Connecting Vision and Language with Video Localized Narratives

Paul Voigtlaender   Soravit Changpinyo   Jordi Pont-Tuset   Radu Soricut   Vittorio Ferrari

Google Research

{voigtlaender,schangpi,jponttuset,rsoricut,vittoferrari}@google.com

Abstract

We propose Video Localized Narratives, a new form of multimodal video annotations connecting vision and language. In the original Localized Narratives [36], annotators speak and move their mouse simultaneously on an image, thus grounding each word with a mouse trace segment. However, this is challenging on a video. Our new protocol empowers annotators to tell the story of a video with Localized Narratives, capturing even complex events involving multiple actors interacting with each other and with several passive objects. We annotated 20k videos of the OVIS, UVO, and Oops datasets, totalling 1.7M words. Based on this data, we also construct new benchmarks for the video narrative grounding and video question answering tasks, and provide reference results from strong baseline models. Our annotations are available at https://google.github.io/video-localized-narratives/.

1. Introduction

Vision and language is a very active research area, which experienced much exciting progress recently [1, 28, 38, 39, 48]. At the heart of many developments lie datasets connecting still images to captions, while grounding some of the words in the caption to regions in the image, made with a variety of protocols [6,20,23,25,29,34,36,44,46,56]. The recent Localized Narratives (ImLN) [36] offer a particularly attractive solution: the annotators describe an image with their voice while simultaneously moving their mouse over the regions they are describing. Speaking is natural and efficient, resulting in long captions that describe the whole scene. Moreover, the synchronization between the voice and mouse pointer yields dense visual grounding for every word. Yet, still images only show one instant in time. Annotating videos would be even more interesting, as they show *entire stories*, featuring a flow of events involving multiple actors and objects interacting with each other.

Directly extending ImLNs to video by letting the annotator move their mouse and talk while the video is playing would lead to a "race against time", likely resulting in following only one salient object. In this paper, we propose

a better annotation protocol which allows the annotator to tell the story of the video in a calm environment (Fig. 1-3). The annotators first watch the video carefully, identify the main actors ("man", "ostrich"), and select a few representative key-frames for each actor. Then, for each actor separately, the annotators tell a story: describing the events it is involved in using their voice, while moving the mouse on the key-frames over the objects and actions they are talking about. The annotators mention the actor name, its attributes, and especially the actions it performs, both on other actors (*e.g.*, "play with the ostrich") and on passive objects (*e.g.*, "grabs the cup of food"). For completeness, the annotators also briefly describe the background in a separate step (bottom row). Working on keyframes avoids the race against time, and producing a separate narration for each actor enables disentangling situations and thus cleanly capturing even complex events involving multiple actors interacting with each other and with several passive objects. As in ImLN, this protocol localizes each word with a mouse trace segment. We take several additional measures to obtain accurate localizations, beyond what was achieved in [36].

We annotated the OVIS [37], UVO [47], and Oops [12] datasets with Video Localized Narratives (VidLNs). These datasets cover a general domain, as opposed to only cooking [10, 62] or first-person videos [16]. Moreover, these videos contain complex scenes with interactions between multiple actors and passive objects, leading to interesting stories that are captured by our rich annotations. In total our annotations span 20k videos, 72k actors, and 1.7 million words. On average, each narrative features transcriptions for 3.5 actors, totaling 75.1 words, including 23.0 nouns, 9.5 verbs, and 8.5 adjectives (Fig. 4). Our analysis demonstrates the high quality of the data. The text descriptions almost always mention objects/actions that are actually present in the video, and the mouse traces do lie inside their corresponding object in most cases.

The richness of our VidLNs makes them a strong basis for several tasks. We demonstrate this by creating new benchmarks for the tasks of Video Narrative Grounding (VNG) and Video Question Answering (VideoQA). The new VNG task requires a method to localize the nouns in

1. Watch **2./3. Select actors and key-frames** **4. Speak and move mouse** **5. Transcription**

"*A man wearing a black t-shirt is holding a cup of food in his right hand. He moves around a piece of food in his left hand to play with the ostrich.*"

A man wearing a black t-shirt is holding a cup of food in his right hand. He moves around a piece of food in his left hand to play with the ostrich.

Figure 1. The five steps of the annotation process for VidLNs. We are able to cleanly describe even complex interactions and events by disentangling the storylines of different actors.

<Man> A man wearing a black t-shirt is holding a cup of food in his right hand. He moves around a piece of food in his left hand to play with the ostrich.

<Ostrich> An ostrich is looking at the piece of food held by the man and suddenly grabs the cup of food and starts eating.

