MOMA-LRG: Language-Refined Graphs for Multi-Object Multi-Actor Activity Parsing

Zelun Luo, Zane Durante, Linden Li, Wanze Xie, Ruochen Liu, Emily Jin,
Zhuoyi Huang, Lun Yu Li, Jiajun Wu, Juan Carlos Niebles, Ehsan Adeli, Li Fei-Fei
Stanford University
{alanzluo, durante, lindenli, wanzexie, ruochenl, emilyjin, zhuoyih, tinally}@stanford.edu
{jiajunwu, jniebles, eadeli, feifeili}@cs.stanford.edu

Abstract

Video-language models (VLMs), large models pre-trained on numerous but noisy video-text pairs from the internet, have revolutionized activity recognition through their remarkable generalization and open-vocabulary capabilities. While complex human activities are often hierarchical and compositional, most existing tasks for evaluating VLMs focus only on high-level video understanding, making it difficult to accurately assess and interpret the ability of VLMs to understand complex and fine-grained human activities. Inspired by the recently proposed MOMA framework, we define **activity graphs** as a single universal representation of human activities that encompasses video understanding at the activity, sub-activity, and atomic action level. We redefine **activity parsing** as the overarching task of activity graph generation, requiring understanding human activities across all three levels. To facilitate the evaluation of models on activity parsing, we introduce **MOMA-LRG** (Multi-Object Multi-Actor Language-Refined Graphs), a large dataset of complex human activities with activity graph annotations that can be readily transformed into natural language sentences. Lastly, we present a model-agnostic and lightweight approach to adapting and evaluating VLMs by incorporating structured knowledge from activity graphs into VLMs, addressing the individual limitations of language and graphical models. We demonstrate a strong performance on activity parsing and few-shot video classification, and our framework is intended to foster future research in the joint modeling of videos, graphs, and language.

1 Introduction

Computer vision is currently undergoing a paradigm shift from models trained on crowd-labeled data [1, 2] to large-scale base models [3, 4, 5, 6, 7] trained on noisy but readily accessible image-text pairs. Video understanding is no exception, with the rise of Video-Language Models (VLMs) [8, 9, 10, 11, 12, 13, 14] that have shown remarkable generalization capabilities on videos from new domains. When compared to fixed-set video classification [2, 15, 16], VLMs are able to learn and represent a wider variety of concepts and demonstrate superior low-shot abilities on many downstream tasks due to the flexibility and open-vocabulary nature of language. Besides, while annotating videos remains one of the most laborious processes in computer vision, VLMs utilize widely and freely available video-text pairs [9, 8, 13, 17] facilitating larger and more diverse pre-training at a lower cost.

Submitted to the 36th Conference on Neural Information Processing Systems (NeurIPS 2022) Track on Datasets and Benchmarks. Do not distribute.

Despite improved generalization and scalability, the application of VLMs to human activity recognition faces a number of challenges. The first challenge is adapting existing VLMs for *fine-grained, actor-centric activity recognition*. Essential computer vision applications in healthcare, surveillance, and robotics are often characterized by complex human activities involving many concurrent events between actors and objects. On the other hand, existing VLMs are typically trained on noisy, coarse-grained internet data and evaluated on downstream tasks that are focused on high-level video understanding [18, 10, 12]. It is unclear how VLMs can be effectively adapted to and accurately evaluated for recognizing fine-grained activities. The second challenge is the *lack of a single overarching task* for evaluating VLMs on activity recognition. Human activities are often hierarchical and compositional [19, 20, 21], requiring explicit modeling and evaluation at multiple levels of granularity. Existing downstream tasks used for VLM evaluation, such as activity classification [18], activity segmentation [10, 12], video-text retrieval [13, 12, 8] and VideoQA [10, 13], only provide an incomplete assessment of VLM performance on activity recognition. Lastly, the black-box nature of VLMs makes their predictions difficult to interpret. This hinders the application of VLMs to many risk-averse domains [22, 23, 24], where it is necessary to interpret VLM predictions in a structured and symbolic manner.

To address the aforementioned challenges, this paper first aims to standardize an overarching representation of human activities across varying levels of granularity and provide a unified task for VLMs by generating this representation. Inspired by the recently proposed MOMA [20] framework, we introduce **activity graphs** as dynamic graphs that encompass activities, sub-activities, and atomic actions in a video. Specifically, an activity graph is a representation that simultaneously (1) provides a class label on the activity level, (2) provides a dynamic sub-activity label that contains all temporal boundaries and categories of sub-activities, and (3) captures fine-grained atomic actions in multi-object multi-actor settings with spatial localization and tracking of all entities and temporal localization of all predicates. Further, we introduce **activity parsing** as the overarching task of predicting the activity graph from a video, thereby achieving multi-level activity recognition via activity graph generation.

Next, we introduce the **MOMA-LRG** dataset, a novel activity recognition dataset that leverages both the descriptive capacity of activity graphs and the expressivity of natural language. MOMA-LRG involves videos with Multiple Objects and Multiple Actors (MOMA) and is designed to enable models to understand a broad set of human activities. To enable few-shot activity recognition with language, MOMA-LRG provides Language-Refined Graph annotations in a format that enables an easy conversion from the structured graph representation into natural language sentences.

Lastly, we introduce **GraphVLM** as our framework for evaluating VLMs on activity parsing, consisting of an activity parsing model and a transfer learning paradigm. We first propose an architecture for activity parsing that can be readily adapted for VLMs, featuring a video stream, a text stream, and video tokenizers shared across all three levels of activity. Although fine-tuning is a widely used transfer learning technique, it requires a fixed architecture and clip sampling approach. In GraphVLM, we propose a transfer learning approach based on knowledge distillation, which enables the adaptation of VLMs in a flexible and lightweight manner.

2 Related work

Activity recognition. Activity recognition tasks a model to identify events performed by human agents. The dominant task has been activity classification on benchmarks such as [2, 15, 16], but other datasets add richer information to their annotations such as spatio-temporal scene graphs [19, 21]. 3D CNNs that jointly model space and time [25, 26, 27] have been popular for activity recognition historically, but recently transformers-based methods have achieved comparable or superior results [28, 29, 30, 31, 32]. Action localization datasets like ActivityNet [15], THUMOS'14 [33], and FineGym [34] label the temporal boundaries and the action class of each action that happens within the video. The methods utilized here tend to propose temporal boundaries and then classify them [35, 36, 37, 38, 39], while others attempt to jointly model both [40, 41]. Other datasets [42, 43] additionally add a spatial dimension which attempts to localize actions in both space and time, where methods include long-term feature banks [44], relation-modeling [45, 46] and pretrained video modules and object detection architectures [43]. MOMA-LRG encompasses and extends these existing video datasets by labeling actions, the sub-activities that compose them, and providing rich scene graph annotations to describe the interactions between entities in a crowded scene.

