Data Scaling Isn't Enough: Towards Improving Compositional Reasoning in Video-Language Models

Kibum Kim
KAIST
kb.kim@kaist.ac.kr

Kyle Min*
Oracle
kyle.min@oracle.com

Chanyoung Park†
KAIST
cy.park@kaist.ac.kr

Abstract

Recent research in Video-Language Models has primarily focused on developing Video Foundation Models (ViFMs) that achieve strong zero-shot performance across various downstream tasks by scaling video-text pair datasets. Meanwhile, the compositional reasoning abilities of ViFMs have gained increasing attention, leading to a critical question: *Does scaling video-text pairs consistently enhance compositional reasoning?* Based on our finding that simply increasing the dataset size does not necessarily improve compositional reasoning, we explore whether compositional reasoning can be enhanced using a small, high-quality dataset instead of relying on dataset scaling. To this end, we focus on video scene graph (VidSG) datasets, which provide rich, structured relational information, and propose SGCR-Vid, a method designed to effectively leverage this information. Specifically, SGCR-Vid consists of two branches: the Text-based Scene Graph branch, which converts VidSG into text format and generates negative samples for fine-grained understanding; and the Visual-based Scene Graph branch, which incorporates structured visual relational information into video embeddings. To evaluate the effectiveness of SGCR-Vid, we apply it to two state-of-the-art ViFMs (ViCLIP and InternVideo2), demonstrating significant performance improvements on compositional reasoning benchmarks (VELOCITI and VideoCon), using less than 0.5% of the pretraining data scale. Our results show that compositional reasoning can be effectively enhanced using an extremely small-scale dataset, while also maintaining competitive performance on downstream tasks, validating the generalizability of our framework.

1 Introduction

Video-Language Models, such as ViCLIP [1] and VideoCLIP [2], which map video and text into a shared representation space, have advanced rapidly in recent years [3, 4, 5, 6]. Especially, there has been a surge of interest in the development of Video Foundation Models (ViFMs)[2] that have demonstrated strong zero-shot performance across various video-language tasks, including video-text retrieval [3, 6], action recognition [4, 7], and video question answering [8].

To enhance the generalizability of ViFMs, numerous studies [1, 7, 9, 8] have focused on expanding video-text pair datasets to web-scale sizes, inspired by the success achieved in the image domain [10]. For example, InternVideo [7] introduced approximately 13M video-text pairs, marking an early milestone. UMT [11] doubled this number to 26M pairs, followed by ViCLIP [1], which increased it to 200M pairs. More recently, VideoPrism [9] and InternVideo2 [8] further scaled the dataset to over

*Work partially done while at Intel Labs. †Corresponding author

[2]We focus on ViFMs trained with a contrastive learning objective, designed for discrimintative tasks, rather than on Multimodal Large Language Models that are tailored for generative tasks.

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: Efficient Reasoning.

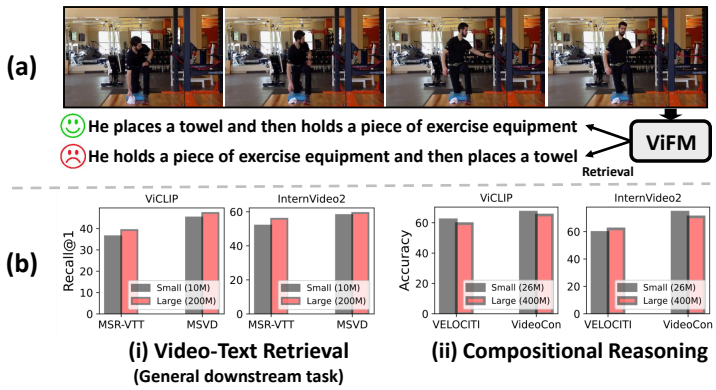

Figure 1: (a) An example of compositional reasoning in a Video Foundation Model (ViFM). (b) Performance comparison across different data scales for two ViFMs (i.e., ViCLIP [1] and InternVideo2 [8]), evaluated on a downstream task and compositional reasoning benchmarks.

400M pairs. These expansions have consistently led to performance improvements across general downstream tasks, such as video-text retrieval and action recognition.

Another line of research on ViFMs [12, 13, 14, 15] has emphasized the importance of *compositional reasoning*—the ability to understand complex, structured concepts such as inter-object relationships (e.g., action and spatial relationships) and their temporal relationships within open-world videos. As illustrated in Figure 1(a), compositional reasoning goes beyond simply recognizing actions like placing a towel or holding a piece of exercise equipment; it requires understanding the temporal relationships between these actions. This capability is essential for real-world applications that demand advanced scene understanding [16]. Despite its importance, several studies [15, 12, 13, 17, 18] have demonstrated that existing ViFMs struggle with compositional reasoning. For example, ICSVR [15] revealed object-level bias in ViFMs by systematically evaluating their performance on syntactic, relation, and object-level reasoning tasks. Recently, VELOCITI [12] introduced benchmark datasets specifically designed to assess compositional reasoning in ViFMs and found that their performance was often close to random guessing. Although the importance of this issue is increasingly recognized, effective methodological approaches to address it remain limited.

Building on the aforementioned studies on dataset scaling and compositional reasoning, we pose the following research question: *Does scaling up the video-text pair datasets consistently enhance the compositional reasoning ability of ViFMs?* To investigate this, we evaluate two state-of-the-art ViFMs (i.e., ViCLIP [1] and InternVideo2 [8]) trained at different data scales[3] on (i) a general downstream task (video–text retrieval task using MSRVTT [19] and MSVD [20]) and (ii) compositional reasoning benchmarks (VELOCITI [12] and VideoCon [14]). As shown in Figure 1(b), we observe that scaling the dataset size consistently boosts performance on the general downstream task, whereas it does not lead to consistent improvements in compositional reasoning. This observation suggests that while scaling helps ViFMs learn a broader vocabulary of objects and actions, thereby improving generalizability, it does not inherently improve their ability to reason over fine-grained, structured concepts. We hypothesize that this is because compositional reasoning requires a high-level, complex understanding beyond simple video-text matching. Nevertheless, as video-text pair datasets are scaled, they often contain an increasing number of repetitive and low-complexity associations, which can cause models to overfit to frequent and simplistic patterns, thereby hindering them from performing complex reasoning.

Motivated by these insights, in this paper, we aim to explore whether compositional reasoning in ViFMs can be enhanced by leveraging a small amount of high-quality data that explicitly captures complex inter-object relationships, rather than simply scaling up low-complexity video-text pair datasets. To this end, we employ video scene graph (VidSG) datasets: although these datasets are limited in scale due to the difficulty and high cost of annotation, they provide structured and detailed representations of inter-object relationships and their temporal dynamics. Specifically, we propose

[3]Note that each model trained on a smaller data scale used a subset of the pretraining dataset that was used to train the corresponding model on a larger data scale, e.g., ViCLIP-10M used a 10M subset of the 200M training data from ViCLIP-200M.

Scene **G**raph-based **C**ompositional **R**easoning for **Vid**eo Foundation Models (SGCR-Vid), a method designed to enhance the compositional reasoning capabilities of ViFMs by effectively leveraging rich relational[4] information. SGCR-Vid consists of two main branches: 1) the Text-based Scene Graph (TSG) branch, which converts video scene graphs into natural language and generates negative samples to help ViFMs understand structured relational information at a fine-grained level. 2) the Visual-based Scene Graph (VSG) branch, which enriches video embeddings by integrating structured visual relational information from VidSGs.

To evaluate the effectiveness of SGCR-Vid, we apply it to two state-of-the-art ViFMs (ViCLIP and InternVideo2). Our results demonstrate that SGCR-Vid significantly improves performance on compositional reasoning benchmarks (VELOCITI [12] and VideoCon [14]), while merely using less than 0.5% (i.e., approximately 41K samples) of the size of the training data used for pretraining. This highlights the potential of enhancing compositional reasoning with a small, high-quality dataset. Moreover, SGCR-Vid maintains competitive performance on general downstream tasks, confirming that it enhances compositional reasoning without sacrificing the generalizability of existing ViFMs.

Our contributions can be summarized as follows: 1) We identify that while scaling video-text pair datasets can improve the performance of ViFMs on general downstream tasks, it does not necessarily improve their compositional reasoning abilities. 2) We propose SGCR-Vid, a method that effectively leverages a small, high-quality VidSG dataset to capture rich, structured relational information. 3) We achieve significant improvements in compositional reasoning performance on VELOCITI and VideoCon using less than 0.5% of the pretraining data scale, without compromising generalizability.

2 Related Works

Video Foundation Models. Video Foundation Models (ViFMs) have attracted significant attention due to their remarkable zero-shot performance across multiple video-language tasks. To improve the generalizability of ViFMs, they have focused on collecting and scaling the video-text pair datasets, with which the video-language models are trained. Specifically, InternVideo [7] was trained on 13M pairs, OmniVL [21] on 17M pairs, and LAVENDER [22] on 30M pairs. As the datasets grew in size, UMT [11] collected 26M pairs, VideoCLIP [2] leveraged HowTo100M [23] with 100M pairs, and VIOLET [24] further extended with the YT-Temporal [25] dataset, reaching a total of 185M pairs. In pursuit of web-scale datasets, ViCLIP [1] released the InternVid dataset with 200M pairs for model training. More recently, VideoPrism [9] and InternVideo2 [8] have scaled their datasets over 400M pairs. Despite these advancements, existing studies have mainly focused on improving downstream task performance through data scaling, while compositional reasoning has often been overlooked. In this paper, we highlight the limitations of relying solely on data scaling to enhance compositional reasoning and investigate the potential of improving it with a small amount of high-quality data.

