Supplementary Material

In this supplement, we provide additional and clarifying details for the main paper. Section A contains implementation details including the model architecture, pre-training objectives, datasets, parameters of optimization, and computational cost of our model. Section B expands the experimental results of Table 2, 6, and 7 in the main paper and offers the analysis of the model behavior in different question types. We also conduct additional experiments testing the modeling decision on ActivityNet-QA, as well as evaluating the influence of temporal resolutions of the image-language model, the number of concatenated videos for Temporal Referring Modeling, and the loss combination strategy.

A Implementation Details

A.1 Model Architectures

We introduce the details of our Decoupled Spatial-Temporal Encoders (DeST). Following [19] and [7], the image encoder is a 12-layer Vision Transformer [6], and the video encoder contains a Video Swin Transformer [22] (Swin-B) pre-trained on Kinetics-600 [2] for feature extraction and a 6-layer Transformer for contextualization. The question and answer encoder are both 6-layer Transformers [30] with each layer composed of a self-attention operation and a feed-forward network (FFN). The image- and video-language encoder are two 6-layer Transformers where each layer contains an additional cross-attention operation [11, 12, 18, 19, 20], in which text features serve as queries and perform attention to visual features. The question, image, and image-language encoder are the same as the modules of ALBEF [19] pre-trained on VQA [8]. The video contextualization module and video-language encoder are initialized from the question and image-language encoder respectively. The image and video encoder are fixed during the whole training process. The detailed parameters are listed in Table I.

Hyperparameter	Value
Embedding Size (D)	768
Number of Patches (N)	576
Video Feature Size (H)	1024
FFN Inner Hidden Size	3072
Number of Attention Heads	12
Attention Dropout	0.1
Dropout	0.1

Table I: Hyperparameters for the architecture.

Since the optimization of video encoding is not included in video-language training, we extract and store video features to save memory. We operate the Video Swin Transformer with the same configuration as Swin-B, which samples every two frames and transforms a window of 32 frames into one feature. For long videos, such as ActivityNet [35] with an average length of 180 seconds, we shift the window by 32 frames. For others, such as the datasets used in pre-training or AGQA 2.0 [10], we shift the window by 16 frames, and thus

every window overlaps with half of its previous and next window. Features of extremely long videos are sampled such that all videos are within a limited length.

A.2 Video-Language Pre-training

A.2.1 Details of Question and Video Synthesis for Temporal Referring Modeling

Temporal Referring Modeling (TRM) generates questions to inquire about absolute and relative temporal positions of specific events in videos. Questions are formed by choosing from five templates and filling in the templates with video descriptions. The choice of templates includes "What happens?", "What happens at the beginning?", "What happens at the end?", "What happens before `[event x]`?", and "What happens after `[event x]`?", where the first question is irrelevant to temporal relations but incorporated to facilitate video-language matching. The other four questions are designed for resemblance to video QA requiring temporal modeling, such as *Temporal Relationships* in ActivityNet-QA [35] or *State Transition* in TGIF-QA [13].

Except for the first question paired with a single video, the corresponding videos of other questions are synthesized by concatenating videos sampled from video captioning datasets. This operation simulates a sequence of events that happen one after another and provides us with the exact position of each event.

One may be concerned that the transitions of events in real videos are rather smooth and ambiguous, instead of clear differences between videos in a random concatenated video sequence, where people, objects, and almost the entire scenes drastically change. For example, in a video where people clean up the table after finishing dinner in the dining room, most of the visual elements, such as the people and furniture, remain the same, but we humans can easily recognize these two events by comparing the actions and interactions between the people in the video. While TRM cannot generate such videos, our model has learned a similar capability with TRM to compare human actions and interactions between moments. During fine-tuning, it can focus on adapting to smooth transitions and thus learn faster than models with neither the capability of temporal reasoning nor event recognition.