Video and language models. Several works pre-train large-scale models jointly on video and language data for a variety of downstream video-language tasks, such as video captioning, VQA, and video-text retrieval [47, 18, 12, 10, 14, 13]. Pre-training these models often either relies on a combination of masked-language-modeling (MLM) and masked-frame-modeling (MFM) [18, 10, 14] or contrastive learning [47, 9, 12, 13]. These VLMs have shown promising zero-shot results for activity recognition tasks such as activity classification [18], action segmentation [12], and action step localization [12, 10]. These methods show powerful zero-shot capabilities for high-level video understanding, however, they lack explicit knowledge about fine-grained interactions between actors and objects.

Fine-tuning large pre-trained models. Finetuning pre-trained large language models for downstream tasks has recently become the most popular learning method in NLP [48, 49]. Methods for efficiently fine-tuning these large language models, such as using adapter modules [50] and prompting [51, 49], use only a small number of learnable parameters while keeping most of the pre-trained model frozen. As a result, these methods are less computationally expensive than full fine-tuning of the entire model, which is the traditional fine-tuning method used in computer vision [52, 53]. There has been recent work adopting these efficient fine-tuning techniques for vision-language models. [54] and [55] both propose adapter module methods and demonstrate comparable performance to full fine-tuning. On the prompt-tuning side, Zhou et al. [56] develops prompt-tuning techniques for vision-language models for zero-shot image classification, Yao et al. [57] uses prompt-tuning for grounding referring expressions in images, and Ju et al. [58] uses image-level tokens and textual prompt tuning for few-shot action recognition.

Low-shot activity recognition. Low-shot activity recognition, which recognizes activities that were either scarce or missing from the training set, reduces the reliance on obtaining expensive labels for crowded scenes. Zero-shot action recognition tries to predict unseen classes, where approaches either project visual features into a semantic embedding space [59, 60, 61, 62, 63], an intermediate embedding space learned from textual and visual data [64, 65, 66], or a visual embedding space that is synthesized by incorporating semantic information [67, 68]. Few-shot learning tends to be based on metric-learning that learn similarities to the scarce in-domain training examples [69, 70, 71, 72, 73, 74]. Other recent methods utilized self-supervised and contrastive or meta-learning approaches [75, 76] with a high degree of success. More recently, visual-language models have shown strong results on zero-shot recognition [18, 4, 12].

Visual grounding and scene graphs. Visual grounding merges visual and language understanding by attempting to localize an object in an image space given a text query. Some datasets associate nouns with bounding boxes in the video [77, 78], while others introduce scene graph annotations that describe the relationships between entities in the image [79]. Yu et al. [80] and Chen et al. [81] demonstrate that using hierarchical text-generated scene graphs allows for better representation of fine-grained semantics than using raw text alone. Other work has proposed the use of an action graph to generate novel videos [82, 83], defining an object-centric graph with objects as nodes and edges as actions with temporal annotations. Wang et al. [84] models videos as space-time region graphs that capture long range dependencies and spatial-temporal relations between objects.

3 Activity Graphs and the MOMA-LRG Dataset

MOMA-LRG improves and extends MOMA [20] by providing the new abstraction of the activity graph as the single universal representation of human activities that encompasses video understanding at the activity, sub-activity and atomic action levels. Thus, our new formulation for the task of activity parsing as activity graph generation allows for a single overarching task for hierarchical

Table 1: A comparison of MOMA-LRG with related video datasets. A dash signifies that the annotation does not exist in the dataset, and n/a indicates that the paper did not report a specific number.

									Atomic action					
			Activity		Sub-activity		Actor		Object		Unary predicate		Binary predicate	
Dataset	Hours	Levels	Classes	Instances	Classes	Instances	Classes	Instances	Classes	Instances	Classes	Instances	Classes	Instances
AVA [42]	107.5	2	-	-	-	-	1	424K	-	-	14	424K	66	651K
AVA-Kinetics [85]	638.9	2	-	-	-	-	1	310K	-	-	13	633K	47	$\sim$ 800K
Action Genome [19]	82	2	157	10K	-	-	-	-	35	0.4M	-	-	25	1.7M
FineGym [34]	708	3	10	4.9K	530	32.7K	-	-	-	-	-	-	-	-
Home Action Genome [21]	25.4	3	75	1.75K	453	24.6K	1	24.6K	86	n/a	-	-	29	583K
MultiSports [43]	18.6	2	66	37.7K	-	-	1	902K	-	-	-	-	-	-
Something V2 [16]	121	1	-	-	-	-	-	-	a few thousand	30K	-	-	174	318K
DALY [86]	31	1	10	3.6K	-	-	n/a	n/a	n/a	n/a	-	-	-	-
MEVA [87]	9.3K	1	37	n/a	-	-	n/a	n/a	n/a	n/a	-	-	-	
TITAN [88]	3	1	-	-	-	-	3	395K	2	249K	16	935K	28	426K
MOMA [20]	66	3	17	373	67	2364	20	80K	120	80K	52	12K	23	119K
MOMA-LRG	148	3	20	1.4K	91	15.8K	26	740K	126	396K	13	704K	52	1.4M

video understanding. The MOMA-LRG dataset also enables the training of few-shot video-language models by encapsulating high-level and fine-grained semantics within activity videos.