Compositional Reasoning. Compositional reasoning refers to the ability to understand structured concepts in language, as well as their alignment to the visual scenes. In the image domain, substantial efforts [26, 27, 28, 29, 30, 31, 32, 33] have been made to understand the inter-object relationships, object attributes, and object states. Among these, SGVL [28], similar to our approach, leveraged a scene graph dataset (i.e., Visual Genome [34]) to improve compositional reasoning. A key distinction of our work is its focus on the video domain, while SGVL focuses on the image domain. This distinction presents a limitation when adopting SGVL to the video domain, as video scene graphs differ fundamentally from image scene graphs due to the inclusion of the additional time dimension.

On the other hand, research in the video domain has mainly focused on providing benchmark datasets to evaluate the compositional reasoning of existing ViFMs, rather than on developing methods to improve it. VITATECS [17] and VideoComp [13] generate counterfactual descriptions based on the temporal aspect of videos to evaluate ViFMs. Meanwhile, ICSVR [15] releases datasets for evaluating syntactic, relational, and object-level reasoning. VideoCon [14] provides a dataset that evaluates compositional reasoning by considering various factors, including attributes, actions, and the temporality of actions. VELOCITI [12] evaluates compositional reasoning by segmenting the assessment into agent, action, and event chronology. While existing research has primarily concentrated on providing datasets for evaluating compositional reasoning and highlighting its deficiencies, we introduce a method that enhances compositional reasoning through the use of a small amount of high-quality video scene graph data.

[4]Hereafter, we use "inter-object relationships" and "relation" interchangeably.

3 Preliminary

In this section, we briefly review the video foundation models (ViFMs), which are generally trained with a contrastive learning objective [8, 1, 7, 9, 11] and the structure of video scene graph datasets.

Video Foundation Models (ViFMs). ViFMs are generally trained on an extensive video-text pair dataset $\{(V_i, C_i)\}_i^N$, where N is the number of pairs, V_i is a video and C_i is the corresponding text caption. ViFMs generally consist of video encoder h and text encoder f that extract the video embedding $\mathbf{v}_i$ (i.e., $\mathbf{v}_i = h(V_i) \in \mathbb{R}^d$) and text embedding $\mathbf{c}_i$ (i.e., $\mathbf{c}_i = f(C_i) \in \mathbb{R}^d$), respectively, with the same dimensionality d. For the video encoder, given a video $V = [I^1, I^2, ..., I^T]$, where I^t is the t-th frame, Vision Transformer (ViT) [35] is employed. Specifically, with the randomly sampled frames $[\hat{I}^1, \hat{I}^2, ..., \hat{I}^{T'}]$, where $\hat{I} \in V$ and $T' \leq T$, M non-overlapped patch tokens for each frame are extracted, followed by concatenation across all video frames. It is formally described as follows:

$$z = [z_{[\text{CLS}]}, P^1, P^2, ..., P^{T'}], \tag{1}$$

where $z_{[\text{CLS}]} \in \mathbb{R}^{d_p}$ is a CLS token, $P^t \in \mathbb{R}^{M \times d_p}$ is a set of patch tokens added with temporal positional embedding for the t-th frame, d_p is the feature dimension for a patch token, and [,] is the concatenation operation. After feed-forwarding z into a transformer and projecting the output to a low-dimensional space d, the final embedding with CLS token is utilized as the video embedding $\mathbf{v}$. Similarly, for the text encoder, a CLS token is prepended to the text C, followed by feed-forwarding to the transformer and projecting to a low-dimensional space d, thereby obtaining the text embedding $\mathbf{c}$. The contrastive learning objective then updates trainable parameters of h and f as follows:

$$\mathcal{L}_{\text{con}} = \underbrace{-\frac{1}{N_B}\sum_{i=1}^{N_B}\log\frac{\exp(\mathbf{v}_i^\top, \mathbf{c}_i)/\tau}{\sum_{j=1}^{N_B}\exp(\mathbf{v}_i^\top, \mathbf{c}_j)/\tau}}_{\mathcal{L}_{V2T}} \underbrace{-\frac{1}{N_B}\sum_{i=1}^{N_B}\log\frac{\exp(\mathbf{v}_i, \mathbf{c}_i^\top)/\tau}{\sum_{j=1}^{N_B}\exp(\mathbf{v}_j, \mathbf{c}_i^\top)/\tau}}_{\mathcal{L}_{T2V}}, \tag{2}$$

where N_B is the training batch size, τ is the temperature parameter, $\mathcal{L}_{V2T}$ is the video-to-text matching loss, and $\mathcal{L}_{T2V}$ is the text-to-video matching loss.

Video Scene Graphs (VidSGs). A VidSG dataset captures the structured inter-object relationships within videos. Specifically, a video V contains multiple clip-level scene graphs (SGs), denoted as $\{(G_g)\}_{g=1}^{N_G}$, where N_G is the number of SGs. Each scene graph G_g contains a set of relational triplets $\{(s_j, p_j, o_j)\}_{j=1}^{N_{SG_g}}$, grounded within a continuous frame interval $[I^{k_g}, I^{e_g}]$, where k_g and e_g indicate the starting and ending frame indices of the clip, respectively. Here, the N_{SG_g} is the number of triplets in G_g. In each triplet, the subject s_j is associated with a class label $s_{j,c}$ and a bounding-box trajectory $\mathcal{T}_{s_j}$. Likewise, the object o_j is labeled with a class label $o_{j,c}$ and associated with a bounding-box trajectory $\mathcal{T}_{o_j}$. The trajectories $\mathcal{T}_{s_j}$ and $\mathcal{T}_{o_j}$ span the same frame interval $[I^{k_g}, I^{e_g}]$, specifying the temporal extent over which they appear. The predicate p_j represents the relationship between s_j and o_j over the frame interval $[I^{k_g}, I^{e_g}]$, and is associated with a class label $p_{j,c}$. It is important to note that various temporal relationships exist among SGs: they may partially overlap (e.g., G_3, G_5), be fully nested within one another (e.g., G_1, G_4), or have no overlap at all (e.g., G_1, G_2). Here, G_1 to G_5 refer to the example SGs shown in Figure 2.

4 Method: SGCR-Vid

Our goal is to improve the compositional reasoning of ViFMs by leveraging a small amount of structured VidSG data. To this end, we begin by grounding the semantics of the VidSG via VidSG-based Frame Sampling and Attribute-level Enhancement (Section 4.1). Next, we propose two branches: Text-based Scene Graph (TSG) branch, which converts the video scene graph structure into text format and generates negative samples, followed by contrastive learning (Section 4.2), and Visual-based Scene Graph (VSG) branch, which incorporates structured visual scene graph into video embedding to enhance the video encoder h (Section 4.3). Finally, we outline the training process for updating the learnable parameters of the ViFMs (Section 4.4). The overall framework is shown in Figure 2.

4.1 Grounding the Semantics of Video Scene Graphs

VidSGs differ from image-based scene graphs (e.g., Visual Genome) in that they include evolving relationships over time. Moreover, while image-based scene graph datasets typically provide object

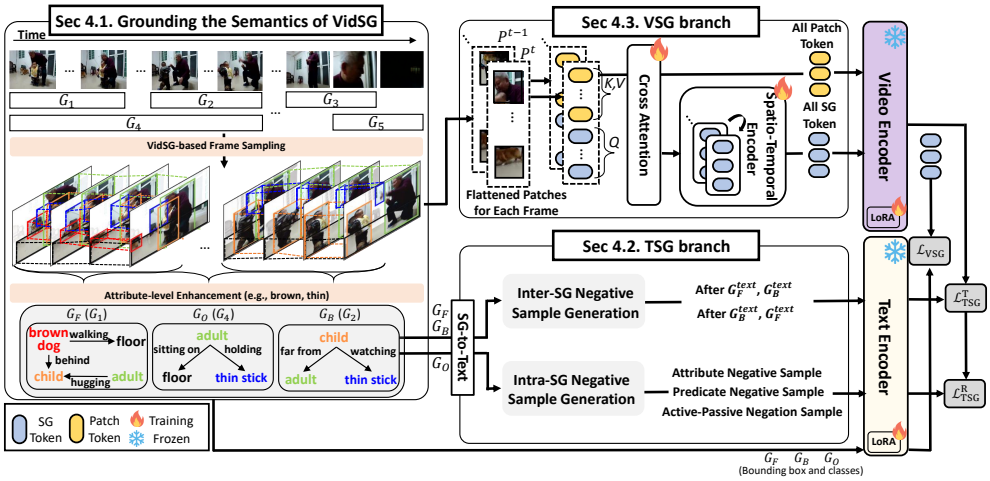

Figure 2: Overall framework of SGCR-Vid. Given the rich, structured VidSG dataset, we sample the frames and assign object attributes based on the VidSG-based Frame Sampling and Attribute-level Enhancement, respectively (Section 4.1). With the grounded VidSG, the TSG branch converts the VidSG into text and applies two negative sample generation (Section 4.2). Meanwhile, the VSG branch integrates visual structured relational information into the video embeddings (Section 4.3).

attributes, VidSG datasets lack this information due to the high cost of annotating all frames. To effectively ground the semantics of VidSG, we adopt two key strategies. The first is a VidSG-based Frame Sampling, designed to fully leverage the temporal dynamics inherent in VidSG. The second is Attribute-level Enhancement, which involves extracting object attributes within the video.

VidSG-based Frame Sampling. Frame sampling is crucial in the video domain to capture meaningful content within a limited set of frames [36, 37]. A naive approach that randomly samples frames within a frame interval $[I^k, I^q]$ of G may overlook temporal relationships between SGs and often results in selecting frames with less relational information, which in turn leads to a sub-optimal frame-sampling strategy. To address this, we meticulously sample informative frames while capturing the temporal relationships by leveraging the time indices (i.e., k) of multiple SGs. Specifically, we randomly choose two non-overlapping SGs, G_F and G_B ($k_F < k_B$), ensuring a meaningful temporal gap between them. Moreover, among the options for selecting two non-overlapping SGs from a pool of SGs, we ensure that there is a third SG, G_O, whose frame interval spans both G_F and G_B, providing key relational information for the TSG branch. From the overlapping frames of these three SGs, we uniformly sample frames $[\hat{I}^1, ..., \hat{I}^{\frac{T'}{2}}]$ from G_F, and the remaining frames $[\hat{I}^{\frac{T'}{2}+1}, ..., \hat{I}^{T'}]$ from G_B, thereby capturing the temporal relationships between G_F and G_B while ensuring that the sampled frames contain sufficient relational information. These sampled frames are then used as inputs to the TSG and VSG branches.