A.2.2 Auxiliary Objective with Contrastive Learning

In addition to TRM, we apply an auxiliary objective during pre-training, which aligns video features with corresponding captions by contrastive learning, widely used in image- and video-language pre-training [14, 19, 23, 29, 31, 36]. Specifically, with the concatenated video feature sequence $\mathbf{e} = \{e^1_1, ..., e^1_{M_1}, e^2_1, ..., e^K_{M_K}\}$, we add the beginning and the end token before and after the sequence, as well as the temporal position encoding to each feature. Then after contextualization, we have $\mathbf{v} = \{v_{\texttt{bos}}, v^1_1, ..., v^1_{M_1}, v^2_1, ..., v^K_{M_K}, v_{\texttt{eos}}\}$. To align each video to its caption, the objective learns a similarity function $\text{sim}(v,c) = g_v(f_v(v))^\top g_c(f_c(c))$, such that parallel video-caption pairs have higher similarity scores. f_v produces the representation of $\mathcal{V}_k$, which averages the features of a video, *e.g.* $f_v(\mathcal{V}_k) = \sum_{m=1}^{M_k} v^k_m$, and f_c delivers the representation of a caption, which is the `[CLS]` embeddings of the caption feature encoded by the question encoder. g_v and g_c are two linear transformations that map the two representations into a normalized lower-dimensional space.

Following [19], we calculate the softmax-normalized video-to-caption and caption-to-

video similarity as:

$$p_k^{\text{v2c}}(\mathcal{V}_k) = \frac{\exp(\text{sim}(\mathcal{V}_k, \mathcal{C}_k)/\tau)}{\sum_{i=1}^{K} \exp(\text{sim}(\mathcal{V}_k, \mathcal{C}_i)/\tau)}, \quad p_k^{\text{c2v}}(\mathcal{C}_k) = \frac{\exp(\text{sim}(\mathcal{C}_k, \mathcal{V}_k)/\tau)}{\sum_{i=1}^{K} \exp(\text{sim}(\mathcal{C}_k, \mathcal{V}_i)/\tau)}, \tag{1}$$

where τ is a learnable temperature parameter. To increase the difficulty, we collect video-caption pairs from all video sequences in the same mini-batch B, and thus K is K times the size of a mini-batch in practice. Then, similar to [19, 25], let $\boldsymbol{y}^{\text{v2c}}(v)$ and $\boldsymbol{y}^{\text{c2v}}(c)$ denote the ground-truth one-hot similarity, where the probability of positive and negative pair are 1 and 0. The video-caption contrastive loss is defined as the cross-entropy CE between $\boldsymbol{p}$ and $\boldsymbol{y}$:

$$\mathcal{L}_{\text{align}} = \frac{1}{2}\mathbb{E}_{(\mathcal{V},\mathcal{C})\sim B}[\text{CE}(\boldsymbol{y}^{\text{v2c}}(\mathcal{V}), \boldsymbol{p}^{\text{v2c}}(\mathcal{V})) + \text{CE}(\boldsymbol{y}^{\text{c2v}}(\mathcal{C}), \boldsymbol{p}^{\text{c2v}}(\mathcal{C}))] \tag{2}$$

A.2.3 Pre-training Datasets

TRM samples video-caption pairs from video captioning datasets. We want the datasets as diverse as possible, not limited to cooking [5], movies [26], or indoor actions [28]. To maintain the computation within an affordable size, videos cannot be too long [16], or a video sequence would consist of few videos, which prohibits the model from learning long-term temporal dependency.

We pre-train the video-language encoder over VATEX [32] and TGIF [21]. VATEX contains 41K videos from Kinetics-600 [2] and 826K sentences, where each video is paired with multiple descriptions. The lengths of the videos are all 10 seconds, cropped for precise action recognition in Kinetics. TGIF is an open-domain dataset containing 100K animated GIFs from Tumblr and 120K sentence descriptions. The average duration is around 3.1 seconds. It is worth noting that using less pre-training data is not the main motivation of this work, but with effective objectives, our method has surpassed large-scale pre-training. If computational cost is affordable, training with more data is expected to advance the performance. We leave pre-training with longer videos and larger datasets for future work.

A.3 Optimization

The pre-training and fine-tuning are all optimized with AdamW optimizer and linear decay scheduling after warmup. All experiments are run with two NVIDIA RTX 3090s, with which the pre-training takes about 18 hours. The detailed hyperparameters are provided in Table II.

A.4 Computational Cost

The overall computation is the sum of the IL and VL models and depends on the number of input frames T, video lengths, and the feature extractors. Let R and S denote the computation of ALBEF and Just-Ask, our method costs about $TR+S$ as we stack more Transformer layers than Just-Ask, but the two streams share the question encoder. Specifically, the frozen image encoders cost about 12 GFLOPs per frame, and the video encoder performs 40 GFLOPs per window. The other modules need 28 GFLOPs.