3.1 Activity Graphs

The key abstraction of MOMA-LRG is the activity graph, an all-encompassing and human-interpretable representation of human activities that captures temporal changes and compositionality. An activity graph is a dynamic graph $\mathbf{G} = [G_1, G_2, \ldots, G_t]$ represented as an ordered list of timed events, such as the addition or deletion of nodes and edges over time. Each $G_i \in \mathbf{G}$ can be represented as the pair (V_i, E_i), where V_i is a set of entities and their attributes and the set of edges $E_i = \{(v_{1i}, r_i, v_{2i}), \ldots\}$ encapsulates the relationships between the source and target entities. An activity graph has two levels of labels: (1) an activity label which stays constant for the entire graph; (2) a dynamic subactivity label that changes when subactivities begin and end. Each unique activity instance is associated with a unique activity graph. Unlike the dynamic scene graphs from [19], activity graphs in MOMA-LRG are *activity-centric* and contain information relevant only to the activity. The activity graph includes three levels of hierarchy:

Activity. An activity is an event where several human (actors) and non-human (objects) entities interact to complete a multi-step task. Parsing the activity returns an activity class label associated with the activity graph.

Sub-activity. A sub-activity is a step that makes up part of a larger activity and are (1) temporally localized within an activity and (2) mutually exclusive between activities. For example, the sub-activity `the adult is comforting the child` is unique to the activity of `babysitting`. Sub-activities are represented with two labels: 1) a temporal boundary indicating the start and end time relative to the activity video; 2) a semantic label which represents the class of the sub-activity. Parsing the activity produces the dynamic sub-activity label that contains the temporal boundaries and class of all sub-activities.

Atomic action. An atomic action describes how entities interact within a sub-activity video, which involves understanding entities and their predicates. Atomic actions are **entity-centric**, i.e. entities involved in an atomic action are spatially and temporally localized. Entity labels are entity-centric to disambiguate which entities are involved in a given atomic action. Atomic actions are activity and sub-activity agnostic—that is, a given atomic action class can be involved in many different sub-activity and activity instances. The predicates are atomic, such that they are generic across all activities. Predicates like `running`, `walking` or `bending` are general and can be involved in multiple sub-activity and activity instances. At this level, activity parsing evaluates the ability of the model to predict: (1) all relationships present in the global context, similar to scene graph generation and relationship retrieval and (2) all the predicates specific entity is involved across time, similar to spatio-temporal atomic action detection in [42].

Atomic actions consist of two components: entities and predicates. An **entity** is defined to be either a human actor or an object that is present in the scene and relevant to the action being performed. In a

Figure 1: An example of the results for the activity parsing task. For the activity level, the model predicts the activity class `haircut` for the video input. For the subactivity level, the temporally localized sub-activity predictions are displayed on the bottom, with the corresponding sub-activity classes on the legend placed on the left. For the atomic action level, the model has localized and tracked all entities (actors and objects) and predicted their interactions as displayed in the graph visualization on the right. Note: this graphic is a live animation that can be viewed in an Adobe Acrobat PDF viewer.

video frame, we annotate each entity with a bounding box, class label, and instance ID. Throughout the video instance, an entity is therefore represented as a spatio-temporal tube with a corresponding semantic label. A **predicate** to describe an interaction that occurs with at least one entity. There are two different types of predicates: a unary predicate defined on a specific entity is called an **attribute**, whereas a predicate defined on two or more entities is called a **higher-order relationship**. Unlike other scene graph datasets [79, 19], relationships in MOMA-LRG can involve two or more actors. To do this, we provide **hyperedge** annotations where higher-order interactions involving multiple entities are grouped into a single edge (e.g. multiple actors `beneath` an object). Note that multi-node edges can easily be converted to a set of binary edges if needed.

Intuition and advantages. The activity graph is a single universal representation of human activities, consisting of three levels of hierarchy ranging from coarse to fine-grained: activity, sub-activity, and atomic action. This is inspired by the fact that complex human activities in real-world settings are usually hierarchical and compositional across space and time. In particular, complex human activities typically involve a number of achievable steps (activity $\rightarrow$ sub-activity). It is also essential to understand the roles of actors, the affordances of objects, and the relationships between these components in order to recognize fine-grained activities (sub-activity $\rightarrow$ atomic action). In contrast, many existing activity recognition benchmarks and tasks [2, 15, 89] only focus on a specific level of granularity.

3.2 The MOMA-LRG Dataset

Dataset statistics. MOMA-LRG provides annotations on 1,412 activity instances from 20 different unique activity classes and 15,842 sub-activity instances from 92 unique classes. We provide 634,194 image-level actor instances and 104,594 video-level actor instances from 27 unique classes, as well as 349,244 image-level object instances and 47,523 video-level object instances from 269 unique classes. On the atomic action level, there are 1,037,319 higher order relationships from 52 classes and 704,230 attribute instances from 13 classes. Overall, there are 161,265 image-level atomic action instances and 15,842 video-level atomic action instances.

Language-Refined Graphs. One of MOMA-LRG's distinguishing features is that it enables few-shot capabilities. To do this, we provide graphical annotations that are easily compatible with natural language through two conventions. First, predicate classes are of the form `[src] [predicate] [trg]`, where `src` is the source entity and `trg` the target entity. This enables easy conversion to natural language given graphical annotations. For example, given an outgoing predicate edge with class `[src] talking to [trg]` from the entity `cashier` onto the entity `customer`, we can produce the sentence `the cashier is talking to a customer`. Second, all of our annotations are in the present continuous tense, e.g. `the player is throwing a frisbee`, which resembles a live narration in a fashion similar to existing video-language datasets (e.g. YouCook2 [90], HowTo100M [9], etc.) created from instructional YouTube videos.

Comparison with existing datasets. Compared to existing datasets, there are several key advantages that the MOMA-LRG dataset provides. First, MOMA-LRG grounds all associated entities. In contexts with more than one entity, it is necessary to disambiguate which entities are involved in a particular interaction. Existing ego-centric datasets [16, 34] dodge this issue since at most interaction is involved in a scene. Second, we classify each actor's role. Typical datasets [34, 43] involve one type of actor and hence do not label the person's role [42, 19, 21]. In a diverse set of scenes, the role of the actor becomes more important in understanding actions since it can provide an important signal in parsing a human activity [91]. Third, the MOMA-LRG dataset differentiates between static and dynamic predicates. For example, the dynamic predicate `sitting down` is a dynamic movement where an actor transitions from the `standing` static predicate to the `sitting` static predicate. We argue that observing state transitions is important for the model to learn, encouraging it to learn perceptual causality [92]. For a more detailed comparison, refer to Table 1.