Attribute-level Enhancement. In general, blur or occlusion in videos often hinders accurate object recognition in the visual scene [38] and makes alignment with objects in the caption more challenging. Under such circumstances, object attributes play a crucial role in identifying objects in the visual scenes and aligning them between the scene and the caption. Therefore, we introduce a simple yet effective attribute-level enhancement strategy. Specifically, given that each sampled frame $\hat{I}^t$ is accompanied by ground-truth (GT) bounding box annotations, we employ a region-to-caption generator [39, 40] to obtain captions for each GT bounding box, followed by processing these captions using the SpaCy NLP toolkit [41] to extract object attributes. Detailed process can be found in Appendix A.1.

4.2 Text-based Scene Graph branch (TSG)

Existing studies on vision-language models [26, 27, 42] suggest that contrastive learning, by generating negative captions for a specific concept (e.g., object state), enables a fine-grained understanding of the altered concept, thereby improving the compositional reasoning. These studies typically train models on large datasets, such as DAC [27] with 3M images and VFC [42] with 381K videos. On the other hand, we focus on using a small amount of data (i.e., 9K VidSGs), which, despite its size,

contains structured and rich relational information. By applying a similar approach [27]— contrastive learning with generated negative captions— we aim to improve ViFM's understanding of relations and temporal order. Specifically, we convert the SGs into text and generate negative samples of a single SG (i.e., G_O) for *Intra-SG Negative Sample Generation* to understand inter-object relationships, and of two SGs (i.e., G_F and G_B) for *Inter-SG Negative Sample Generation* to understand temporal order. To convert the SGs into text, we flatten them into sequences of subject, predicate, and object. For example, G_O in Figure 2 is converted into text as "adult sitting on floor. adult holding thin stick", which we refer to as G_O^{Text}.

Intra-SG Negative Sample Generation (Relation). To understand the relations of G_O^{Text}, we generate negative samples from three perspectives: attribute, predicate, and active-passive relations. Specifically, for the *attribute* negative samples, $\tilde{G}_O^A$, we replace attributes (e.g., thin) with negative ones. A straightforward approach might randomly select attributes from predefined vocabularies for each case. However, randomly selected attributes (e.g., long) may not be negative within the context of the video, which could impair the model's understanding of the attribute. To address this, we leverage an LLM [43] to predefine obvious negative attributes for each case, such as defining thick as a negative counterpart to thin. For *predicate* negative samples, $\tilde{G}_O^P$, we swap predicates (e.g., sitting on and holding) with negative predicates. Similar to the attribute case, the LLM is used to predefine negative predicates to ensure validity. For the *active-passive* negative samples, $\tilde{G}_O^{AP}$, we swap the subject (e.g., adult) and object (e.g., floor) to help ViFM understand the distinction between active and passive roles within the scene. With negative samples from these three perspectives, we apply a contrastive learning objective using the positive sample, G_O^{Text}, to derive the loss $\mathcal{L}_{\text{TSG}}^{\text{R}}$. For the detailed loss function, please refer to Appendix A.3.

It is important to note that we do not use G_F^{Text} and G_B^{Text} for this negative sample generation, as the relations in these two SGs do not represent the overall relations of the selected frames. This would introduce potential noise, as the negative sample from G_F could correspond to G_B and vice versa.

Inter-SG Negative Sample Generation (Temporality). To understand the temporal order of relationships, we generate negative samples using two SGs, G_F^{Text} and G_B^{Text}. Specifically, positive samples are generated by applying temporal indicators such as after and before (e.g., After G_F^{Text}, G_B^{Text}) while negative samples are generated by reversing the order (e.g., After G_B^{Text}, G_F^{Text}). To support this, we design seven predefined templates that capture various temporal relationships, ensuring that ViFMs can understand these temporal relationships. Similar to the Intra-SG Negative Sample Generation, we apply a contrastive learning objective to the positive and negative samples, thereby deriving the $\mathcal{L}_{\text{TSG}}^{\text{T}}$ loss. Please refer to Appendix A.4 regarding the template of temporal relationships.

In summary, the total loss in the TSG branch is described as $\mathcal{L}_{\text{TSG}} = \mathcal{L}_{\text{TSG}}^{\text{R}} + \mathcal{L}_{\text{TSG}}^{\text{T}}$.

4.3 Visual-based Scene Graph branch (VSG)

The video encoder h is generally trained on large-scale video-text pair datasets to understand the context of many patch tokens $[P^1, P^2, ...P^{T'}]$, which allows the encoder to *implicitly* capture inter-object relationships. In contrast, we aim to *explicitly* encode the visual, structured inter-object relationships inherent in SGs by introducing a small number of tokens. Specifically, for each sampled frame, we prepend a small number of learnable SG tokens Z_{SG}^t to the patch tokens P^t. These SG tokens are trained to capture the structured relational information per se, and the complete sequence of tokens is defined as follows:

$$z = [z_{[\text{CLS}]}, Z_{SG}^1, P^1, ..., Z_{SG}^t, P^t, ..., Z_{SG}^{T'}, P^{T'}], \quad Z_{SG}^t = [z_{SG}^1, ..., z_{SG}^{S'}] \in \mathbb{R}^{S' \times d_p} \tag{3}$$

where Z_{SG}^t is the SG tokens for the t-th frame, and S' is the number of SG tokens per frame. Note that S' is much smaller than the number of patch tokens M (i.e., S'/M= 0.063).

Spatio-Temporal VidSG Encoding. Given that visual content changes across frames, we first apply Cross Attention independently for each frame, with Z_{SG}^t as the query and P^t as the key and value. This allows each Z_{SG}^t to be aware of the specific content of its corresponding frame. Then, to capture the spatio-temporal dependencies between SG tokens, we use two Transformers [44], each of which is responsible for handling spatial and temporal contexts. Let the multi-head attention be denoted as $\text{MHA}(Q, K, V)$, where the Q is the query, K is the key, and V is the value. With this notation, the

attention process is formally described as follows:

$$\dot{Z}^t_{SG} = \text{CrossAttn} = \text{MHA}(Z^t_{SG}, P^t, P^t) - \text{for being aware of visual content} \quad (4)$$

$$\ddot{Z}^t_{SG} = \text{MHA}(\dot{Z}^t_{SG}, \dot{Z}^t_{SG}, \dot{Z}^t_{SG}) - \text{for spatial context} \quad (5)$$

$$\dddot{Z}^t_{SG} = \text{MHA}(\ddot{Z}^t_{SG}, \ddot{Z}^{1:t-1}_{SG}, \ddot{Z}^{1:t-1}_{SG}) - \text{for temporal context,} \quad (6)$$

where $\ddot{Z}^{1:t-1}_{SG}$ represents all previous SG tokens for the t-th frame, processed by spatial transformer. Finally, the sequence $[z_{[\text{CLS}]}, \dddot{Z}^1_{SG}, P^1, ..., \dddot{Z}^{T'}_{SG}, P^{T'}]$ is fed forward to the video encoder h. The output from the SG tokens is denoted as $\bar{Z}^t_{SG}$, while the output from $z_{[\text{CLS}]}$, which aggregates all patch and SG tokens, is used as the video embedding $\mathbf{v}$.

Objective Function. We aim to encode G_F and G_O into $\bar{Z}^1_{SG}, ..., \bar{Z}^{\frac{T'}{2}}_{SG}$, while encoding G_B and G_O into $\bar{Z}^{\frac{T'}{2}+1}_{SG}, ..., \bar{Z}^{T'}_{SG}$[5]. At the same time, we aim to ensure that the class labels for entities (i.e., subject and object) and predicates within SGs are open-vocabulary, allowing for generalization across a wide range of classes. To achieve open-vocabulary functionality, we avoid using a classifier head with fixed classes [45]. Instead, we leverage the text embeddings of the entity classes within the batch, which are extracted from ViFM's text encoder g, and compute the similarity between these embeddings and $\bar{Z}^t_{SG}$, thus obtaining the likelihood of entity classes. A similar approach is applied to the predicates. For spatial context of entities, we use the GT bounding boxes for the subject and object, and the predictions of them are derived by multiplying $\bar{Z}^t_{SG} \in \mathbb{R}^{d_p}$ with the learnable matrices $W^b_s \in \mathbb{R}^{d_p \times 4}$ and $W^b_o \in \mathbb{R}^{d_p \times 4}$, respectively. Similar to prior works [45, 46], we employ the Hungarian matching algorithm [47] to match the GT triplets in the t-th frame with $\bar{Z}^t_{SG}$, thereby computing the matching loss $\mathcal{L}_{\text{VSG}}$. For details on the matching cost and the loss computation, please refer to Appendix A.5.

4.4 Training

To maintain the inherent knowledge of the existing ViFMs and avoid overfitting to the small amount of VidSG data, we include a subset of the video-text pair dataset (i.e., InternVid [1]) used during pretraining, and derive the loss (i.e., $\mathcal{L}_{\text{con}}$) using Equation 2. The model is trained end-to-end with the following loss function:

$$\mathcal{L} = \mathcal{L}_{\text{con}} + \mathcal{L}_{\text{TSG}} + \alpha \mathcal{L}_{\text{VSG}}, \quad (7)$$

where α is a hyperparameter that controls the weight. Furthermore, we apply the LoRA [48], an efficient fine-tuning strategy during training to reduce the computational resource requirements.