Hyperparameter	Pre-train	ActQA	AGQA
Learning Rate (Base)	1e-5	2e-5	2e-5
Learning Rate (Video)	5e-5	2e-4	5e-5
Learning Rate (MLP)	2.5e-4	1e-3	2e-4
Learning Rate (Ans)	2e-5	2e-5	2e-5
Weight Decay	1e-2	1e-2	1e-2
AdamW ε	1e-8	1e-8	1e-8
AdamW β_1	0.9	0.9	0.9
AdamW β_2	0.98	0.98	0.98
Training Steps	60K	-	-
Training Epochs	-	5	4
Warmup	0.03	0.1	0.1
Batch Size	128	64	64
Max Video Length	100	100	100
Max Question Length	50	-	-
Number of Videos (K)	8	-	-
Number of Frames (T)	-	16	8

Table II: Hyperparameters for pre-training (Pre-train), ActivityNet-QA (ActQA), and AGQA 2.0 (AGQA). Base: the question, image, and image-language encoder. Video: the video and video-language encoder. Ans: the answer encoder.

B Experimental Details

B.1 Details of Temporal Modeling Analysis

Some may question our preliminary analysis of temporal modeling, in which we first train a model with normal inputs and test it with normal and shuffled inputs. The performance drops imply the sensitivity to the order of frames, and thus little difference may indicate the incompetence of temporal modeling. Training and testing a model with shuffled input can also completely eliminate the temporal information, but this approach only reveals how well a model solves a task with spatial information (or dataset bias if the task is designed for evaluating temporal modeling), and thus it is not suitable for assessing a model's capability of temporal modeling.

We conduct the analysis on AGQA and VIOLIN as some other video QA benchmarks are less appropriate. For example, some questions in ActivietNet-QA need only spatial knowledge. In NeXT-QA [33], while 29% of questions are about temporal relations, others aim at spatial information or more advanced cognition, *e.g.* causal reasoning. The split of *State Transition* in TGIF-QA [13], though expected to suit this analysis well, could be solved by VIOLET without understanding the order of frames in our experiment (Table III).

B.2 Pre-training Data Used by Prior Approaches

Compared with state-of-the-art approaches, DeST performs better on ActivityNet-QA with orders of magnitude less pre-training data. We include some widely-used pre-training datasets that are abbreviated in Table 3 of the main paper: 100M: HowTo100M [24]; 69M: HowToVQA69M [34]; 180M: YT-Temporal-180M [36]; 2.5M: WebVid [1]; 14M/3M: Conceptual Caption [3, 27]; 5.6M: COCO [4] + VisualGenome [17].

Method	Benchmark	Accuracy
VIOLET	TGIF-QA	95.34
	TGIF-QA*	95.36±.08

Table III: Results of VIOLET taking shuffled frames as input on the questions of *State Transition* of TGIF-QA. (* signifies that input frames are shuffled. We report the average of three results for the shuffle experiment.)

B.3 Full Results and Analysis on AGQA 2.0

AGQA 2.0 provides extensive annotations. Each question is associated with the reasoning abilities necessary to answer the question. The annotations cover four aspects: reasoning types, semantics class, structures, and answer types. Reasoning types define the design of question templates for evaluating certain reasoning abilities. We list some examples of question templates created by [9] in Table IV for the following analysis of our model's behavior. The semantics class of a question describes its main subject: an object, relationship, or action. Question structures include open questions (query), comparing attributes of two options (compare), choosing between two options (choose), yes/no questions (verify), and understanding of logical operators, such as *and* or *or*. Questions with binary answer types have restricted answer choices, such as Yes/No, Before/After, or two specified options, while the answers to open-ended questions are much more diverse.

Reasoning Type	Example of Template
Object-Relationship	What/Who/When/Where/How did they <rel> <object>?
Relationship-Action	Did they <relation> something before or after <action>?
Object-Action	Did they interact with <object> before or after <action>?
Superlative	What were they <action> first/last?
Sequencing	What did the person do after <action>?
Exists	Did/Does/Do <concept> occur?
Duration Comparison	Did they <action1> or <action2> for longer?
Activity Recognition	What does the person do before/after/while <action>?