<Background> In the background, there are hills, white barriers, a flag, the sky, and soil on the ground.
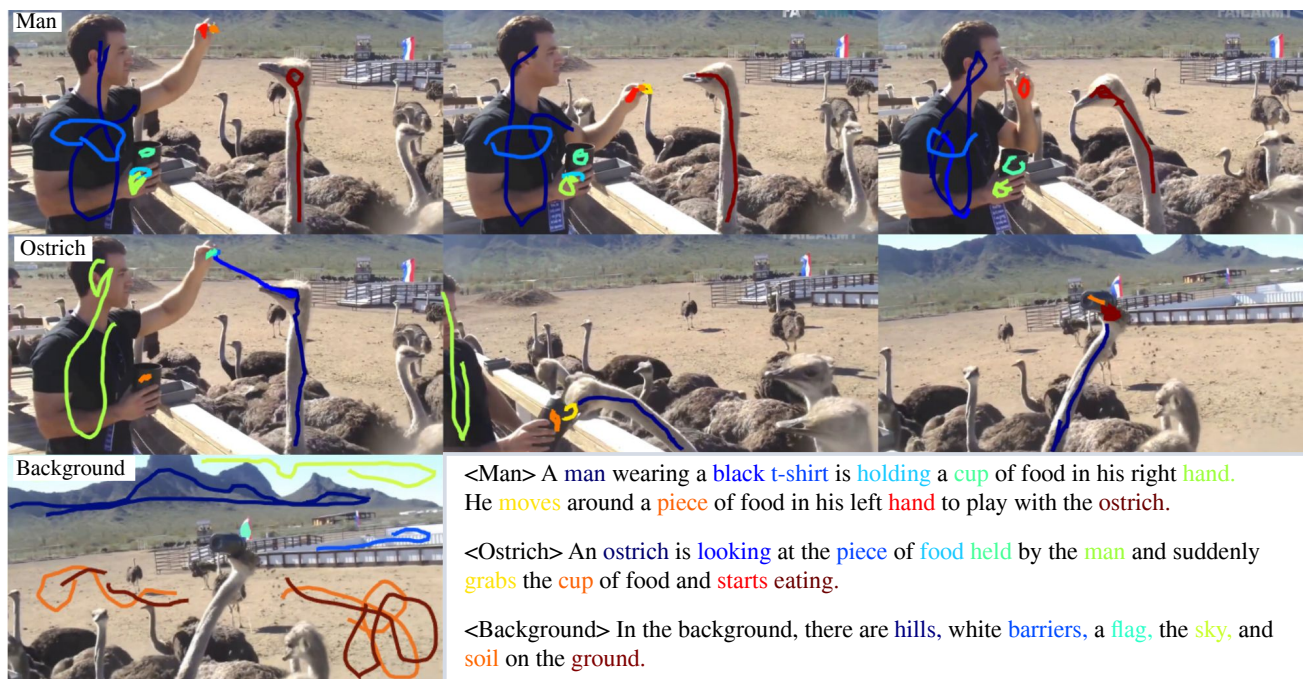
Figure 2. An example VidLN annotation. Each row shows the story of the video from the perspective of a different actor. Each mouse trace segment drawn on selected key-frames localizes the word highlighted in the same color. More examples are in the supplement.

an input narrative with a segmentation mask on the video frames. This is a complex task that goes beyond open-vocabulary semantic segmentation [14], as often the text has multiple identical nouns that need to be disambiguated using the context provided by other words. We construct two VNG benchmarks on OVIS and UVO, with a total of 8k videos involving 45k objects. We also build baseline models that explicitly attempt this disambiguation and report experiments showing it's beneficial on this dataset, while also highlighting these new benchmarks are far from solved.

For VideoQA, we construct a benchmark on the Oops dataset, consisting of text-output questions (*e.g.*, "What is the person in the blue suit doing?"), where the answer is free-form text (*e.g.*, "paragliding"), and location-output questions (*e.g.*, "Where is the woman that is wearing a black and white dress?"), where the answer is a spatio-temporal location in the video. The benchmark features a total of 62k questions on 9.5k videos. Answering many of these questions requires a deep understanding of the whole video. We

also implement baseline methods to establish initial results on this task.

2. Related Work

Datasets Connecting Vision and Language. Many datasets exist that connect vision and language on still images, at different granularities of grounding [6, 20, 23, 25, 29, 34, 36, 44, 46, 56]. In the video domain, there is also a wide range of vision-and-language datasets. Ego4D [16] and Epic-Kitchens [9] are large-scale collections of *daily-life* egocentric videos accompanied with narrations, which were also used as a seed for creating benchmarks for different tasks. We instead focus on third-person view videos that were especially selected to contain complex interactions between multiple actors and passive objects. Our per-actor narration annotation protocol enables disentangling situations and thus cleanly describing such complex videos. Moreover, our descriptions are long story-like narrations connecting various objects, actions, and attributes (*e.g.*,
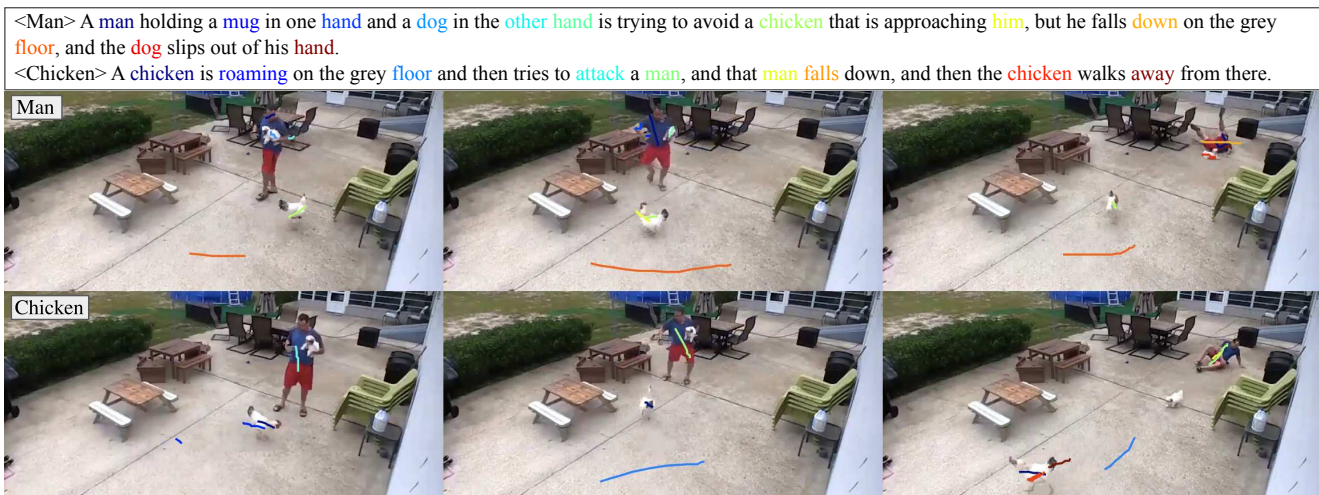
<Man> A man holding a mug in one hand and a dog in the other hand is trying to avoid a chicken that is approaching him, but he falls down on the grey floor, and the dog slips out of his hand.
<Chicken> A chicken is roaming on the grey floor and then tries to attack a man, and that man falls down, and then the chicken walks away from there.

