3, 2.5, 2.7\}$. In MIF, we first uniformly pre-sample 16 frames in all experiments, then we calculate question–caption matching score based on these sampled frames. For all other hyperparameters (batch size, vocabulary size, learning rate, etc), we keep them same as original setting from their blogs or papers (for CLIP we adopt the same setting as GIT).