Table IV: Reasoning types and examples of their templates of AGQA 2.0.

B.3.1 Full Results of Temporal Modeling Analysis

The full results of Table 2 in the main paper are presented in Table V, where we gauge the efficacy of temporal modeling of prior approaches by inputting shuffled videos and measuring performance drop. While Just-Ask [34] demonstrates improvement in *Relationship-Action*, *Object-Action* and *Sequencing*, VIOLET [7] performs similar in most types. The poor performance of VIOLET may be attributed to sparsely sampling, by which they enabled end-to-end training, but it turns out that taking few frames seems not able to summarize the temporal dynamics of whole videos.

B.3.2 Full Results and Analysis of Our Method

We show the full results of our method on AGQA 2.0 with ablation of components and pre-training strategies in Table VI.

	Type	Just-Ask*	Just-Ask	VIOLET*	VIOLET
Reasoning	Object-Relationship	46.30	47.83	49.01	48.91
	Relationship-Action	50.78	66.55	50.04	50.02
	Object-Action	50.77	68.78	50.13	50.24
	Superlative	37.96	39.83	39.47	39.49
	Sequencing	50.66	67.01	49.86	49.91
	Exists	57.15	59.35	54.58	54.70
	Duration Comparison	50.66	50.49	30.70	30.64
	Activity Recognition	19.87	21.53	3.13	3.13
Semantic	Object	46.34	49.31	49.18	49.08
	Relationship	54.63	59.60	52.32	52.41
	Action	49.78	58.03	41.47	41.45
Structure	Query	45.53	47.25	48.15	47.98
	Compare	50.84	65.11	47.65	47.69
	Choose	39.78	41.00	46.97	46.90
	Logic	54.87	56.20	50.99	51.24
	Verify	56.22	58.13	55.42	55.46
Overall	Binary	49.95	55.35	50.30	50.33
	Open	45.53	47.25	48.15	47.98
	All	47.72	51.27	49.22	49.15

Table V: Full results of the preliminary analysis of temporal modeling on AGQA 2.0. (* means shuffled input. We report the result of one experiment.)

Type	T	T+F	T+**F**	T+V	T+**V**	T+**F**+V	T+**F**+**V***	T+**F**+**V**
Object-Relationship	39.15	49.21	50.33	51.67	53.40	56.39	57.16	59.66
Relationship-Action	50.05	50.61	50.00	49.83	71.57	53.25	51.64	72.98
Object-Action	49.99	50.11	50.00	50.03	74.74	56.27	54.42	75.20
Superlative	34.00	37.96	38.87	41.82	43.80	44.54	45.70	48.94
Sequencing	49.89	50.26	49.86	49.86	72.60	54.92	53.14	73.53
Exists	50.09	57.77	59.06	50.86	53.68	59.95	59.04	63.21
Duration Comparison	48.71	51.43	55.04	44.96	37.34	62.58	60.26	60.39
Activity Recognition	14.63	14.81	16.84	13.16	19.60	21.25	21.44	27.78
Object	39.25	49.24	50.16	51.44	55.28	56.50	57.31	61.27
Relationship	50.08	54.73	55.76	50.58	57.14	57.33	56.07	63.93
Action	48.49	49.98	50.86	47.09	56.52	56.35	54.39	65.96
Query	33.28	48.18	49.33	51.99	56.48	57.46	58.90	61.22
Compare	49.99	50.62	50.73	49.42	68.28	56.11	54.23	72.04
Choose	48.10	46.24	46.76	49.50	42.38	50.34	50.40	53.01
Logic	50.03	54.28	56.36	50.68	51.91	57.52	55.78	59.18
Verify	49.98	57.48	58.45	51.28	53.44	59.49	59.45	63.02
Binary	49.47	52.00	52.70	50.17	54.74	55.76	55.01	62.61
Open	33.28	48.18	49.33	51.99	56.48	57.46	58.90	61.22
All	41.32	50.07	51.00	51.08	55.62	56.61	56.97	61.91

Table VI: Full results of our method on AGQA 2.0 with ablation of components and pre-training strategies. (T: questions; F: frames; **F**: frames with the image-language encoder pre-trained on VQA; V: videos; **V**: videos with the video-language encoder pre-trained with TRM; *: shuffled video inputs.)