Comparison with MOMA [20]. First of all, MOMA-LRG introduces a new dataset and a new abstraction of human activity. MOMA-LRG contains an order of magnitude more annotations, along with longer videos from a greater variety of scenes. In addition, MOMA-LRG the introduction of activity graphs as the overarching graphical representation across all three levels of hierarchy, as opposed to only the atomic level. Secondly, MOMA-LRG is directly motivated by the rise and limitations of VLMs. While VLMs have demonstrated remarkable generalization on videos from new domains and improved scalability through training on free video-language pairs, there is a lack of a single overarching task for evaluating VLMs on complex activity recognition. MOMA-LRG introduces a new annotation schema that can be easily converted from natural language to graphical annotations to enable few-shot learning, and a new framework (GraphVLM) to evaluate VLMs on activity parsing.

Ethics. Prior research [93] shows that Youtube videos exhibit geographic bias. To mitigate potential ethical issues associated with the dataset, we have adopted the following protocols: (1) The videos are sourced from YouTube, where each video has been cross-verified by multiple annotators to ensure that it does not contain offensive material. (2) Instead of providing the raw videos, we provide YouTube IDs and a script for downloading them. Note that this is a standard practice for many video datasets collected from the Internet, such as ActivityNet [15] and Kinetics [2]. (3) Since we only provide YouTube IDs, only publicly available videos are available for download, and video owners may disable accessibility at any time. (4) Our group will verify any image data released for demonstration purposes, and in order to minimize the risk of personally identifiable information being compromised, we blur faces whenever possible.

4 Activity Parsing and the GraphVLM Model

In this section, we introduce a method for performing activity parsing and provide a transfer learning framework to adapt Video-Language Models (VLMs) to activity parsing.

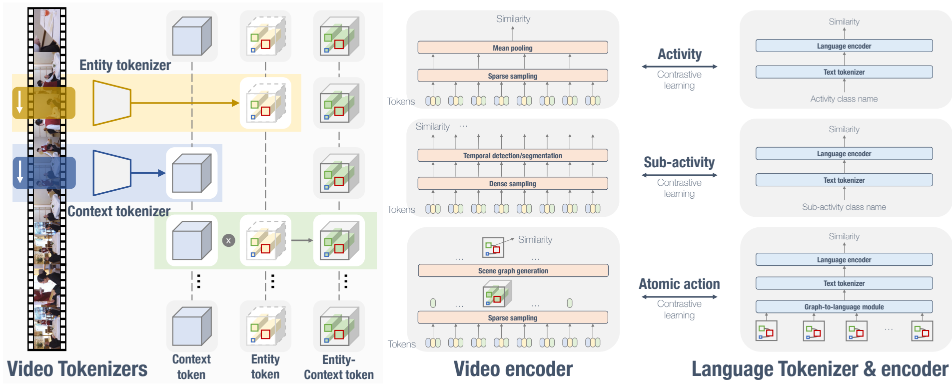

Figure 2: The GraphVLM model: We utilize two different tokenizers for entities and contexts, and a task specific head for each level of the MOMA hierarchy.

4.1 Activity Parsing

We define activity parsing as the task of generating an activity graph as defined in Section 3.1. Specifically, given a video as input, an activity parsing model (1) returns an activity class label, (2) temporally localizes and classifies all sub-activities, as well as (3) localizes each entity in the scene. Following this, it will need to detect all predicates: i.e. all unary predicates (i.e., attributes) involving a single entity and all binary predicates (i.e. relationships) which are between pairs of distinct entities. Refer to Figure 1 for an example of the end results of activity parsing.

4.2 GraphVLM: Video Stream

Our video stream module consists of two tokenizers and an encoder for each level of the activity hierarchy.

Tokenization. The first tokenizer is the context tokenizer, which consists of a clip sampler and a clip feature extractor. The clip sampler takes a video as input and samples non-overlapping short video clips. It has two parameters: the number of sampled frames T and temporal stride τ_c, meaning that each sampled clip consists of $T \times \tau_c$ total frames. For our clip feature extractor, we evaluate on a Swin-B [29], MViT-B [94], and SlowFast-R50 [25] pre-trained on Kinetics-400 [2]. The second tokenizer is the entity tokenizer, which consists of a frame sampler and an entity detector. The role of the frame sampler is to uniformly sample frames across the whole video, parameterized by τ_e, where τ_e is the temporal stride (i.e. we sample one frame every τ_e frames). After frame sampling, we detect entities at the frame-level and generate bounding boxes as well as ROI features associated with each entity, which we call the entity token. These two tokens are used to generate an *entity-context token* by applying the bounding boxes extracted from entity tokenization to the context tokens through ROIAlign.

Entity detection. To detect entities and extract entity tokens for activity parsing, we use a Faster-RCNN [89] object detector with a ResNet-101 [95] FPN [96] backbone pre-trained on ImageNet [1]. We use maskrcnn-benchmark [1] for our implementation. For our activity parsing experiment, we treat all human role classes as a single class to facilitate downstream predicate classification. Separately, we also experiment with object detection using the role classes in our dataset. We use Detectron2 [2] for actor role detection, and use pre-trained weights from COCO keypoints[97], with the same architecture described above. We show results for entity detection in Table 4.

Activity encoding. The video encoder for activity videos *sparsely* samples N_a context tokens produced by the context tokenizer and performs a mean pool to get the activity feature. This encoding works both with and without a text stream: using the features, we can train an action classifier

[1] github.com/facebookresearch/maskrcnn-benchmark
[2] github.com/facebookresearch/detectron2

using a cross-entropy loss exclusively on the features or train jointly with the text stream utilizing a contrastive loss.

Sub-activity encoding. The video encoder for sub-activity videos *densely* samples N_s context tokens which are used to run either temporal action detection or segmentation. For temporal action detection, we input the context tokens into a G-tad [35] model and train a model to predict temporal boundaries using a cross entropy loss. For temporal action segmentation, the encoding is flexible to work with and without the text stream: we can train a classifier with cross-entropy loss using only the features and classify each token as belonging to a sub-activity class or a background class and also train jointly with a text stream using a contrastive loss.