5 Experiment

We apply SGCR-Vid to two state of-the-art ViFMs: ViCLIP [1] and InternVideo2 [8]. In this section, we describe the experimental results to demonstrate the effectiveness of SGCR-Vid. Due to the space limitation, please refer to Appendix B.1 regarding the details of baselines.

5.1 Experimental Setup

Datasets. For **training**, we use the VidSG datasets, a combination of VidVRD [49] and VidOR [50], which have detailed manual annotations of object trajectories and relational information. The combination of the two datasets provides a total of 9K videos, with 95 entity labels and 167 predicate labels. Considering their dense triplet annotations and broad range of entity and predicate labels, this combined dataset is used as a small yet high-quality data. Regarding the video-text pair dataset, we use 32K samples selected from InternVid [1], which was used for the pretraining of ViCLIP and InternVideo2 that we aim to fine-tune. For **evaluation**, we use two benchmark datasets to evaluate the compositional reasoning of ViFMs: VELOCITI [12] and VideoCon [14], where the ViFM is required to select between a positive caption and a negative caption[6]. VELOCITI requires evaluation on three semantic concepts: agent, action, and event chronology. VideoCon is a dataset where human-generated negative captions are curated by perturbing one or more of the entities, actions, or temporal

[5]The VSG branch encodes all SGs within each frame to capture the full inter-object relationships.

[6]Given that this task involves a ViFM identifying the positive caption from the negative caption, the accuracy metric is applied to both datasets.

Table 2: Performance comparison on the VELOCITI dataset. The accuracy (%) metric is used. The subscript in the model indicates the scale of the data used for training.

Model	Agent Test			Action Test				Avg.
	Iden	Bind	Coref	Adv	Bind	Modif	Chrono	
Random	50.0	50.0	50.0	50.0	50.0	50.0	50.0	50.0
CLIP$_{400M}$ [10]	77.6	56.3	52.6	64.0	57.6	52.1	49.4	58.5
NegCLIP$_{2.9M}$ [26]	83.4	55.6	50.5	61.8	52.3	61.1	51.2	59.4
Owl-Con$_{165K}$ [14]	67.4	44.6	50.0	73.0	51.1	63.2	45.6	56.4
ViFi-CLIP$_{400K}$ [5]	82.3	58.7	54.6	63.0	59.3	60.5	49.8	61.2
CLIP-ViP$_{100M}$ [6]	75.3	52.4	55.7	70.2	53.5	51.2	48.5	58.1
ViCLIP$_{200M}$ [1]	83.0	55.7	52.9	59.4	59.4	58.8	53.3	59.4
InternVideo2$_{400M}$ [8]	85.8	53.7	51.9	69.4	55.8	62.8	54.6	62.0
ViCLIP$_{10M}$	84.9	58.3	53.4	66.8	58.3	60.3	51.9	62.0
+SGCR-Vid$_{41K\ (9K\ is\ VidSG)}$	**87.8**$_{+2.9}$	**59.6**$_{+1.3}$	51.7$_{-1.7}$	67.2$_{+0.4}$	**59.9**$_{+1.6}$	**65.8**$_{+5.5}$	53.5$_{+1.6}$	63.6
InternVideo2$_{26M}$	80.7	52.4	53.6	61.8	55.4	58.8	53.3	59.4
+SGCR-Vid$_{41K\ (9K\ is\ VidSG)}$	82.1$_{+1.4}$	54.4$_{+2.0}$	**56.2**$_{+2.6}$	**74.8**$_{+13.0}$	58.0$_{+2.6}$	62.6$_{+3.8}$	**57.9**$_{+4.6}$	**63.7**

Table 3: Performance comparison on the VideoCon dataset. The accuracy metric (%) is used.

Model	Accuracy (%)
Random	50.0
CLIP$_{400M}$	66.7
NegCLIP$_{2.9M}$	62.2
ViFi-CLIP$_{400K}$	63.6
CLIP-ViP$_{100M}$	68.0
ViCLIP$_{200M}$	65.1
InternVideo2$_{400M}$	71.1
ViCLIP$_{10M}$	66.0
+SGCR-Vid$_{41K}$	70.1$_{+4.1}$
InternVideo2$_{26M}$	73.8
+SGCR-Vid$_{41K}$	**74.6**$_{+0.8}$

relationships, and organizing them into a single set. Regarding the evaluation on general downstream tasks, we use three widely used datasets—MSRT-VTT [19], MSVD [20], and DeDiMo [51]—for the video-text retrieval task, as well as one dataset—Kinetics-400 (K400 [52])—for the action recognition task. Regarding the details of the dataset, please refer to Appendix B.2.

Implementation Details. We apply SGCR-Vid to two state-of-the-art ViFMs: ViCLIP [1] trained on a 10M pretraining dataset (i.e., InternVid [1]), and InternVideo2 [8], trained on a 25.5M pretraining dataset (i.e., InternVid and WebVid [53]). For further details on the implementation, please refer to Appendix B.3.

5.2 Quantitative Results of Compositional Reasoning

We evaluate SGCR-Vid's compositional reasoning capabilities on the VELOCITI and VideoCon datasets and compare its performance with the baselines.

VELOCITI: Table 2 shows the performance comparison between SGCR-Vid and the baselines on the VELOCITI dataset. We have the following observations. **1)** The performance of most baselines is generally subpar, except for Agent Iden, which is a relatively simple task of recognizing the entity. It indicates a need for further exploration of methodologies to enhance compositional reasoning. Especially, the performance of baselines on Chrono is nearly random, suggesting that existing ViFMs struggle with understanding temporal relationships. **2)** Despite the efforts of the baselines trying to scale the data, there is no consistent improvement in compositional reasoning performance with larger data. For example, even though ViCLIP$_{200M}$ and CLIP-ViP$_{100M}$ were trained on more than 100M data, their average performance is still lower compared to ViFi-CLIP$_{400K}$, which was trained on 400K data. This becomes even more apparent when comparing models trained on the same type of data but with different data scales, such as ViCLIP$_{10M}$ vs. ViCLIP$_{200M}$. **3)** SGCR-Vid significantly improves performance across various semantic concepts, demonstrating its effectiveness of leveraging the rich relational information even with a small amount of data. Beyond the simple expansion of video-text pair data with low-complexity patterns, it suggests that even an extremely small size of data (i.e., 9K), yet providing rich relational information, can boost the compositional reasoning capabilities of ViFMs. However, we observe a slight performance drop in Agent Coref for ViCLIP$_{10M}$+SGCR-Vid. According to VELOCITI, Agent Coref is the most challenging subcategory, requiring multi-step reasoning, which underscores the importance of high reasoning capabilities in the ViFM's text encoder. In this vein, ViCLIP, with its 12-layer text encoder (i.e., small model size), struggles to deliver such reasoning ability, thereby dropping the performance despite the utilization of rich relational information. In comparison, InternVideo2$_{26M}$ increases the improvement in Agent Coref, likely due to its 24-layer text encoder (i.e., large model size), which enables more complex reasoning and synergizes with rich relational information.

VideoCon: Table 3 shows the results on the VideoCon dataset. Similar to our observations in the VELOCITI dataset, scaling the data does not consistently increase the performance regarding the compositional reasoning, which further supports our claim discussed in Section 1. Furthermore, when applying SGCR-Vid to ViCLIP and InternVideo2, we observed consistent performance improvements, further confirming the effectiveness of our proposed method.

5.3 Quantitative Results on Downstream Tasks

To evaluate the performance of SGCR-Vid on general downstream tasks, we conducted experiments on video-text retrieval and action recognition using widely used benchmark datasets. As shown in Table 1, we have the following observations: **1)** Comparing models trained on the same dataset but with different scales (e.g., ViCLIP$_{200M}$ vs. ViCLIP$_{10M}$), training with the larger data scale generally leads to performance improvement on general downstream tasks, indicating that increasing data scale enhances the generalizability of ViFMs. **2)** The average performance of SGCR-Vid drops by less than 1% compared to its corresponding backbone model, demonstrating that SGCR-Vid remains competitive while maintaining the ViFM's generalizability.

Table 1: Performance comparison on downstream tasks under the zero-shot setting. Recall@1 is reported for all tasks.

Model	MSRVTT T2V	MSRVTT V2T	MSVD T2V	MSVD V2T	DeDiMo T2V	DeDiMo V2T	K400	Avg.
ViCLIP$_{200M}$	**39.3**	**39.5**	**47.3**	**70.0**	**17.1**	25.5	**59.8**	**42.6**
ViCLIP$_{10M}$	37.2	37.0	44.1	68.4	16.8	**25.7**	57.6	41.0
+SGCR-Vid$_{41K}$	38.4	37.0	44.5	66.3	16.8	24.4	56.6	40.6
InternVideo2$_{400M}$	**51.9**	**53.7**	**59.3**	83.1	**57.9**	**57.1**	**72.7**	**62.2**
InternVideo2$_{26M}$	49.6	52.1	58.9	83.9	51.0	54.5	66.9	59.6
+SGCR-Vid$_{41K}$	50.4	50.8	58.1	**84.2**	51.3	54.4	65.1	59.2

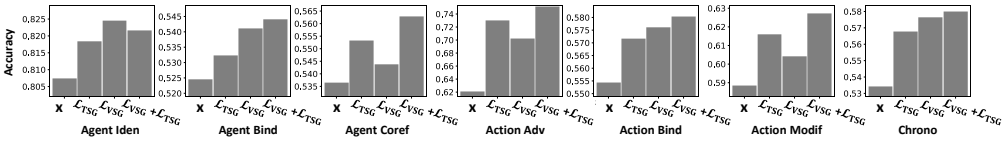

Figure 3: Ablation studies on VELOCITI using InternVideo2.