We first examine the performance of inputting only questions (T), which reveals the bias of the datasets as these questions can be solved without grounding to videos. With a rigorous balancing procedure, this model cannot achieve more than 50% accuracy on any question type, but some questions, for example, those belonging to *Relationship-Action*, *Object-Action*, and *Exists* appear easier than others.

Inputting frames (T+F) improves the overall performance by about 10% accuracy, which mostly comes from *Object-Relationship* and *Exists*. This is reasonable as these questions involve less temporal information according to the templates, and they are more likely to be solved with a few static frames with spatial information about humans, objects, and scenes. Pre-training the image-language encoder with VQA [8] (T+**F**) shows further improvement in *Exists*, which seems more similar to the question design of image QA.

Accessing videos (T+V) is helpful for different question types such as *Superlative*, in which the questions ask about something happening first or last, but some other questions that also require temporal modeling, including *Relationship-Action* or *Sequencing*, are not improved. Besides, video inputs do not enhance the performance of questions improved by frame inputs. This complementary advantage of frames and videos is consistent with our findings in the preliminary analysis, and inputting both frames and videos (T+**F**+V) does surpass inputting only one of them in all reasoning types.

Pre-training the video-language encoder with TRM (T+**F**+**V**) boosts the performance of most reasoning types, especially *Relationship-Action*, *Object-Action*, and *Sequencing*. These questions all need temporal modeling of event sequences in videos and have question formats more similar to TRM. The huge performance gap (20% accuracy) between normal (T+**F**+**V**) and shuffled video inputs (T+**F**+**V***), as well as the little gap between no pre-training (T+**F**+V) and shuffled inputs (T+**F**+**V***), suggests successful temporal modeling and verifies the efficacy of TRM.

Despite the enhancement in most questions, TRM still struggles with some reasoning types, for example, *Duration Comparison*, asking a machine which action lasts longer. These questions require a machine to memorize multiple events and identify their starting and ending point to obtain their duration. Such abilities are beyond the intention of developing TRM, and we leave it for future exploration.

B.3.3 Full Results of Ablation Study of Encoding Streams

Table 7 in the main paper is expanded as Table VII, where we first train a model with both image- and video-language encoders, and evaluate each stream with the test set.

B.4 Ablation Study on ActivityNet-QA

The ablation study is also conducted on ActivityNet-QA, as reported in Table VIII. The result proves that the image-language model is capable of answering some video QA problems, and the strategy of bridging image QA and video QA further increases the performance.

B.5 Temporal Resolutions of the Image-Language Encoder

We estimate the influence of varying temporal resolutions (the number of frames T) of the image-language encoder. As displayed in Figure I, taking more frames substantially increases the performance on ActivityNet-QA, while the improvement on AGQA is insignificant. This discrepancy could be explained by the distribution of question types in the two

Type	VL	IL
Object-Relationship	49.04	20.91
Relationship-Action	50.00	0.60
Object-Action	50.00	0.99
Superlative	36.00	15.86
Sequencing	49.85	0.80
Exists	57.09	0.07
Duration Comparison	58.99	0.00
Activity Recognition	13.06	0.92
Object	48.92	20.60
Relationship	54.58	0.20
Action	52.19	0.38
Query	47.35	26.68
Compare	51.24	0.69
Choose	48.29	22.11
Logic	55.09	0.04
Verify	56.51	0.08
Binary	52.52	6.30
Open	47.35	26.68
All	49.91	16.56

Table VII: Full results of the ablation study on two encoding streams. (VL: ablating the video-language encoder; IL: ablating the image-language encoder.)

Question	Frames	Video	Acc
✓			31.01
✓	✓		43.15
✓	VQA		46.66
✓	VQA	TRM	46.79

Table VIII: Ablation study on ActivityNet-QA. (✓ means the modality is presented. VQA: pre-trained on VQA. TRM: pre-trained with TRM.)

benchmarks, where ActivityNet-QA contains more questions of static information and the prediction is likely stabler and more robust when more frames are provided. More questions in AGQA are related to temporal dynamics and thus less affected by the number of frames.