Figure 3. Another VidLN example (also featuring a third actor "Dog", and "Background", not shown). From this VidLN, we automatically generate text-output Q+A pairs, including "What falls out of the man's hand?" with answer "dog".

Fig. 2); instead of short descriptions of atomic actions (*e.g.*, "C closes bottle"). Finally, Ego4D provides grounding only for some nouns, Epic-Kitchens for all nouns, whereas our VidLN annotations ground every word (including adjectives, verbs, *etc.*).

YouCook2-BB [62] and ActivityNet-Entities [61] also contain video descriptions with grounded noun phrases. YouCook2-BB focuses only on cooking, and provides grounding (bounding boxes) only for the test set. ActivityNet-Entities annotations ground the 432 most common nouns mentioned in ActivitiyNet-Captions [24] with bounding boxes. In contrast, our VidLNs ground every word with a mouse trace. Besides, their annotation protocol is simpler: writing sentences describing major events and then finding their corresponding video segments [24]. It does not facilitate the disentanglement between actors or complex actor-object interactions as our protocol does.

Many other datasets [13,19,24,30–32,45,51,53,63] offer textual descriptions for videos, but have no spatial localization annotation of any kind.

Tasks Related to VNG. Panoptic Narrative Grounding (PNG) [15] creates a panoptic segmentation that grounds the nouns of an input caption describing an image. In contrast, our proposed VNG operates on videos and focuses on concrete objects only. ImLNs were automatically combined with the panoptic segmentation annotations in COCO [3,27] to create PNG. We also leveraged pre-existing segmentations, with the key difference that we manually revised and annotated the missing ones, which greatly improves the accuracy and completeness of the benchmark.

In Referring Video Object Segmentation (R-VOS) [21, 43], each referring expression is a short phrase specifically designed to identify one object. In contrast, VNG requires taking a whole description as input and then localizing each noun, which is a more natural formulation of the task and

introduces extra challenges such as co-reference resolution. The same noun can appear multiple times in the description, referring to different objects in the video. These cases must be disambiguated based on context offered by other words.

Video Object Grounding [62] is close to VNG as the expected output is the grounding of the noun phrases in the descriptions, but as a bounding box instead of a segmentation, and only evaluated on the top 63 recurring objects.

Other VideoQA benchmarks. There is a wide spectrum of datasets on Video Question Answering (VideoQA) [60]. Closest to our location-output questions are the so-called *Factoid VideoQA* benchmarks with open-ended answers [60]. Many datasets contain question-answer pairs automatically derived from video descriptions [42, 50, 53–55] without any manual checking, and thus are not suitable as a high-quality evaluation benchmark. Besides, some of them [53, 54] are based on the native audio track of instructional videos, which often mentions objects/verbs that are not actually visible in the video. Other datasets are manually curated: WildQA [4] focuses on outdoor videos, iVQA [53] covers instructional videos, ActivityNet-QA [57] contains various human activities, and VideoQA [59] contains varied web-crawled videos. Our text-output questions are set apart from these works in that (i) they are on videos selected to contain complex interactions between different actors/objects and (ii) they are seeded from dense video narrations that describe the actions from the point of view of these different actors.

Related to our location-output questions are the VideoQA datasets that ground parts of their textual questions and answers (*e.g.*, noun phrases) with bounding boxes on the video [8, 26, 58]. Closest to ours is TVQA+ [8], where the task is to choose one answer out of five and ground it spatially and temporally, on videos from a single TV show. In contrast, our answers can refer to any object,

Dataset	#Videos	#Narratives	#Actors (per narr.)	#Words (per narr.)	Domain	Grounding
OVIS-VidLN	607	610	1,799 (2.95)	28,676 (47.01)	General	Every word
UVO-VidLN	7,588	8,587	25,755 (3.00)	549,303 (63.97)	General	Every word
Oops-VidLN	12,128	12,894	44,422 (3.45)	1,080,211 (83.78)	General	Every word
VidLN All	20,323	22,091	71,976 (3.54)	1,658,190 (75.06)	General	Every word
ActivityNet-Ent [61]	14,926	14,926	-	745,876 (49.97)	General	432 nouns
MPII-MD [40]	94	94	-	647,814 (6,891.7)	Movies	People names
YouCook2-BB [62]	667	667	-	∼40,500 (60.72)	Cooking	67 objects
Ego4D-summary [16]	9,643	18,870	-	165,274 (8.76)	1st-person	None
Ego4D-narration [16]	9,643	18,870	-	23,924,308 (1,267.85)	1st-person	Some nouns
Epic-Kitchens [10]	273	273	-	81,501 (298.54)	1st-p. Cooking	Every noun

Table 1. Statistics of our Video Localized Narrative annotations for three datasets, compared to related datasets. Our datasets focus on a general domain (not movies, cooking, or first-person) and provide grounding for each word.

and our videos come from varied sources chosen to contain complex interactions.