5.4 In-depth Analysis

Ablation Studies. In Figure 3, to understand the effectiveness of each branch of SGCR-Vid, we conduct ablation studies on the VELOCITI dataset using InternVideo2 as the baseline of SGCR-Vid. Note that the variant without any branch (i.e., X) corresponds to the vanilla InternVideo2. We observe that applying the TSG branch and VSG branch separately improves performance across all concepts. This indicates that each branch can effectively enhance compositional reasoning even with a small amount of rich relational data. Moreover, when the two branches are combined, the best performance is ultimately achieved, indicating that the two branches can be integrated synergistically. For Agent Iden concept, applying only the VSG branch results in higher performance since it already covers the basic entity recognition, leading to no improvement when TSG and VSG are combined.

Figure 4: (a) Correctly predicted case, (b) Incorrectly predicted case of InternVideo2$_{26M}$+SGCR-Vid in the VELOCITI dataset, showing attention scores for SG tokens and patch tokens across all layers.

Contribution of SG tokens. In Figure 4, to understand the contribution of SG tokens (Z_{SG}) compared to patch tokens (P) in enhancing compositional reasoning, we analyze the relative importance of each by measuring the amount of information the CLS token ($z_{[\text{CLS}]}$) in the video encoder h aggregates from each in two cases: (a) correctly predicated case, and (b) incorrectly predicted case. Specifically, for each case, we track the sum of attention scores that $z_{[\text{CLS}]}$ aggregates from patch tokens and SG tokens across all layers. We observe two main patterns: **1)** In both cases (a) and (b),

the contribution of SG tokens with structured information is high in lower layers (layers 2-12) while decreasing as the layer depth increases. This indicates that SG tokens play an important role in the early layers. **2)** In the correctly predicted case (a), SG tokens show a stronger contribution in the early layers compared to the incorrectly predicted case (b), indicating that SG tokens play a crucial role in compositional reasoning. Although noise in SG tokens may occasionally introduce excessive interference and lead to incorrect predictions, such cases are relatively rare, supported by the overall improvement in compositional reasoning performance on the VELOCITI and VideoCon datasets.

Data Usage Analysis. In Figure 5, to study the impact of data usage within the 41K dataset (consisting of 32K InternVid and 9K VidSG samples), we conducted experiments on the VELOCITI dataset based on the InternVideo2$_{26M}$+SGCR-Vid under two scenarios. First, we fixed the VidSG data at 9K while gradually increasing the proportion of InternVid data from 10% to 100%. Second, we fixed the InternVid data at 32K and gradually increased the portion of VidSG data. Our observations are as follows: **1)** In the first scenario (Figure 5(a)), we observe that performance increases as the amount of InternVid data increases, saturating near 0.8 to 1.0 of the 32K samples. This indicates that leveraging existing training data alongside VidSG data is crucial, and that approximately 32K InternVid samples are sufficient. **2)** In the second scenario (Figure 5(b)), we observe that performance generally improves as more VidSG data is added. This indicates that additional VidSG data could further enhance performance.

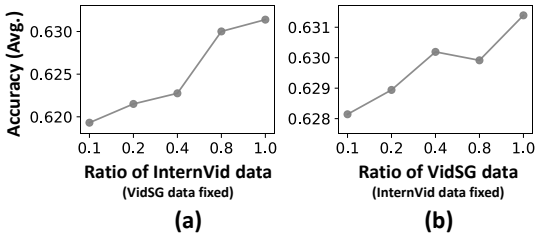

Figure 5: Performance across different data usage ratios.

6 Conclusion

In this paper, motivated by the observation that simply scaling data size does not consistently enhance compositional reasoning, we aim to explore whether a small yet high-quality dataset can enhance the compositional reasoning capabilities of ViFMs. To this end, we focus on the video scene graph (VidSG) dataset, which captures rich, structured relational information, and propose SGCR-Vid, a method designed to effectively enhance compositional reasoning in ViFMs. SGCR-Vid consists of two branches: the Text-based Scene Graph branch, which converts VidSG into text and generates negative samples for fine-grained understanding; and the Visual-based Scene Graph branch, which enriches video embeddings by integrating the rich, structured visual relational information from VidSG. Through extensive experiments on VELOCITI and VideoCon, we demonstrate significant improvements in compositional reasoning while maintaining competitive performance on downstream tasks, thereby preserving the model's generalizability.

7 Limitations and Future Works

Despite the fact that even a small amount of VidSG data can enhance compositional reasoning, we show in Section 5.4 of the main paper that larger quantities of VidSG data lead to greater performance gains. However, due to the high cost of manual annotation for the VidSG dataset, SGCR-Vid faces a limitation in data scalability. A potential solution is to explore weakly supervised video scene graph approaches [54, 55], which allow for the generation of VidSG data without the need for expensive human annotations. This approach could help mitigate the scalability limitations associated with VidSG data.

As future work, we plan to work on enhancing the compositional reasoning capabilities of Multimodal Large Language Models (MLLMs) [56, 57] designed for video understanding, as prior works [12, 17] have shown that current MLLMs still struggle with compositional reasoning. Building on our approach, a promising direction is to explore whether even a small amount of high-quality data can also effectively enhance the compositional reasoning abilities of MLLMs.

References

[1] Yi Wang, Yinan He, Yizhuo Li, Kunchang Li, Jiashuo Yu, Xin Ma, Xinhao Li, Guo Chen, Xinyuan Chen, Yaohui Wang, et al. Internvid: A large-scale video-text dataset for multimodal understanding and generation. *arXiv preprint arXiv:2307.06942*, 2023.

[2] Hu Xu, Gargi Ghosh, Po-Yao Huang, Dmytro Okhonko, Armen Aghajanyan, Florian Metze, Luke Zettlemoyer, and Christoph Feichtenhofer. Videoclip: Contrastive pre-training for zero-shot video-text understanding. *arXiv preprint arXiv:2109.14084*, 2021.

[3] Huaishao Luo, Lei Ji, Ming Zhong, Yang Chen, Wen Lei, Nan Duan, and Tianrui Li. Clip4clip: An empirical study of clip for end to end video clip retrieval. *arXiv preprint arXiv:2104.08860*, 2021.

[4] Ziyi Lin, Shijie Geng, Renrui Zhang, Peng Gao, Gerard De Melo, Xiaogang Wang, Jifeng Dai, Yu Qiao, and Hongsheng Li. Frozen clip models are efficient video learners. In *European Conference on Computer Vision*, pages 388–404. Springer, 2022.

[5] Hanoona Rasheed, Muhammad Uzair Khattak, Muhammad Maaz, Salman Khan, and Fahad Shahbaz Khan. Fine-tuned clip models are efficient video learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6545–6554, 2023.

[6] Hongwei Xue, Yuchong Sun, Bei Liu, Jianlong Fu, Ruihua Song, Houqiang Li, and Jiebo Luo. Clip-vip: Adapting pre-trained image-text model to video-language representation alignment. *arXiv preprint arXiv:2209.06430*, 2022.

[7] Yi Wang, Kunchang Li, Yizhuo Li, Yinan He, Bingkun Huang, Zhiyu Zhao, Hongjie Zhang, Jilan Xu, Yi Liu, Zun Wang, et al. Internvideo: General video foundation models via generative and discriminative learning. *arXiv preprint arXiv:2212.03191*, 2022.

[8] Yi Wang, Kunchang Li, Xinhao Li, Jiashuo Yu, Yinan He, Guo Chen, Baoqi Pei, Rongkun Zheng, Zun Wang, Yansong Shi, et al. Internvideo2: Scaling foundation models for multimodal video understanding. In *European Conference on Computer Vision*, pages 396–416. Springer, 2024.

[9] Long Zhao, Nitesh B Gundavarapu, Liangzhe Yuan, Hao Zhou, Shen Yan, Jennifer J Sun, Luke Friedman, Rui Qian, Tobias Weyand, Yue Zhao, et al. Videoprism: A foundational visual encoder for video understanding. *arXiv preprint arXiv:2402.13217*, 2024.

[10] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021.

[11] Kunchang Li, Yali Wang, Yizhuo Li, Yi Wang, Yinan He, Limin Wang, and Yu Qiao. Unmasked teacher: Towards training-efficient video foundation models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19948–19960, 2023.

[12] Darshana Saravanan, Varun Gupta, Darshan Singh, Zeeshan Khan, Vineet Gandhi, and Makarand Tapaswi. Velociti: Benchmarking video-language compositional reasoning with strict entailment, 2025.

[13] Dahun Kim, AJ Piergiovanni, Ganesh Mallya, and Anelia Angelova. Videocomp: Advancing fine-grained compositional and temporal alignment in video-text models. *arXiv preprint arXiv:2504.03970*, 2025.

[14] Hritik Bansal, Yonatan Bitton, Idan Szpektor, Kai-Wei Chang, and Aditya Grover. Videocon: Robust video-language alignment via contrast captions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13927–13937, 2024.

[15] Avinash Madasu and Vasudev Lal. Icsvr: Investigating compositional and syntactic understanding in video retrieval models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1733–1743, 2024.

[16] Haiyi Qiu, Minghe Gao, Long Qian, Kaihang Pan, Qifan Yu, Juncheng Li, Wenjie Wang, Siliang Tang, Yueting Zhuang, and Tat-Seng Chua. Step: Enhancing video-llms' compositional reasoning by spatio-temporal graph-guided self-training. *arXiv preprint arXiv:2412.00161*, 2024.

[17] Shicheng Li, Lei Li, Yi Liu, Shuhuai Ren, Yuanxin Liu, Rundong Gao, Xu Sun, and Lu Hou. Vitatecs: A diagnostic dataset for temporal concept understanding of video-language models. In *European Conference on Computer Vision*, pages 331–348. Springer, 2024.

[18] Ilker Kesen, Andrea Pedrotti, Mustafa Dogan, Michele Cafagna, Emre Can Acikgoz, Letitia Parcalabescu, Iacer Calixto, Anette Frank, Albert Gatt, Aykut Erdem, et al. Vilma: A zero-shot benchmark for linguistic and temporal grounding in video-language models. *arXiv preprint arXiv:2311.07022*, 2023.