Atomic action encoding. Atomic action encoding consists of two parts: per-frame scene graph generation and spatio-temporal atomic action detection. For scene graph generation, relationships are grounded over all entities in the scene, whereas spatio-temporal atomic action detection is actor-centric and only considers a single actor at a time. We use entity tokens (i.e. object labels, bounding boxes, and ROI features) as input for scene graph detection. We train a RelDN [98] model for scene graph detection using Microsoft's Scene Graph Benchmark [3] for our implementation. We evaluate our model on the tasks of predicate classification, scene graph classification, and scene graph detection without graphical constraints as in Xu et al. [99], since the MOMA-LRG dataset often contains multiple relationships for a given source and target entity. Results are shown in Table 4. For spatio-temporal action detection, the entity-context token of an actor object is taken as input, and the model outputs frame-level predicate labels for the actor in a multi-label classification setting. In our implementation, we train a single layer classifier with a sigmoid activation function, though we note that our framework is compatible with using the generated natural language predicate sentences as supervision via contrastive learning. We show results in Table 4.

4.3 GraphVLM: Text Stream

In order to effectively leverage the natural language capabilities of VLMs, we convert all levels of the MOMA-LRG activity graph hierarchy to natural language via our *graph-to-language* module.

Graph-to-language module. At the activity level, each class name is a noun, thus it can be represented by its class name or via prompting (e.g. by prepending `"A video of [CLS_NAME]"`). At the sub-activity level, class names are descriptions of the sub-activities in the present continuous tense (narration-style). At the atomic action level, we tag all predicates with `[src]`, and `[trg]` templates to allow for easy conversion into a full grammatically correct sentences in present continuous form. For example, the predicate `touching` is represented as `[src] touching [trg]`. So, given the entities `[src]=person` and `[trg]=table`, the sentence is `A person is touching the table`.

Text encoding. After converting the associated activity graph level to language, we use a pre-trained language model to encode the text. When evaluating existing VLMs (e.g. VideoCLIP [12], FiT [8]) using our framework, we use their respective text encoders. For our model agnostic use-case, we use bert-base-uncased [7].

4.4 GraphVLM: Few-shot and Transfer Learning

MOMA-LRG includes a few-shot split, which splits the MOMA dataset into non-overlapping activity and sub-activity classes. The few-shot training set contains 10 activity classes and 45 sub-activity classes, the validation set contains 5 activity classes and 24 sub-acitivity classes, and the test set contains 5 activity classes and 22 sub-activity classes. For our baseline methods, we report results using two meta-learning classifiers, OTAM [100] and CMN [101]. To evaluate the performance of video-language models in the few shot setting, we perform out of the box classification for a VideoCLIP [12] and a Frozen-In-Time [8] video-language model on activity and sub-activity classification on the meta-test set. We use class names as text (either activity or sub-activity) and raw videos as input. To compute the class label for a video input, we find the text embedding that is closest

[3] github.com/microsoft/scene_graph_benchmark

Table 2: Temporal action detection. AP is reported at thresholds 0.1, 0.3, and 0.5 for different backbones.

		Temporal Detection		
Backbones	Pre-train	AP@0.1	AP@0.3	AP@0.5
MVIT-B	K-400+LVIS	17.906	9.369	5.107
SlowFast-R50	K-400+LVIS	21.797	11.782	4.904
Swin-B	K-400+LVIS	22.102	10.853	4.860

Table 3: Video classification. Results are reported for activity and sub-activity classification with different video backbones.

			Activity		Sub-activity	
Model	$T \times \tau$	Pre-train	acc@1	acc@5	acc@1	acc@5
MVIT-B	16×4	Kinetics-400	0.7731	0.9468	0.6032	0.9473
	16×4	None	0.5140	0.8010	0.4375	0.7500
SlowFast-R50	8×8	Kinetics-400	0.7569	0.9375	0.5625	0.9226
	8×8	None	0.4375	0.7500	0.3739	0.7731
Swin-B	4×3	Kinetics-400	0.8576	0.9688	0.6450	0.9781
	4×3	None	0.5282	0.8415	0.3817	0.7868
GCN	30×1	None	0.7837	0.9539	0.3829	0.8276
GCN (oracle bbox)	30×1	None	0.9502	0.9964	0.563	0.9706

in dot product similarity to the video embedding as in Xu et al. [12]. We visualize the performance of this method on the regular MOMA-LRG test set in Figure 3. For k-shot video classification, we sample k videos per class and average the representation to obtain a prototype video. We compute a weighted average between the text embedding and the video prototype and classify using the same method as in the zero-shot setting. A more detailed explanation and ablation study of our method can be found in the Appendix.

In addition to our framework for evaluating VLMs without training, we also propose a method for using VLMs for activity parsing in a more flexible manner than full fine-tuning. We use knowledge distillation to incorporate visual and linguistic knowledge from VLMs into the activity parsing framework. This is considerably more flexible than full fine-tuning since it is model-agnostic. In this method, we can use a different backbone network than that used by the VLMs, and can sample clips differently so long as the clips from the student and the teacher model are centered at the same frame. We report results using our framework and investigating the effect of incorporating linguistic information for spatio-temporal atomic action detection in Table 4. Details for our approach are explained in the Appendix.

5 Activity Parsing Evaluation

In this section, we evaluate our dataset on methods across two different tasks. First, we evaluate model performance on the activity parsing task which leverages the hierarchy of our dataset. Next, we examine our method on activity parsing in the few-shot setting.

We evaluate each level of activity parsing using the following metrics.

Activity: Activity performance is measured by the top 1 accuracy (acc@1) and top 5 accuracy (acc@5) for video-level activity classification.

Sub-activity: Sub-activity performance is measured by two metrics. First, we report the sub-activity acc@1 and acc@5 (where the pre-segmented sub-activity video is used as input). Second, we evaluate temporal detection using mAP at thresholds 0.1, 0.3, and 0.5 and report the average mAP following [33].

Atomic action: We report atomic action performance using several metrics. First, we evaluate entity detection using standard average prevision (AP) metrics. To evaluate scene graph generation, we follow work in [79, 19, 99] and perform the following tasks: predicate classification (PredCls), which takes ground truth bounding boxes and object categories as input and returns predicate labels, scene graph classification (SGCls) which only takes in ground truth bounding boxes as input and predicts object categories and predicate labels, and scene graph detection (SGDet) which simply takes in an input image and predicts bounding box locations, object categories, and predicate labels.

6 Conclusion

We introduce the MOMA-LRG dataset, a large activity recognition dataset of complex human activities that enables evaluation and fine-tuning of large, generalizable video-language models.

Table 4: Scene graph, entity, and sub-activity detection results from the methods described in Section 4.2. Our results show that sub-activity detection and entity detection are challenging in the MOMA-LRG dataset. The entity detection results pose challenges for scene graph detection, as is evidenced by the relatively higher scores for SGCls and PredCls, where ground truth bounding boxes and class labels are known.