3. Video Localized Narrative Annotations

A trivial extension from ImLNs to video would be to simply let the annotator move their mouse and talk while the video is playing. However, this would lead to a "race against time" likely resulting in following only one salient object. Hence, we introduce a better protocol based on key-frames, that gives the annotator an opportunity to tell the story of the video in a calm environment (Fig. 1-3).

3.1. Annotation Protocol

Our annotation protocol has the following 5 steps.
1. Understand the Video. The annotator watches the video, possibly multiple times, to understand its story.
2. Actor Selection. Some objects carry out actions, such as the man and the ostrich in Fig. 2, as opposed to passive objects receiving the action, such as the cup. The annotator selects and names the actors of this video, *e.g.*, "man" and "ostrich" (these names are mainly a memory aid for the annotator, and not a core part of the annotations).
3. Key-frame Selection. We present the annotator with a set of key-frame candidates, uniformly sampled over time. For each actor, the annotator selects a few key-frames covering the main actions it performs (the story of that actor).
4. A Story for each Actor. This is the main part of the annotation process and it is performed separately for each actor. We show to the annotator the selected key-frames for the actor and they describe the events it is involved in, with full natural language sentences that mention the actor name, its attributes, the actions it performs, other objects and its interactions with them. We instruct the annotators to put special emphasis on the actions performed by the actor, including also actions performed on passive objects (*e.g.*, "grabs the cup of food"). While talking, the annotator moves the mouse pointer on the key-frames over the spatial

positions of the objects and actions they are talking about. For completeness, we ask the annotators to also describe the background in a separate row, albeit in less detail and typically on a single key-frame.

We want to have a good localization for every word, with a mouse trace segment. To achieve this, we explicitly instruct the annotators to talk slowly and to stop talking while moving the mouse between two objects. Additionally, we give the annotators the option to stop a mouse trace by clicking the mouse, to avoid spurious traces, *e.g.*, when moving the mouse between different key-frames or objects. This results in mouse traces that are easy to segment at the beginning and end times of word utterances, leading to well-localized trace segments (see Sec. 3.2 for evaluation).
5. Transcription and Time Alignment. After finishing steps 1-4, the annotator is asked to manually transcribe their voice recording. This ensures the captions are of high quality (as automatic speech transcription can make mistakes).

The annotations for each actor now include a mouse trace, an audio recording, and a manual transcription, but we still need an alignment between words and trace segments to provide localization. To this end, we use an automatic algorithm [2] to align the manual transcription directly to the audio. This yields time-stamps for each word in the transcription, revealing which part of the mouse trace belongs to which word. Note that this direct alignment of audio to text works significantly better than the process used by ImLNs [36] which first invokes an automatic speech-to-text transcription model, and then aligns the automatic transcriptions to the manual ones.

3.2. Statistics

We annotated three datasets with VidLNs: OVIS [37], UVO [47], and Oops [12] (Tab. 1). In total, we annotated about 20k videos with over 1.65 million words, 508k of which are nouns (30.7%), 209k verbs (12.6%), and 188K adjectives (11.3%). In contrast to existing datasets, our narratives feature a separate transcription for each actor, dis-
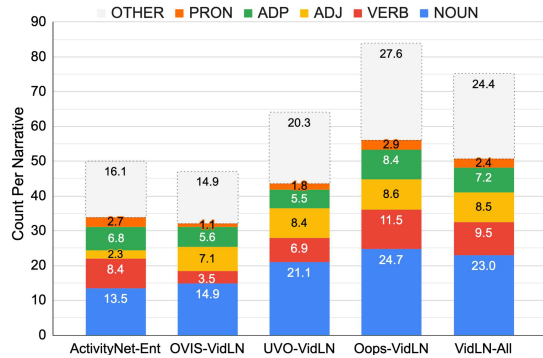
Figure 4. Richness of VidLNs compared to ActivityNet-Entities. We characterize linguistic richness via word type count *per narrative*. PRON: Pronouns, ADP: Adpositions, ADJ: Adjectives. Our Oops-VidLN is the most complex dataset in all major dimensions.

entangling them (3.54 actors per video on average), and we provide a mouse trace grounding for every word. Additionally, our VidLNs cover a general domain rather than only first-person [16] or cooking [10, 62].

In Fig. 4, we compare the per-narrative counts for the 5 main word types to those of ActivityNet-Entities [61], the most related video dataset with grounded captions. On average, our narratives have 75.1 words (including 23.0 nouns, 9.5 verbs, 8.5 adjectives, 7.2 adpositions, 2.4 pronouns). Thus our narratives are longer, and contain more words of each type, than those in ActivityNet-Entities. In summary, we present the largest and richest collection of general-domain video datasets with grounded captions to date.

Semantic Accuracy. We evaluate manually how well the verbs and noun phrases of VidLN describe the actions and objects in the video. We randomly picked 70 videos in each dataset and checked for every verb and noun phrase whether the described object/action is present in the video, and if it is correctly described. Tab. 2 shows the results, demonstrating that the captions are almost perfectly semantically accurate.