[19] 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.

[20] David Chen and William B Dolan. Collecting highly parallel data for paraphrase evaluation. In *Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies*, pages 190–200, 2011.

[21] Junke Wang, Dongdong Chen, Zuxuan Wu, Chong Luo, Luowei Zhou, Yucheng Zhao, Yujia Xie, Ce Liu, Yu-Gang Jiang, and Lu Yuan. Omnivl: One foundation model for image-language and video-language tasks. *Advances in neural information processing systems*, 35:5696–5710, 2022.

[22] Linjie Li, Zhe Gan, Kevin Lin, Chung-Ching Lin, Zicheng Liu, Ce Liu, and Lijuan Wang. Lavender: Unifying video-language understanding as masked language modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23119–23129, 2023.

[23] Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2630–2640, 2019.

[24] Tsu-Jui Fu, Linjie Li, Zhe Gan, Kevin Lin, William Yang Wang, Lijuan Wang, and Zicheng Liu. Violet: End-to-end video-language transformers with masked visual-token modeling. *arXiv preprint chen2011collectingarXiv:2111.12681*, 2021.

[25] Rowan Zellers, Ximing Lu, Jack Hessel, Youngjae Yu, Jae Sung Park, Jize Cao, Ali Farhadi, and Yejin Choi. Merlot: Multimodal neural script knowledge models. *Advances in neural information processing systems*, 34:23634–23651, 2021.

[26] Mert Yuksekgonul, Federico Bianchi, Pratyusha Kalluri, Dan Jurafsky, and James Zou. When and why vision-language models behave like bags-of-words, and what to do about it? *arXiv preprint arXiv:2210.01936*, 2022.

[27] Sivan Doveh, Assaf Arbelle, Sivan Harary, Roei Herzig, Donghyun Kim, Paola Cascante-Bonilla, Amit Alfassy, Rameswar Panda, Raja Giryes, Rogerio Feris, et al. Dense and aligned captions (dac) promote compositional reasoning in vl models. *Advances in Neural Information Processing Systems*, 36:76137–76150, 2023.

[28] Roei Herzig, Alon Mendelson, Leonid Karlinsky, Assaf Arbelle, Rogerio Feris, Trevor Darrell, and Amir Globerson. Incorporating structured representations into pretrained vision & language models using scene graphs. *arXiv preprint arXiv:2305.06343*, 2023.

[29] Cheng-Yu Hsieh, Jieyu Zhang, Zixian Ma, Aniruddha Kembhavi, and Ranjay Krishna. Sugarcrepe: Fixing hackable benchmarks for vision-language compositionality. *Advances in neural information processing systems*, 36:31096–31116, 2023.

[30] Le Zhang, Rabiul Awal, and Aishwarya Agrawal. Contrasting intra-modal and ranking cross-modal hard negatives to enhance visio-linguistic compositional understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13774–13784, 2024.

[31] Hadi Wazni, Kin Ian Lo, and Mehrnoosh Sadrzadeh. Verbclip: Improving verb understanding in vision-language models with compositional structures. In *Proceedings of the 3rd Workshop on Advances in Language and Vision Research (ALVR)*, pages 195–201, 2024.

[32] Sri Harsha Dumpala, Aman Jaiswal, Chandramouli Shama Sastry, Evangelos Milios, Sageev Oore, and Hassan Sajjad. Sugarcrepe++ dataset: Vision-language model sensitivity to semantic and lexical alterations. *Advances in Neural Information Processing Systems*, 37:17972–18018, 2024.

[33] Ugur Sahin, Hang Li, Qadeer Khan, Daniel Cremers, and Volker Tresp. Enhancing multimodal compositional reasoning of visual language models with generative negative mining. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 5563–5573, 2024.

[34] 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*, 123:32–73, 2017.

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

[36] Sullam Jeoung, Goeric Huybrechts, Bhavana Ganesh, Aram Galstyan, and Sravan Bodapati. Adaptive video understanding agent: Enhancing efficiency with dynamic frame sampling and feedback-driven reasoning. *arXiv preprint arXiv:2410.20252*, 2024.

[37] K Chen, Y Li, Z Zhang, X Ren, and H Li. Sas video-qa: Self-adaptive sampling for efficient video question-answering. *arXiv preprint arXiv:2307.04192*, 5, 2023.

[38] Angtian Wang, Yihong Sun, Adam Kortylewski, and Alan Yuille. Robust object detection under occlusion with context-aware compositionalnets, 2020.

[39] Fangyi Chen, Han Zhang, Zhantao Yang, Hao Chen, Kai Hu, and Marios Savvides. Rtgen: Generating region-text pairs for open-vocabulary object detection. *arXiv preprint arXiv:2405.19854*, 2024.

[40] Jialian Wu, Jianfeng Wang, Zhengyuan Yang, Zhe Gan, Zicheng Liu, Junsong Yuan, and Lijuan Wang. Grit: A generative region-to-text transformer for object understanding. In *European Conference on Computer Vision*, pages 207–224. Springer, 2024.

[41] Matthew Honnibal, Ines Montani, Sofie Van Landeghem, Adriane Boyd, et al. spacy: Industrial-strength natural language processing in python. 2020.

[42] Liliane Momeni, Mathilde Caron, Arsha Nagrani, Andrew Zisserman, and Cordelia Schmid. Verbs in action: Improving verb understanding in video-language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15579–15591, 2023.

[43] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

[44] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[45] Hangjie Yuan, Jianwen Jiang, Samuel Albanie, Tao Feng, Ziyuan Huang, Dong Ni, and Mingqian Tang. Rlip: Relational language-image pre-training for human-object interaction detection. *Advances in Neural Information Processing Systems*, 35:37416–37431, 2022.

[46] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020.

[47] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

[48] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.

[49] Xindi Shang, Tongwei Ren, Jingfan Guo, Hanwang Zhang, and Tat-Seng Chua. Video visual relation detection. In *Proceedings of the 25th ACM international conference on Multimedia*, pages 1300–1308, 2017.

[50] Xindi Shang, Donglin Di, Junbin Xiao, Yu Cao, Xun Yang, and Tat-Seng Chua. Annotating objects and relations in user-generated videos. In *Proceedings of the 2019 on International Conference on Multimedia Retrieval*, pages 279–287, 2019.

[51] Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell. Localizing moments in video with natural language. In *Proceedings of the IEEE international conference on computer vision*, pages 5803–5812, 2017.

[52] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

[53] 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.

[54] Kibum Kim, Kanghoon Yoon, Yeonjun In, Jaehyeong Jeon, Jinyoung Moon, Donghyun Kim, and Chanyoung Park. Weakly supervised video scene graph generation via natural language supervision. In *The Thirteenth International Conference on Learning Representations*.

[55] Ziyue Wu, Junyu Gao, and Changsheng Xu. Weakly-supervised video scene graph generation via unbiased cross-modal learning. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 4574–4583, 2023.

[56] 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.

[57] 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. *arXiv preprint arXiv:2311.10122*, 2023.

[58] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023.

[59] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*, 2023.

[60] OpenAI. Gpt-4o: Large language model. `https://openai.com/research/gpt-4o`, 2023. Accessed: YYYY-MM-DD.

[61] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 658–666, 2019.

[62] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer vision–ECCV 2014: 13th European conference, zurich, Switzerland, September 6-12, 2014, proceedings, part v 13*, pages 740–755. Springer, 2014.

[63] Qinghao Ye, Haiyang Xu, Guohai Xu, Jiabo Ye, Ming Yan, Yiyang Zhou, Junyang Wang, Anwen Hu, Pengcheng Shi, Yaya Shi, et al. mplug-owl: Modularization empowers large language models with multimodality. *arXiv preprint arXiv:2304.14178*, 2023.

[64] Rohan Anil, Andrew M Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, et al. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*, 2023.

[65] Xin Wang, Jiawei Wu, Junkun Chen, Lei Li, Yuan-Fang Wang, and William Yang Wang. Vatex: A large-scale, high-quality multilingual dataset for video-and-language research. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4581–4591, 2019.

[66] Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell. Localizing moments in video with temporal language. *arXiv preprint arXiv:1809.01337*, 2018.

[67] Hongwei Xue, Tiankai Hang, Yanhong Zeng, Yuchong Sun, Bei Liu, Huan Yang, Jianlong Fu, and Baining Guo. Advancing high-resolution video-language representation with large-scale video transcriptions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5036–5045, 2022.

[68] Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 24185–24198, 2024.

[69] Limin Wang, Bingkun Huang, Zhiyu Zhao, Zhan Tong, Yinan He, Yi Wang, Yali Wang, and Yu Qiao. Videomae v2: Scaling video masked autoencoders with dual masking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14549–14560, 2023.

[70] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Supplementary Material

- Data Scaling Isn't Enough: Towards Improving Compositional Reasoning in Video-Language Models -

---

A SGCR-Vid **16**

A.1 Detailed Explanation of Attribute-level Enhancement 16

A.2 Prompt for Attribute and Predicate Negatives 16

A.3 Loss Function for the TSG branch 17

A.4 Template of Temporal Relationships 17

A.5 Further Explanation of the VSG branch 17

B Experiments **18**

B.1 Baselines 18

B.2 Details of Dataset 18

B.3 Additional Implementation Details 19

B.4 Hyperparameter Sensitivity 19

B.5 Additional Qualitative Results for Contribution of SG Tokens 20

A SGCR-Vid

A.1 Detailed Explanation of Attribute-level Enhancement

In this section, we provide a detailed explanation of the Attribute-level Enhancement described in Section 4.1 of the main paper. For each frame's ground-truth bounding boxes, we use the GRiT [40] region-to-caption generator to obtain descriptive captions and process these captions using the SpaCy [41] toolkit to extract object attributes. However, we observe that attributes of the same object, extracted from different frames, often vary due to noise or inconsistencies in the captions generated by the region-to-caption generator. To mitigate this, we apply a majority voting scheme over the attributes of the same object across different frames, ensuring greater consistency. Although more advanced strategies—such as using multimodal large language models [58, 59]—could potentially yield richer attributes, they are computationally intensive and impractical for processing the large number of bounding boxes present across video frames. Hence, we adopt the proposed approach to balance effectiveness and efficiency.