	Scene Graph Detection		
	Recall@20	Recall@50	Recall@100
PredCls	58.2243	62.4389	64.0983
SGCls	44.3065	48.3825	50.2992
SGDet	37.6275	43.8594	47.9960

	Entity Detection					
	AP	AP50	AP75	APs	APm	APl
Actor role	38.3567	58.1256	41.2369	7.8053	19.4897	40.0392
Entity	15.2896	29.3032	14.0742	4.3134	10.1288	17.1472

	Temporal Detection	
	Video only	Video + text stream
MViT-B	0.2130	0.2353
SlowFast-R50	0.1975	0.2023

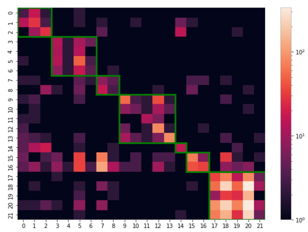

Figure 3: A confusion matrix for zero-shot classification of VideoCLIP [12] on the standard MOMA-LRG sub-activity test set. The sub-activities are ordered to be adjacent to other sub-activities within the same activity. As is indicated by the green squares and the results in Table 5, there is a significant degree of within-activity confusion for zero-shot video language models.

Table 5: Low-shot video classfication. We evaluate both VideoCLIP [12] and Frozen-in-Time [8] within our few-shot framework for activity and sub-activity classification. We note that although the video-language models we tested performed well for high-level activity classification, they performed significantly worse for the more granular task of sub-activity classification.

	Activity			Sub-activity		
Model	0-shot	1-shot	5-shot	0-shot	1-shot	5-shot
OTAM [100]	-	80.71	92.07	-	**57.14**	**72.59**
CMN [101]	-	73.57	86.30	-	52.30	66.60
VideoCLIP [12]	75.90	84.40	84.80	**30.80**	32.70	32.70
Frozen [8]	**90.80**	**92.30**	**92.50**	19.10	26.50	26.30

We define activity parsing as the overarching task of activity graph generation, requiring video understanding at multiple levels of granularity. We demonstrate the capacity of MOMA-LRG to train video-language models by introducing a model-agnostic and lightweight approach for adaptation, and we evaluate VLMs by demonstrating strong few-shot classification performance. We hope that MOMA-LRG will enable further research into generalizable activity recognition models that are trained with multiple input modalities or generating language descriptions for videos.

References

[1] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[2] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, Mustafa Suleyman, and Andrew Zisserman. The kinetics human action video dataset, 2017.

[3] Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

[4] 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 *ICML*, 2021.

[5] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[6] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pages 8821–8831. PMLR, 2021.

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

[8] 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.

[9] 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 *ICCV*, 2019.

[10] Hu Xu, Gargi Ghosh, Po-Yao Huang, Prahal Arora, Masoumeh Aminzadeh, Christoph Feichtenhofer, Florian Metze, and Luke Zettlemoyer. Vlm: Task-agnostic video-language model pre-training for video understanding. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4227–4239, 2021.

[11] 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.

[12] 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. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6787–6800, 2021.

[13] 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, 2021.

[14] Linjie Li, Yen-Chun Chen, Yu Cheng, Zhe Gan, Licheng Yu, and Jingjing Liu. Hero: Hierarchical encoder for video+ language omni-representation pre-training. *arXiv preprint arXiv:2005.00200*, 2020.

[15] Bernard Ghanem Fabian Caba Heilbron, Victor Escorcia and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 961–970, 2015.

[16] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, et al. The" something something" video database for learning and evaluating visual common sense. In *Proceedings of the IEEE international conference on computer vision*, pages 5842–5850, 2017.

[17] Antoine Miech, Jean-Baptiste Alayrac, Lucas Smaira, Ivan Laptev, Josef Sivic, and Andrew Zisserman. End-to-End Learning of Visual Representations from Uncurated Instructional Videos. In *CVPR*, 2020.

[18] Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. Videobert: A joint model for video and language representation learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[19] Jingwei Ji, Ranjay Krishna, Li Fei-Fei, and Juan Carlos Niebles. Action genome: Actions as compositions of spatio-temporal scene graphs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10236–10247, 2020.

[20] Zelun Luo, Wanze Xie, Siddharth Kapoor, Yiyun Liang, Michael Cooper, Juan Carlos Niebles, Ehsan Adeli, and Fei-Fei Li. Moma: Multi-object multi-actor activity parsing. *Advances in Neural Information Processing Systems*, 34, 2021.

[21] Nishant Rai, Haofeng Chen, Jingwei Ji, Rishi Desai, Kazuki Kozuka, Shun Ishizaka, Ehsan Adeli, and Juan Carlos Niebles. Home action genome: Cooperative compositional action understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11184–11193, 2021.

[22] Sungjin Park, Seongsu Bae, Jiho Kim, Tackeun Kim, and Edward Choi. Graph-text multi-modal pre-training for medical representation learning. In *Conference on Health, Inference, and Learning*, pages 261–281. PMLR, 2022.

[23] Hadas Kress-Gazit, Georgios E Fainekos, and George J Pappas. Translating structured english to robot controllers. *Advanced Robotics*, 22(12):1343–1359, 2008.

[24] Calin Belta, Antonio Bicchi, Magnus Egerstedt, Emilio Frazzoli, Eric Klavins, and George J Pappas. Symbolic planning and control of robot motion [grand challenges of robotics]. *IEEE Robotics & Automation Magazine*, 14(1):61–70, 2007.

[25] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6202–6211, 2019.

[26] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017.

[27] Christoph Feichtenhofer. X3d: Expanding architectures for efficient video recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 203–213, 2020.

[28] Yanyi Zhang, Xinyu Li, Chunhui Liu, Bing Shuai, Yi Zhu, Biagio Brattoli, Hao Chen, Ivan Marsic, and Joseph Tighe. Vidtr: Video transformer without convolutions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13577–13587, 2021.

[29] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer. *arXiv preprint arXiv:2106.13230*, 2021.

[30] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6836–6846, 2021.

[31] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding. *arXiv preprint arXiv:2102.05095*, 2(3):4, 2021.