Localization Accuracy. We evaluate the localization accuracy of our mouse traces on the OVIS-VNG dataset, where we have ground-truth segmentation masks for many nouns (see Sec. 4). For each such noun we measure what percentage of its mouse trace segment is inside the ground truth mask. We find that the average precision is high at 73.2%. Additionally, we can discard mouse trace segments that consist of multiple disconnected components, as potentially indicative of some error. Note that this can be done automatically without access to the ground-truth masks. This process discards 15.8% of the mouse trace segments, and the remaining ones have an even higher precision of 77.3%.

Now we compare to the localization accuracy of the original ImLNs [36]. For ImLNs, the mouse trace precision measured against ground-truth bounding boxes is 57.6% for Open Images [22] and 54.8% for COCO [27] (as derived from the raw data behind Fig. 4 in [36], which the authors

Accuracy (%)	OVIS-VLN	UVO-VLN	Oops-VLN
Noun phrases	97.7	96.8	97.2
Verbs	96.0	97.8	97.9

Table 2. Semantic accuracy for noun phrases and verbs of our three VidLN datasets.

shared with us). For our VidLNs, the precision measured on bounding boxes is 83.1% (73.2% measured on segmentation masks). This means that our improvements in the annotation protocol have been very successful, and that VidLN traces have much higher quality than ImLN traces (+25%).

4. Video Narrative Grounding (VNG)

Task Definition. We propose the VNG task (Fig. 5), as a first example of the applicability of our VidLN annotations. The input is a video with a text description (the narrative) and the positions of certain nouns marked. For each marked noun, the method must output a segmentation mask for the object it refers to, in each video frame. Importantly, the same noun (*e.g.*, "parrot") can appear multiple times in the description, referring to different objects in the video. The key challenge is to disambiguate those cases, based on the context offered by other words (*e.g.*, "red-black neckline").

As said in Sec. 2, VNG is related to PNG and R-VOS, but PNG is only defined on images, and for R-VOS each referring expression is a short phrase for a single object. Instead VNG inputs a long natural description containing multiple nouns to be disambiguated and segmented.

Evaluation Measure. We adopt the well-established $\mathcal{J}\&\mathcal{F}$ measure [35], which is the average of the mean intersection-over-union $\mathcal{J}$ and the boundary measure $\mathcal{F}$ [33]. Note that different nouns can refer to the same object (*e.g.*, "person" and "hand" in Fig. 5). Hence, overlapping segmentation masks are allowed and the masks for each noun are evaluated separately.

4.1. Benchmarks

We propose two new benchmarks: OVIS-VNG, based on the OVIS dataset [37], and UVO-VNG, based on the UVO dataset [47]. To build such a benchmark, we first need to select nouns in the VidLN captions, and then get ground-truth masks for these nouns. To select nouns we run a noun tagger [18], and then keep only singular nouns of concrete objects (*e.g.*, car, parrot), as opposed to stuff categories (*e.g.*, sky, water). We then get masks for these nouns as follows. OVIS/UVO already have mask annotations for some objects, which we ask annotators to manually match to their corresponding nouns. Finally, we manually annotate masks for all nouns for which OVIS/UVO does not provide one.

As Tab. 3 shows, with 7,587 videos and 43,058 objects, our UVO-VNG has almost 2× as many videos as the largest R-VOS benchmark (Refer-YouTube-VOS), almost 3× total

[Person]: A person is holding the parrots in hand.
[Parrot one]: A green parrot with a red-black neckline is playing with the other parrot.
[Parrot two]: Another green-red parrot is sitting on person's hand.
[background]: In the background, there are green and dry leaves.
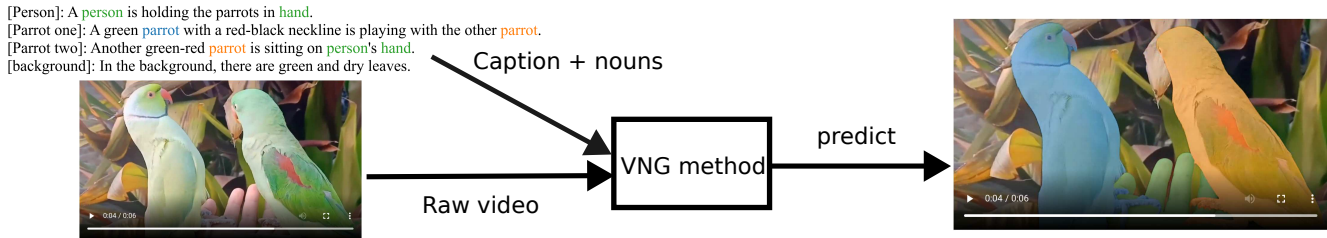
Caption + nouns

Raw video

VNG method

predict

Figure 5. The Video Narrative Grounding (VNG) task. The raw video and the caption together with the position of nouns are given as input to the VNG method, which then predicts segmentation masks for each noun in each frame (masks shown in the same color as the nouns) . Note that the method has to disambiguate between two parrots based on other words, *i.e.*, the red-black neckline.

Benchmark	Task	Videos	Objects (per vid.)
OVIS-VNG	VNG	505	2,407 (4.77)
UVO-VNG	VNG	7,587	43,058 (5.68)
Ref-YTB-VOS [43]	R-VOS	3,978	15,009 (3.77)
Ref-DAVIS'17 [21]	R-VOS	90	205 (2.28)

Table 3. Statistics of our VNG benchmarks compared to R-VOS datasets. "Objects" are not necessarily unique object instances, as multiple nouns or expressions can refer to the same object. For VNG, "Objects" is the number of nouns with ground truth masks. For R-VOS, it is the number of full-video referring expressions.

objects, and about $1.5\times$ objects per video. In addition to the large UVO-VNG benchmark, we propose OVIS-VNG which features many occlusions that make it challenging and its small size makes it well-suited for development.