A.2 Prompt for Attribute and Predicate Negatives

Extraction of Negative Attributes. In the Intra-SG Negative Sample Generation of the TSG branch (Section 4.2 of the main paper), we obtain obvious negative attributes of objects using an LLM [60]. Furthermore, to ensure clarity, we exclude ambiguous or abstract attributes (e.g., serious, handmade) that are difficult to infer from the visual scene. This leads us to manually define criteria across three categories (i.e., color, size, and position) and extract only the attributes belonging to these categories. For this purpose, we craft a prompt, into which the attribute of our interest is inserted at {QUERY}. The full prompt is as follows:

Your task is to generate a list of negation attributes for a given attribute associated with an object. The input attribute must belong to one of the following categories: color, size, or position. If it does not belong to any of these categories, return an empty list. Furthermore, ensure that all negations you provide come strictly from the same category as the given attribute. For example, if the input attribute is related to position, the negations should only include other position-related attributes.
Input: black
Output: ['red', 'white', 'green', 'purple', 'yellow', 'blue', 'bright', 'violet', 'pink', 'navy', 'gold', 'beige', 'bronze', 'orange', 'brown']
Input: white
Output: ['red', 'green', 'purple', 'yellow', 'blue', 'gray', 'dark', 'violet', 'pink', 'navy', 'gold', 'beige', 'bronze', 'orange', 'black', 'brown', 'silver']
Input: first
Output: []
Input: huge
Output: ['small', 'tiny', 'little', 'minor', 'compact', 'modest']
Input: extended
Output: []
Input: plastic
Output: []
Input: front
Output: ['back', 'rear', 'backside', 'behind']
Input: wild
Output: []
Input: {QUERY}
Output:

Extraction of Negative Predicates. Similarly, to obtain obvious negative predicates, we craft a prompt where the predicate of our interest is inserted into {QUERY}. The complete prompt is provided below.

Your task is to generate a list of negation predicates for a given predicate. Make sure that each negation predicate represents an obvious negation of the given predicate, meaning that it is impossible for both to occur in the visual scene.

Input: sit above
Output: sit below, sit beneath, sit under, stand
Input: open
Output: close, shut, seal, lock, fasten
Input: walk right
Output: walk left, move left, shift left
Input: hold
Output: release, drop, push away, put down, not hold
Input: shout at
Output: whisper to, speak softly to, is silent towards
Input: get off
Output: get on, stay on
Input: chase
Output: evade, flee from, run away from
Input: drive
Output: walk, stop, park, halt
Input: jump left
Output: jump right, move right, jump back
Input: creep beneath
Output: creep above, stay above, creep on
Input: {QUERY}
Output:

For both the generated negative attributes and predicates, we manually verified their validity to ensure they were clear and accurate.

A.3 Loss Function for the TSG branch

In Section 4.2 of the main paper, we apply contrastive learning in the Intra-SG Negative Sample Generation, using one positive sample (G_O^{Text}) and three negative samples ($\tilde{G}_O^A, \tilde{G}_O^P, \tilde{G}_O^{AP}$) to enable fine-grained semantic understanding of each concept. The loss function used is defined as follows:

$$\mathcal{L}_{\text{TSG}}^{\text{R}} = -\log \frac{\exp(\mathbf{v}^\top, f(G_O^{Text}))/\tau}{\exp(\mathbf{v}^\top, f(G_O^{Text}))/\tau + \sum_{j\in(A,P,AP)} \exp(\mathbf{v}^\top, f(\tilde{G}_O^j))/\tau} \tag{8}$$

A.4 Template of Temporal Relationships

In Section 4.2 of the main paper, for Inter-SG Negative Sample Generation, we designed seven templates to enable ViFMs to capture the temporal relationship between the two SGs, G_F^{Text} and G_B^{Text}. The complete set of templates is described below:

---

1. G_F^{Text} *before* G_B^{Text}.
2. G_B^{Text} *after* G_F^{Text}.
3. *First* G_F^{Text}, *then* G_B^{Text}.
4. *After* G_F^{Text}, G_B^{Text}.
5. G_F^{Text}, *and then* G_B^{Text}.
6. *Once* G_F^{Text}, G_B^{Text}.
7. *Before* G_B^{Text}, G_F^{Text}.

A.5 Further Explanation of the VSG branch

In this section, we provide a detailed explanation of the Hungarian matching algorithm [47] used to match the ground-truth (GT) triplets and $\bar{Z}_{SG}^t$, as well as the loss function for the VSG branch. Specifically, we define the cost for optimal assignment between two bipartite groups (i.e., GT triplets and $\bar{Z}_{SG}^t$) as the sum of the subject, object and predicate class similarity of $\bar{Z}_{SG}^t$ with respective to their GT classes, along with the GIoU [61] and L1 loss between the predicted and GT bounding boxes. After applying the Hungarian algorithm to each frame, each GT triplet in t-th frame is assigned to the

$\bar{Z}_{SG}^{t}$, and SG tokens that do not correspond to any triplet are padded. Finally, the SG loss is computed as follows:

$$\mathcal{L}_{\text{VSG}} = \lambda_1(\mathcal{L}_s + \mathcal{L}_o + \mathcal{L}_r) + \lambda_2\mathcal{L}_{l1} + \lambda_3\mathcal{L}_{GIoU}, \tag{9}$$

where $\mathcal{L}_{l1}$, $\mathcal{L}_{GIoU}$, $\mathcal{L}_s$, $\mathcal{L}_o$, and $\mathcal{L}_r$ denote the L1 loss, GIoU [61] loss, cross-entropy (CE) loss for subject and object classes, and CE loss for predicate classes, respectively. The λ terms are the weights for each loss component. Following [28], we set λ_1, and λ_2, λ_3 to 1/3, 5.0, and 2.0, respectively.

B Experiments

B.1 Baselines

We include seven baselines to compare with SGCR-Vid. **1) CLIP** [10] is an image-based vision-language model trained on 400M image-text pairs. To apply this model to video, following VELOCITI [12], we average the visual embeddings of sampled frames to obtain the video embedding $\mathbf{v}$. **2) NegCLIP** [26] is a CLIP variant fine-tuned to enhance compositional reasoning capabilities using negative captions generated from the COCO [62] dataset with 2.9M samples. Similar to CLIP, it averages the sampled frame embeddings for $\mathbf{v}$. **3) Owl-Con** [14] is a Multimodal Large Language Model [63] fine-tuned using negative captions generated by an LLM [64] from video-caption pair datasets (i.e., MSR-VTT [19], VaTeX [65], and TEMPO [66]) totaling 165K samples. **4) ViFi-CLIP** [5] is a video-adapted version of CLIP, fine-tuned on the Kinetics-400 [52] dataset with 400K samples to capture temporal and contextual information in videos. **5) CLIP-ViP** [6] enhances CLIP for video tasks by introducing learnable visual proxy prompt tokens that capture temporal information, trained on the HD-VILA-100M [67] dataset with 100M samples. **6) ViCLIP** [1] fine-tunes the CLIP model using the proposed video-text pair dataset, InternVid, with 200M samples. **7) InternVideo2** [8] is a large-scale video-language model whose video encoder is distilled from video expert models [68, 69] and trained on collected web-scale data (e.g., WebVid [53] and InternVid [1]) totaling 400M samples.

B.2 Details of Dataset

In this section, we describe the training and evaluation datasets in detail.

Training. We use the combination of two VidSG datasets: VidVRD [49] and VidOR [50]. VidVRD consists of 1K videos with an average of 10.1 triplets per frame. It includes 35 entity class labels and 132 predicate labels. VidOR consists of 8K videos with an average of 10.4 triplets per frame. It contains 80 entity class labels and 50 predicate labels. The combination of the two datasets provides a total of 9K videos, with 95 entity labels and 167 predicate labels. It is important to note that, given that 32K video-text pair samples (i.e., InternVid [1]) are used, we use a total of 41K, which is less than 0.5% of the data used during pretraining of ViCLIP and InternVideo2.

Evaluation. For the VELOCITI [12] dataset, three concepts—*agent*, *action*, and *event chronology*—are defined to evaluate different aspects of model understanding. The *agent* concept evaluates the ViFM's ability to distinguish the entities within a video, and is further categorized into three subtypes: Identification (Iden), Binding (Bind), and Co-reference (Coref). For the Iden subtype, negative captions are curated by randomly replacing the original entity with another. On the other hand, the Bind subtype replaces the entity with another one appearing in the same video, making it more challenging due to the increased contextual similarity. The Coref subtype evaluates the model's ability to identify entities through referential expressions (e.g., The person who is greeting a man wearing a black hat). The *action* concept focuses on recognizing the action that unfolds over time, and is divided into three subtypes: Adversarial (Adv), Binding (Bind), and Modifier (Modif). In the Adv subtype, the action in the positive caption is replaced with another that does not appear in the video. In the Bind subtype, the action is replaced with a different action that occurs within the same video. The Modif subtype alters the action modifier in the positive caption with another plausible, but incorrect, modifier. *Event chronology* (chron) evaluates the temporal relationships of events. For the VideoCon [14] dataset, following the split proposed in [14], we use a subset of temporally-challenging 290 samples, selected from a total of 570.