[32] Roei Herzig, Elad Ben-Avraham, Karttikeya Mangalam, Amir Bar, Gal Chechik, Anna Rohrbach, Trevor Darrell, and Amir Globerson. Object-region video transformers. *ArXiv*, abs/2110.06915, 2021.

[33] Haroon Idrees, Amir R Zamir, Yu-Gang Jiang, Alex Gorban, Ivan Laptev, Rahul Sukthankar, and Mubarak Shah. The thumos challenge on action recognition for videos “in the wild”. *Computer Vision and Image Understanding*, 155:1–23, 2017.

[34] Dian Shao, Yue Zhao, Bo Dai, and Dahua Lin. Finegym: A hierarchical video dataset for fine-grained action understanding. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2616–2625, 2020.

[35] Mengmeng Xu, Chen Zhao, David S Rojas, Ali Thabet, and Bernard Ghanem. G-tad: Sub-graph localization for temporal action detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10156–10165, 2020.

[36] Tianwei Lin, Xiao Liu, Xin Li, Errui Ding, and Shilei Wen. Bmn: Boundary-matching network for temporal action proposal generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3889–3898, 2019.

[37] Zheng Shou, Dongang Wang, and Shih-Fu Chang. Temporal action localization in untrimmed videos via multi-stage cnns. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1049–1058, 2016.

[38] Yuanjun Xiong, Yue Zhao, Limin Wang, Dahua Lin, and Xiaoou Tang. A pursuit of temporal accuracy in general activity detection. *arXiv preprint arXiv:1703.02716*, 2017.

[39] Jiyang Gao, Zhenheng Yang, Kan Chen, Chen Sun, and Ram Nevatia. Turn tap: Temporal unit regression network for temporal action proposals. In *Proceedings of the IEEE international conference on computer vision*, pages 3628–3636, 2017.

[40] Humam Alwassel, Fabian Caba Heilbron, and Bernard Ghanem. Action search: Spotting actions in videos and its application to temporal action localization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 251–266, 2018.

[41] Shyamal Buch, Victor Escorcia, Bernard Ghanem, Li Fei-Fei, and Juan Carlos Niebles. End-to-end, single-stream temporal action detection in untrimmed videos. In *Procedings of the British Machine Vision Conference 2017*. British Machine Vision Association, 2019.

[42] Chunhui Gu, Chen Sun, David A Ross, Carl Vondrick, Caroline Pantofaru, Yeqing Li, Sud-heendra Vijayanarasimhan, George Toderici, Susanna Ricco, Rahul Sukthankar, et al. Ava: A video dataset of spatio-temporally localized atomic visual actions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6047–6056, 2018.

[43] Yixuan Li, Lei Chen, Runyu He, Zhenzhi Wang, Gangshan Wu, and Limin Wang. Multisports: A multi-person video dataset of spatio-temporally localized sports actions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13536–13545, 2021.

[44] Chao-Yuan Wu, Christoph Feichtenhofer, Haoqi Fan, Kaiming He, Philipp Krahenbuhl, and Ross Girshick. Long-term feature banks for detailed video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[45] Yutong Feng, Jianwen Jiang, Ziyuan Huang, Zhiwu Qing, Xiang Wang, Shiwei Zhang, Mingqian Tang, and Yue Gao. Relation modeling in spatio-temporal action localization. *arXiv preprint arXiv:2106.08061*, 2021.

[46] Junting Pan, Siyu Chen, Zheng Shou, Jing Shao, and Hongsheng Li. Actor-context-actor relation network for spatio-temporal action localization. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 464–474, 2021.

[47] Hassan Akbari, Liangzhe Yuan, Rui Qian, Wei-Hong Chuang, Shih-Fu Chang, Yin Cui, and Boqing Gong. Vatt: Transformers for multimodal self-supervised learning from raw video, audio and text. In *Advances in Neural Information Processing Systems*, 2021.

[48] Junxian He, Chunting Zhou, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. Towards a unified view of parameter-efficient transfer learning. *arXiv preprint arXiv:2110.04366*, 2021.

[49] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *arXiv preprint arXiv:2107.13586*, 2021.

[50] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *ICML*, 2019.

[51] Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597, 2021.

[52] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017.

[53] Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. A closer look at few-shot classification. In *International Conference on Learning Representations*, 2018.

[54] Yi-Lin Sung, Jaemin Cho, and Mohit Bansal. Vl-adapter: Parameter-efficient transfer learning for vision-and-language tasks. *arXiv preprint arXiv:2112.06825*, 2021.

[55] Peng Gao, Shijie Geng, Renrui Zhang, Teli Ma, Rongyao Fang, Yongfeng Zhang, Hongsheng Li, and Yu Qiao. Clip-adapter: Better vision-language models with feature adapters. *arXiv preprint arXiv:2110.04544*, 2021.

[56] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *arXiv preprint arXiv:2109.01134*, 2021.

[57] Yuan Yao, Ao Zhang, Zhengyan Zhang, Zhiyuan Liu, Tat-Seng Chua, and Maosong Sun. Cpt: Colorful prompt tuning for pre-trained vision-language models. *arXiv preprint arXiv:2109.11797*, 2021.

[58] Chen Ju, Tengda Han, Kunhao Zheng, Ya Zhang, and Weidi Xie. Prompting visual-language models for efficient video understanding. *arXiv preprint arXiv:2112.04478*, 2021.

[59] Jingen Liu, Benjamin Kuipers, and Silvio Savarese. Recognizing human actions by attributes. In *CVPR 2011*, pages 3337–3344. IEEE, 2011.

[60] Yanwei Fu, Timothy M Hospedales, Tao Xiang, and Shaogang Gong. Attribute learning for understanding unstructured social activity. In *European Conference on Computer Vision*, pages 530–543. Springer, 2012.

[61] Biagio Brattoli, Joseph Tighe, Fedor Zhdanov, Pietro Perona, and Krzysztof Chalupka. Rethinking zero-shot video classification: End-to-end training for realistic applications. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4613–4623, 2020.

[62] Mina Bishay, Georgios Zoumpourlis, and Ioannis Patras. Tarn: Temporal attentive relation network for few-shot and zero-shot action recognition. *arXiv preprint arXiv:1907.09021*, 2019.

[63] Ioannis Alexiou, Tao Xiang, and Shaogang Gong. Exploring synonyms as context in zero-shot action recognition. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 4190–4194. IEEE, 2016.