A key challenge in VNG is to disambiguate multiple occurrences of the same noun in the input description. We analyzed the OVIS training set, which has class label annotations, and found that for 96.4% of the object instances, there is another instance of the same class in the same video. Hence, our benchmark contains substantial ambiguity that can only be resolved using context information.

4.2. ReferFormer-VNG Baseline Method

We modify the ReferFormer [49] method to address our VNG task. The original ReferFormer was designed to tackle the R-VOS task, taking a short phrase describing a single object as input. For VNG, instead the input is a longer caption that can mention different objects and their interactions (Fig. 5), along with the positions of the nouns to be segmented. For the sentence "A green parrot with a red-black neckline is playing with the other parrot", ReferFormer would predict a single mask per frame. For VNG, we instead need a separate mask for each parrot.

ReferFormer uses a visual encoder to extract features from the video. A text encoder extracts text features, from which conditional query features (conditioned on the text) are generated. Both types of features are fed together into a decoder to predict a mask for each frame. The text encoder extracts both per-text-token features and whole-sentence features. The sentence features are used to generate the

conditional queries, which makes sense in the case of R-VOS, where the whole sentence describes a single object. For ReferFormer-VNG, we instead take the features of each text-token that belongs to the noun we want to segment, and average-pool over them to obtain the conditional query (conditioned on the noun). To segment the two different parrots, we run ReferFormer-VNG first using the features belonging to the first "parrot" noun, and a second time using the features of the second occurrence.

4.3. Experiments

Simple baseline. As a simple baseline we use the original ReferFormer. We either input the full narrative as text ("Full narrative") or we pre-process the text input based on the noun which has to be segmented ("Noun") by keeping only the noun and add "a" as prefix, *e.g.*, "a parrot". The results in Tab. 4 show that the "Full Narrative" version performs poorly. This baseline inputs the same text for each noun, and hence is unable to distinguish between different nouns. The "Noun" version tries to force the model to segment the correct noun, which works somewhat better. It essentially gets the class name of the object to be segmented, but cannot differentiate between different instances (*e.g.*, two "parrot"). This explains the modest results, as most OVIS videos contain multiple instances of the same class (*cf*. Sec. 4.1).

ReferFormer-VNG. For ReferFormer-VNG's visual encoder, we use a ResNet-50 backbone [17] initialized on ImageNet [41]. For the best results, we pre-train ReferFormer-VNG on the COCO-PNG dataset which provides annotations for a similar task as VNG, but defined on images instead of videos. Afterwards, we do the main training pass on our new UVO-VNG training set. Finally, we can optionally fine-tune on the OVIS-VNG training set for evaluation on the OVIS-VNG test set.

Tab. 5 shows results for the various training regimes. Generally, these results are superior to those of the simple baselines, showing that ReferFormer-VNG's ability to disambiguate among multiple objects with the same class name is beneficial on this dataset. Using the most training data (first row) leads to the strongest results. Because of having so much training data, the effect of a final fine-tuning pass on OVIS-VNG is small (from 32.0 to 32.7). The

Text pre-proc.	OVIS-VNG	UVO-VNG
Full narrative	22.9	25.8
Noun	25.7	35.6

Table 4. $\mathcal{J}\&\mathcal{F}$ scores for simple baselines for the VNG task based on the original ReferFormer.

COCO-PNG pre-training	UVO-VNG training	OVIS-VNG no ft	OVIS-VNG ft	UVO-VNG
yes	yes	32.0	32.7	46.4
yes	no	28.5	32.4	39.6
no	yes	26.2	29.8	43.1

Table 5. Results of ReferFormer-VNG on the proposed OVIS-VNG and UVO-VNG benchmarks. "no-ft" and "ft" indicate whether we fine-tuned on the OVIS-VNG training set before evaluating on the OVIS-VNG test set. The first two columns indicate different (pre-)training data. All numbers are $\mathcal{J}\&\mathcal{F}$ scores.

second and third rows show that removing either COCO-PNG pre-training or UVO-VNG main-training reduces performance on both test sets. This also shows that training on our proposed UVO-VNG training set is beneficial, both for evaluating on similar data (the UVO-VNG test set), and for evaluating on a different dataset (OVIS-VNG), even without fine-tuning. Implementation details and more experiments can be found in the supplement.

5. Video Question Answering (VideoQA)

As a second example of the applicability of VidLN annotations, we create a Video Question Answering (VideoQA) benchmark. We generate Q+A pairs automatically from the VidLN annotations and only require humans to verify them.

Based on the VidLN annotations for the Oops dataset [12], we create Oops-QA, featuring two kinds of questions: text-output questions and location-output questions (Tab. 6). Below we describe how we generate questions, how we verify them, and present evaluation measures to assess the performance of methods for the two tasks.

5.1. Text-Output Questions

Text-output questions are analog to those in other VideoQA benchmarks [4, 8, 26, 53, 57–59], where the answer is given as free-form text. Questions can for example be about colors ("what color is the cat?"), the number of objects ("How many basketballs are in the background?"), the type of object ("The ball falls out of what?"), or about actions ("what action does the man perform?").