B.3 Additional Implementation Details

For ViCLIP, we follow its pretraining configuration, sampling 8 frames (T'=8), each with 196 patch tokens (M=196) and a feature dimension of 768 (d_p=768). The hyperparameters are set to use 12 SG tokens (S'=12), $\alpha = 0.1$, and a training batch size of 128. Likewise, for InternVideo2, we adopt its pretraining configuration, sampling 4 frames, each with 256 patch tokens and a feature dimension of 1,408. Due to its large model size of 1B parameters, we use a smaller batch size of 18, with hyperparameters set to 16 SG tokens and $\alpha = 1.0$. We search the hyperparameter space with $S' \in \{8, 12, 16, 24\}$, and $\alpha \in \{0.1, 0.5, 1.0\}$. For ViCLIP, a training batch consists of video-text pairs and VidSG data in an 8:2 ratio, while for InternVideo2, the ratio is 6:4. Regarding the LoRA configuration, the rank is set to 8, and the dropout rate is 0.05. Both models are optimized using AdamW [70] with a learning rate of 2e-7, a cosine learning rate scheduler, and a weight decay of 0.2. Training is conducted for 10 epochs on a NVIDIA GeForce A6000 48GB GPU.

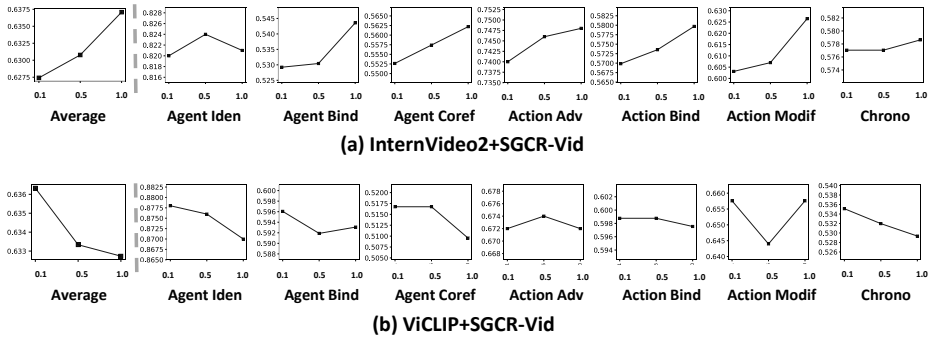

Figure 6: Performance over different α values.

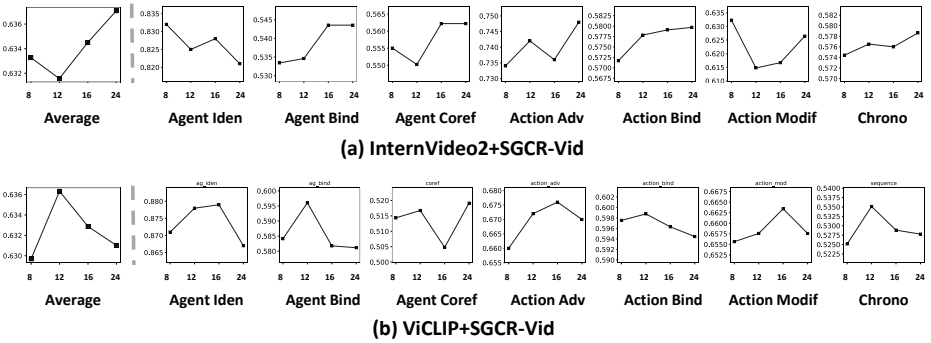

Figure 7: Performance over different S' values.

B.4 Hyperparameter Sensitivity

SGCR-Vid has two key hyperparameters: the loss coefficient for the VSG branch (α) and the number of SG tokens assigned per frame (S'). To investigate their impact, we evaluate performance changes across different hyperparameter values on the VELOCITI dataset using InternVideo2+ SGCR-Vid and ViCLIP+SGCR-Vid.

Loss coefficient for the VSG branch (α). Figure 6 shows how varying α affects performance. We observe that, on average, performance of InternVideo2+SGCR-Vid improves as α increases, while ViCLIP+SGCR-Vid achieves better results with smaller α values. We attribute this difference to the fact that InternVideo2's video encoder has a deeper layer, with 40 layers compared to ViCLIP's 12 layers. Therefore, InternVideo2 requires a larger α to achieve effective training. On the other hand, increasing α in ViCLIP may cause overfitting due to its shallower architecture, which leads to a drop in performance.

Number of SG tokens (S'). Figure 7 shows the effect of varying the number of SG tokens per frame. We observe that InternVideo2+SGCR-Vid achieves its highest average performance with 24 SG tokens, while ViCLIP+SGCR-Vid performs best using 12 SG tokens rather than a larger number.

This suggests that simply increasing or decreasing the number of SG tokens does not guarantee better performance. Rather, the optimal number of SG tokens depends on the model architecture.

Overall, we observe that the best performance of both models varies across different subtypes as the hyperparameters are adjusted. Considering that each subtype requires different reasoning, this indicates that the impact of hyperparameter changes on reasoning differs accordingly.

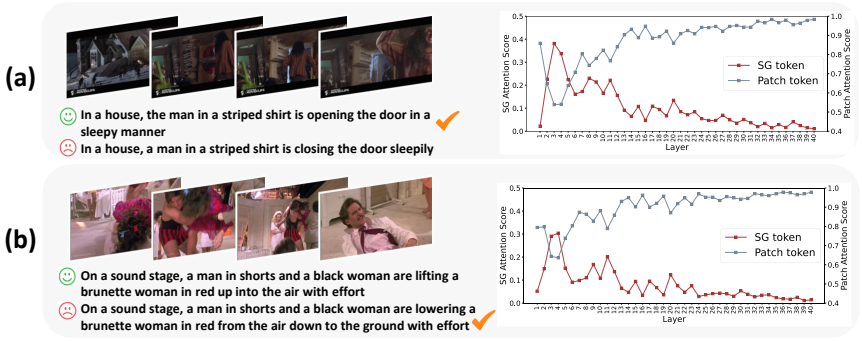

Figure 8: (a) Correctly predicted case, (b) Incorrectly predicted case of InternVideo2$_{26M}$+SGCR-Vid.

Figure 9: (a) Correctly predicted case, (b) Incorrectly predicted case of InternVideo2$_{26M}$+SGCR-Vid.

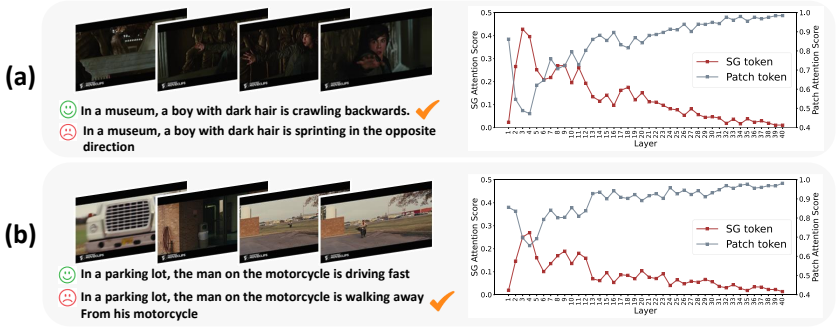

Figure 10: (a) Correctly predicted case, (b) Incorrectly predicted case of InternVideo2$_{26M}$+SGCR-Vid.

B.5 Additional Qualitative Results for Contribution of SG Tokens

In Section 5.4 of the main paper, we analyzed the relative contribution of the SG token compared to the patch token in terms of the CLS token in the video encoder. We observed that, in both correctly predicted cases (a) and incorrectly predicted cases (b), the SG token had a higher contribution in the lower layers. Furthermore, in the correctly predicted cases, we observed that the SG token contributed more in the early layers compared to the incorrectly predicted cases. In this section, we provide additional results to further consolidate these observations, shown in Figure 8- 12.

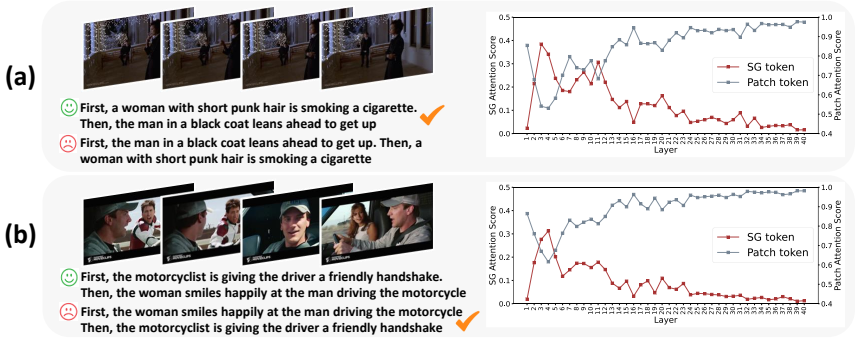

Figure 11: (a) Correctly predicted case, (b) Incorrectly predicted case of InternVideo2$_{26M}$+SGCR-Vid.

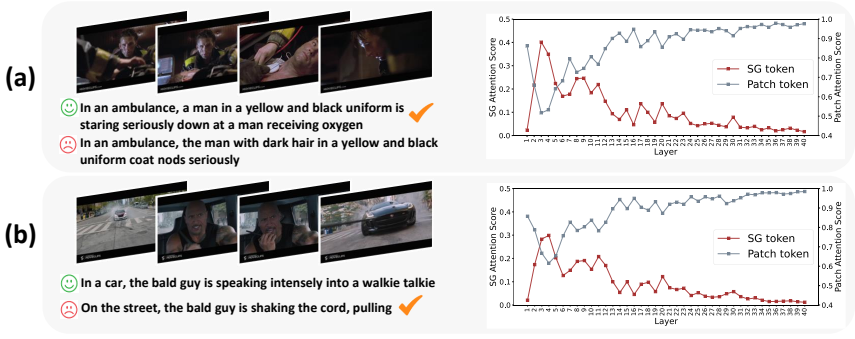

Figure 12: (a) Correctly predicted case, (b) Incorrectly predicted case of InternVideo2$_{26M}$+SGCR-Vid.