[64] AJ Piergiovanni and Michael S Ryoo. Learning shared multimodal embeddings with unpaired data. *CoRR*, 2018.

[65] Bowen Zhang, Hexiang Hu, and Fei Sha. Cross-modal and hierarchical modeling of video and text. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 374–390, 2018.

[66] Pallabi Ghosh, Nirat Saini, Larry S Davis, and Abhinav Shrivastava. All about knowledge graphs for actions. *arXiv preprint arXiv:2008.12432*, 2020.

[67] Devraj Mandal, Sanath Narayan, Sai Kumar Dwivedi, Vikram Gupta, Shuaib Ahmed, Fahad Shahbaz Khan, and Ling Shao. Out-of-distribution detection for generalized zero-shot action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9985–9993, 2019.

[68] Valter Estevam, Helio Pedrini, and David Menotti. Zero-shot action recognition in videos: A survey. *Neurocomputing*, 439:159–175, 2021.

[69] Han-Jia Ye, Hexiang Hu, De-Chuan Zhan, and Fei Sha. Few-shot learning via embedding adaptation with set-to-set functions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8808–8817, 2020.

[70] Chi Zhang, Yujun Cai, Guosheng Lin, and Chunhua Shen. Deepemd: Few-shot image classification with differentiable earth mover's distance and structured classifiers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12203–12213, 2020.

[71] Kaidi Cao, Jingwei Ji, Zhangjie Cao, Chien-Yi Chang, and Juan Carlos Niebles. Few-shot video classification via temporal alignment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10618–10627, 2020.

[72] Linchao Zhu and Yi Yang. Label independent memory for semi-supervised few-shot video classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(1):273–285, 2022.

[73] Chris Careaga, Brian Hutchinson, Nathan Hodas, and Lawrence Phillips. Metric-based few-shot learning for video action recognition. *arXiv preprint arXiv:1909.09602*, 2019.

[74] Zhenxi Zhu, Limin Wang, Sheng Guo, and Gangshan Wu. A closer look at few-shot video classification: A new baseline and benchmark. *arXiv preprint arXiv:2110.12358*, 2021.

[75] Xiatian Zhu, Antoine Toisoul, Juan-Manuel Perez-Rua, Li Zhang, Brais Martinez, and Tao Xiang. Few-shot action recognition with prototype-centered attentive learning. *arXiv preprint arXiv:2101.08085*, 2021.

[76] Hongguang Zhang, Li Zhang, Xiaojuan Qi, Hongdong Li, Philip HS Torr, and Piotr Koniusz. Few-shot action recognition with permutation-invariant attention. In *European Conference on Computer Vision*, pages 525–542. Springer, 2020.

[77] Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. Dense-captioning events in videos. In *Proceedings of the IEEE international conference on computer vision*, pages 706–715, 2017.

[78] Luowei Zhou, Yannis Kalantidis, Xinlei Chen, Jason J Corso, and Marcus Rohrbach. Grounded video description. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6578–6587, 2019.

[79] 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(1):32–73, 2017.

[80] Fei Yu, Jiji Tang, Weichong Yin, Yu Sun, Hao Tian, Hua Wu, and Haifeng Wang. Ernie-vil: Knowledge enhanced vision-language representations through scene graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 3208–3216, 2021.

[81] Shizhe Chen, Yida Zhao, Qin Jin, and Qi Wu. Fine-grained video-text retrieval with hierarchical graph reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10638–10647, 2020.

[82] Amir Bar, Roei Herzig, Xiaolong Wang, Anna Rohrbach, Gal Chechik, Trevor Darrell, and Amir Globerson. Compositional video synthesis with action graphs. *arXiv preprint arXiv:2006.15327*, 2020.

[83] Roei Herzig, Elad Levi, Huijuan Xu, Hang Gao, Eli Brosh, Xiaolong Wang, Amir Globerson, and Trevor Darrell. Spatio-temporal action graph networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019.

[84] Xiaolong Wang and Abhinav Gupta. Videos as space-time region graphs. In *Proceedings of the European conference on computer vision (ECCV)*, pages 399–417, 2018.

[85] Ang Li, Meghana Thotakuri, David A. Ross, João Carreira, Alexander Vostrikov, and Andrew Zisserman. The ava-kinetics localized human actions video dataset. *CoRR*, abs/2005.00214, 2020.

[86] Philippe Weinzaepfel, Xavier Martin, and Cordelia Schmid. Human action localization with sparse spatial supervision. *arXiv preprint arXiv:1605.05197*, 2016.

[87] Kellie Corona, Katie Osterdahl, Roderic Collins, and Anthony Hoogs. Meva: A large-scale multiview, multimodal video dataset for activity detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 1060–1068, January 2021.

[88] Srikanth Malla, Behzad Dariush, and Chiho Choi. TITAN: future forecast using action priors. *CoRR*, abs/2003.13886, 2020.

[89] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015.

[90] Luowei Zhou, Chenliang Xu, and Jason J Corso. Towards automatic learning of procedures from web instructional videos. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[91] Vignesh Ramanathan, Bangpeng Yao, and Li Fei-Fei. Social role discovery in human events. *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2475–2482, 2013.

[92] Amy Fire and Song-Chun Zhu. Learning perceptual causality from video. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 7(2):1–22, 2015.

[93] Anders Brodersen, Salvatore Scellato, and Mirjam Wattenhofer. Youtube around the world: geographic popularity of videos. In *Proceedings of the 21st international conference on World Wide Web*, pages 241–250, 2012.

[94] Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6824–6835, 2021.

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

[96] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.

[97] 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 *European conference on computer vision*, pages 740–755. Springer, 2014.

[98] Ji Zhang, Kevin J Shih, Ahmed Elgammal, Andrew Tao, and Bryan Catanzaro. Graphical contrastive losses for scene graph parsing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11535–11543, 2019.

[99] Danfei Xu, Yuke Zhu, Christopher Bongsoo Choy, and Li Fei-Fei. Scene graph generation by iterative message passing. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3097–3106, 2017.

[100] Kaidi Cao, Jingwei Ji, Zhangjie Cao, Chien-Yi Chang, and Juan Carlos Niebles. Few-shot video classification via temporal alignment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10618–10627, 2020.

[101] Linchao Zhu and Yi Yang. Compound memory networks for few-shot video classification. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 751–766, 2018.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [Yes]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [N/A]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]