Automatic Q+A Generation. For the caption of each actor of each VidLN, we use the VQ^{2}A method [5] to generate a large pool of Q+A pairs (Fig. 3). This produces about 230 Q+A pairs per video. To facilitate free-form text prediction evaluation, we only keep pairs whose answer has one or two words. Finally, to reduce redundancy we (1) keep only one of multiple questions with the same answer, and (2) remove near-duplicate questions. These steps greatly reduce the number of Q+A pairs to around 22 per video.

Manual Verification. As the Q+A pairs are generated from the VidLN captions that have near-perfect semantic accuracy, they are unlikely to be factually wrong. However, some Q+A pairs are irrelevant, ambiguous, or contain grammatical errors introduced by the generation algorithm. For example: "What color is the sky?" is irrelevant as it can be answered without looking at the video. The question "What color is the cat?" is ambiguous for a video containing two

cats of different colors. Hence, we ask two annotators to independently verify that each Q+A pair is relevant, unambiguous, and grammatically sound. Since only verification is done manually, rather than annotating Q+A from scratch, it can be done with little effort. Manual verification retains 27% of the generated Q+A pairs (about 6 per video).

Evaluation Measures. We evaluate the text answer predicted by a VideoQA method using the exact match accuracy against the ground-truth [57, 60]. The final result is the percentage of correct answers across the dataset.

PaLI Baseline. To establish initial results on this new benchmark, we use the state-of-the-art VideoQA model PaLI [7]. PaLI is an encoder-decoder model that inputs an image and a text prompt, and outputs text. We use the 1.5B version with ViT-L16 [11] (vision) and mT5-Large [52] (language) backbones. At test time, the input is 1-3 video frames and the question as prompt. Using a single frame in the middle of the video as visual input, PaLI-1.5B achieves 24.1% zero-shot accuracy and 44.9% when fine-tuned on the Oops-QA training set. Using 3 equally spaced frames, the results are 25.1% and 49.0%. These results demonstrate the complexity of our benchmark and leave room for the development of even more advanced techniques.

5.2. Location-output Questions

The second part of the Oops-QA benchmark consists of location-output questions. These start with "where is", and their answer is a space-time location in the video (Fig. 6).

Automatic Q+A Generation. For the caption of each actor of a VidLN, we first use spaCy [18] to assign part-of-speech tags for each word (*e.g.*, verb, adjective), and a parse tree for each sentence (*e.g.*, connecting subjects to their verbs). Based on this information, we try to transform each sentence into a question about where the subject is. As sentences can be long, we first use the parse tree to focus on the sub-tree containing the verb and the subject (*e.g.*, "the girl that is wearing a pink dress" in a much longer caption). In this case we prepend "where is" to obtain the question. In other cases, we add "that" or "that is" before the verb (*e.g.*, "where is the dog *that is* playing with the cat"). In this manner we generate about 3.7 questions per video.

For each question, we use the mouse trace segment asso-

Figure 6. An example location-output question of the proposed Oops-QA benchmark: "Where is the girl that is wearing a pink dress?" The ground truth answer is based on a mouse trace segment (green). We use a special evaluation methodology which accounts for the fact that the trace does not cover the whole object.

Oops-QA	Text-output		Location-output	
	train	test	train	test
#Videos	7,509	1,982	8,113	1,691
#Questions (total)	31,760	12,417	14,779	3,179
#Questions (per vid.)	4.23	6.26	1.82	1.88

Table 6. Statistics of our Oops-QA benchmark, divided into text-output questions and location-output questions.

ciated to the subject as a basis to construct a ground-truth answer. To facilitate the subsequent manual verification process, when the trace segment spans multiple frames, we retain only the frame with the longest segment. To improve quality, we only consider trace segments that consist of a single connected component (*cf*. Sec. 3.2).

Manual Verification. We show to two annotators each question together with its trace segment overlaid on the frame selected above. We only keep a Q+A pair if both annotators agree the question is grammatically correct, the trace segment is on the correct object, and the question refers to a unique object in that video. For example, in Fig. 6, "where is the kid?" is ambiguous, whereas "where is the kid with the pink dress" is not. About 52% of the generated Q+A pairs pass this verification step, yielding about 2 per video for the final benchmark. As a side benefit, the localization accuracy of the mouse traces in the verified questions is extremely high (92.9% on the correct object mask on average, see supplement for details).

Evaluation Measures. For each question, a VideoQA method must output a bounding box for every frame where the object is present. However, we evaluate only in the one frame where we have a verified ground-truth trace segment t. For a predicted box b to be correct, it has to fulfill both the precision and the recall criterion described below.

Recall criterion: b has to contain most of t, according to intersection-over-area: $\frac{|b \cap t|}{|t|} \geq 0.5$.

Precision criterion: A mouse trace segment only covers part of an object surface (Fig. 6). To mitigate this effect, we derive from t an approximate ground-truth box g around the object, and use it to evaluate the precision of b (see supplement for how we learn a transformation to enlarge the trace segment into an approximate box around the object). This requires most of b to be contained in g: $\frac{|b \cap g|}{|b|} \geq 0.5$.

Quality of Evaluation Measure. To demonstrate the validity of our evaluation procedure based on approximate ground-truth boxes, we perform an experiment on OVIS-VNG. We simulate a perfect VideoQA method which out-

puts a bounding box on the ground-truth segmentation masks of the OVIS-VNG test set. We now evaluate these perfect predictions against our approximate ground-truth boxes. The result is that 99.1% of the perfectly predicted boxes fulfill the recall criterion above, 85.1% fulfill the precision criterion, and 84.4% fulfill both. Hence, a strong method will be properly rewarded by our evaluation. Additionally, on average 82.3% of the image area is not covered by the approximate ground-truth box. Hence, the task is challenging: as the ground-truth boxes are small on average, a method cannot easily localize them (predictions falling outside the ground-truth are penalized by a low precision).