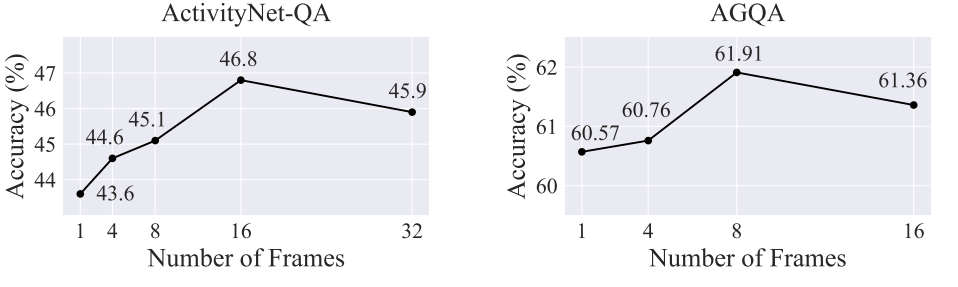

Figure I: The results on two benchmarks of inputting different numbers of frames to the image-language encoder.

B.6 The Number of Videos for Temporal Referring Modeling

We alter the number of videos concatenated for TRM (the variable K) and study its influence. The accuracy on AGQA 2.0 with respect to the number of videos is presented in Figure II. We can observe that increasing the number of videos is not always beneficial to the downstream

task. Concatenating too many videos may result in extremely long temporal dependency, which is hard for a model to encode.

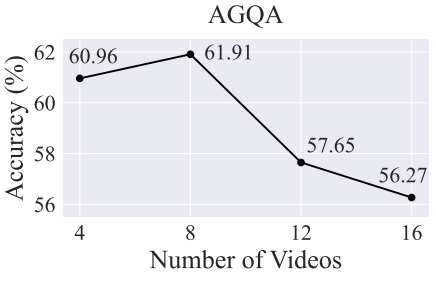

Figure II: The performance on AGQA 2.0 of varying the numbers of concatenated videos for Temporal Referring Modeling.

B.7 Multi-Task Loss Weighing

As described in A.2.2, we align visual and linguistic content by contrastive learning. An experiment is conducted to evaluate different approaches to loss combinations. Let $\mathcal{L}_{\text{TRM}}$ and $\mathcal{L}_{\text{align}}$ denote the loss of TRM and video-language contrastive loss. We compare adding losses directly $\mathcal{L}_1$ and weighing losses by uncertainty [15] $\mathcal{L}_2$:

$$\begin{aligned} \mathcal{L}_1 &= \mathcal{L}_{\text{TRM}} + \mathcal{L}_{\text{align}}, \\ \mathcal{L}_2 &= \frac{1}{2\sigma_1^2}\mathcal{L}_{\text{TRM}} + \frac{1}{2\sigma_2^2}\mathcal{L}_{\text{align}} + \log\sigma_1^2 + \log\sigma_2^2, \end{aligned} \tag{3}$$

where σ_1 and σ_2 are two learnable parameters. Following [15], in practice the model learns to predict $s := \log\sigma^2$ for numerical stability.

Loss combination	Acc
Unweighted	61.91
Weighing by uncertainty	60.15

Table IX: Comparison between different approaches to loss combination.

Table IX compares the accuracy on AGQA 2.0 of pre-training with the sum of losses unweighted and weighted by uncertainty. The result shows that the difference between the two approaches to combining losses is insignificant.

References

[1] 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 (ICCV)*, 2021.

[2] Joao Carreira, Eric Noland, Andras Banki-Horvath, Chloe Hillier, and Andrew Zisserman. A short note about kinetics-600. *arXiv preprint arXiv:1808.01340*, 2018.

[3] Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. Conceptual 12m: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[4] Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollar, and C. Lawrence Zitnick. Microsoft COCO captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*, 2015.

[5] Pradipto Das, Chenliang Xu, Richard F. Doell, and Jason J. Corso. A thousand frames in just a few words: Lingual description of videos through latent topics and sparse object stitching. In *2013 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[6] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2021.

[7] 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 arXiv:2111.12681*, 2022.

[8] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the V in VQA matter: Elevating the role of image understanding in Visual Question Answering. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[9] Madeleine Grunde-McLaughlin, Ranjay Krishna, and Maneesh Agrawala. AGQA: A benchmark for compositional spatio-temporal reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[10] Madeleine Grunde-McLaughlin, Ranjay Krishna, and Maneesh Agrawala. AGQA 2.0: An updated benchmark for compositional spatio-temporal reasoning. *arXiv preprint arXiv:2204.06105*, 2022.

[11] Andrew Jaegle, Felix Gimeno, Andy Brock, Oriol Vinyals, Andrew Zisserman, and Joao Carreira. Perceiver: General perception with iterative attention. In *International conference on machine learning (ICML)*, 2021.

[12] Andrew Jaegle, Sebastian Borgeaud, Jean-Baptiste Alayrac, Carl Doersch, Catalin Ionescu, David Ding, Skanda Koppula, Daniel Zoran, Andrew Brock, Evan Shelhamer,

Olivier J Henaff, Matthew Botvinick, Andrew Zisserman, Oriol Vinyals, and Joao Carreira. Perceiver IO: A general architecture for structured inputs & outputs. In *International Conference on Learning Representations (ICLR)*, 2022.

[13] Yunseok Jang, Yale Song, Youngjae Yu, Youngjin Kim, and Gunhee Kim. TGIF-QA: Toward spatio-temporal reasoning in visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[14] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, 2021.

[15] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[16] Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. Dense-captioning events in videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2017.

[17] 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 (IJCV)*, 2017.

[18] Jie Lei, Tamara L. Berg, and Mohit Bansal. Revealing single frame bias for video-and-language learning. *arXiv preprint arXiv:2206.03428*, 2022.

[19] Junnan Li, Ramprasaath R. Selvaraju, Akhilesh Deepak Gotmare, Shafiq Joty, Caiming Xiong, and Steven Hoi. Align before fuse: Vision and language representation learning with momentum distillation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[20] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. BLIP: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *Proceedings of the 39th International Conference on Machine Learning (ICML)*, 2022.

[21] Yuncheng Li, Yale Song, Liangliang Cao, Joel Tetreault, Larry Goldberg, Alejandro Jaimes, and Jiebo Luo. TGIF: A New Dataset and Benchmark on Animated GIF Description. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[22] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[23] Huaishao Luo, Lei Ji, Botian Shi, Haoyang Huang, Nan Duan, Tianrui Li, Jason Li, Taroon Bharti, and Ming Zhou. UniVL: A unified video and language pre-training model for multimodal understanding and generation. *arXiv preprint arXiv:2002.06353*, 2020.

[24] 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 (ICCV)*, 2019.

[25] 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. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, 2021.

[26] Anna Rohrbach, Atousa Torabi, Marcus Rohrbach, Niket Tandon, Christopher Pal, Hugo Larochelle, Aaron Courville, and Bernt Schiele. Movie description. *International Journal of Computer Vision (IJCV)*, 2017.

[27] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2018.

[28] Gunnar A. Sigurdsson, Gül Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *European Conference on Computer Vision (ECCV)*, 2016.

[29] Chen Sun, Fabien Baradel, Kevin Murphy, and Cordelia Schmid. Learning video representations using contrastive bidirectional transformer. *arXiv preprint arXiv:1906.05743*, 2019.

[30] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[31] Alex Jinpeng Wang, Yixiao Ge, Rui Yan, Yuying Ge, Xudong Lin, Guanyu Cai, Jianping Wu, Ying Shan, Xiaohu Qie, and Mike Zheng Shou. All in one: Exploring unified video-language pre-training. *arXiv preprint arXiv:2203.07303*, 2022.

[32] 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 (CVPR)*, 2019.

[33] Junbin Xiao, Xindi Shang, Angela Yao, and Tat-Seng Chua. NExT-QA: Next phase of question-answering to explaining temporal actions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[34] Antoine Yang, Antoine Miech, Josef Sivic, Ivan Laptev, and Cordelia Schmid. Just ask: Learning to answer questions from millions of narrated videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[35] Zhou Yu, Dejing Xu, Jun Yu, Ting Yu, Zhou Zhao, Yueting Zhuang, and Dacheng Tao. ActivityNet-QA: A dataset for understanding complex web videos via question answering. *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2019.

[36] Rowan Zellers, Ximing Lu, Jack Hessel, Youngjae Yu, Jae Sung Park, Jize Cao, Ali Farhadi, and Yejin Choi. MERLOT: Multimodal neural script knowledge models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.