Baseline Method. We repurpose ReferFormer-VNG (Sec. 4.2) to answer location-output questions. We turn the question into a statement by dropping the "where is", then input it together with the position of the first noun into ReferFormer-VNG to produce a segmentation mask in each video frame. Finally, we convert the masks into bounding boxes, and evaluate them as described above. This baseline fulfills the recall criterion for 66.7% of the questions, the precision criterion for 53.9%, and both for 48.3%. While these results are good, they are far from perfect, showing the challenge posed by our new benchmark and leaving plenty of room for the development of better methods.

5.3. Combined Score

We define a combined score for the whole Oops-QA benchmark as the average of the two sub-tasks. Hence our two baselines together achieve an overall score of 50.8% accuracy (mean of 53.2% and 48.3%). With the unified Oops-QA benchmark of location-output questions and text-output questions, we want to encourage the development of a single model that can answer both kinds of questions and at the same time improve the results.

6. Conclusion

We introduce VidLNs, an annotation procedure that obtains rich video descriptions, that are semantically correct and densely grounded with accurate spatio-temporal localizations. We annotated roughly 20k videos of three different datasets, and obtained captions with more than 1.6 million words in total. We demonstrated the versatility of VidLNs by using them to generate two Video Narrative Grounding benchmarks, and the Oops-QA benchmark.

References

[1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, Andrew Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karen Simonyan. Flamingo: a visual language model for few-shot learning. In *NeurIPS*, 2022. 1

[2] Antoine Bruguier, Danushen Gnanapragasam, Leif Johnson, Kanishka Rao, and Françoise Beaufays. Pronunciation learning with rnn-transducers. In *INTERSPEECH*, pages 2556–2560, 2017. 4

[3] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. COCO-Stuff: Thing and stuff classes in context. In *CVPR*, 2018. 3

[4] Santiago Castro, Naihao Deng, Pingxuan Huang, Mihai Burzo, and Rada Mihalcea. WildQA: In-the-wild video question answering. In *COLING*, 2022. 3, 7

[5] Soravit Changpinyo, Doron Kukliansky, Idan Szpektor, Xi Chen, Nan Ding, and Radu Soricut. All you may need for VQA are image captions. In *NAACL*, 2022. 7

[6] Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO captions: Data collection and evaluation server. *arXiv*, 1504.00325, 2015. 1, 2

[7] Xi Chen, Xiao Wang, Soravit Changpinyo, AJ Piergiovanni, Piotr Padlewski, Daniel Salz, Sebastian Goodman, Adam Grycner, Basil Mustafa, Lucas Beyer, Alexander Kolesnikov, Joan Puigcerver, Nan Ding, Keran Rong, Hassan Akbari, Gaurav Mishra, Linting Xue, Ashish Thapliyal, James Bradbury, Weicheng Kuo, Mojtaba Seyedhosseini, Chao Jia, Burcu Karagol Ayan, Carlos Riquelme, Andreas Steiner, Anelia Angelova, Xiaohua Zhai, Neil Houlsby, and Radu Soricut. PaLI: A jointly-scaled multilingual language-image model. In *ICLR*, 2023. 7

[8] Seongho Choi, Kyoung-Woon On, Yu-Jung Heo, Ahjeong Seo, Youwon Jang, Minsu Lee, and Byoung-Tak Zhang. Dramaqa: Character-centered video story understanding with hierarchical qa. In *American Ass. of Art. Intelligence*, 2021. 3, 7

[9] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Antonino Furnari, Jian Ma, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. Rescaling egocentric vision: Collection, pipeline and challenges for epic-kitchens-100. *IJCV*, 130:33–55, 2022. 2

[10] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al. Scaling egocentric vision: The EPIC-KITCHENS dataset. In *ECCV*, 2018. 1, 4, 5

[11] 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 *ICLR*, 2021. 7

[12] Dave Epstein, Boyuan Chen, and Carl Vondrick. Oops! predicting unintentional action in video. In *CVPR*, 2020. 1, 4, 7

[13] Spandana Gella, Mike Lewis, and Marcus Rohrbach. A dataset for telling the stories of social media videos. In *EMNLP*, 2018. 3

[14] Golnaz Ghiasi, Xiuye Gu, Yin Cui, and Tsung-Yi Lin. Open-vocabulary image segmentation. In *ECCV*, 2022. 2

[15] Cristina González, Nicolás Ayobi, Isabela Hernández, José Hernández, Jordi Pont-Tuset, and Pablo Arbeláez. Panoptic narrative grounding. In *ICCV*, 2021. 3

[16] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, Miguel Martin, Tushar Nagarajan, Ilija Radosavovic, Santhosh Kumar Ramakrishnan, Fiona Ryan, Jayant Sharma, Michael Wray, Mengmeng Xu, Eric Zhongcong Xu, Chen Zhao, Siddhant Bansal, Dhruv Batra, Vincent Cartillier, Sean Crane, Tien Do, Morrie Doulaty, Akshay Erapalli, Christoph Feichtenhofer, Adriano Fragomeni, Qichen Fu, Christian Fuegen, Abrham Gebreselasie, Cristina Gonzalez, James Hillis, Xuhua Huang, Yifei Huang, Wenqi Jia, Weslie Khoo, Jachym Kolar, Satwik Kottur, Anurag Kumar, Federico Landini, Chao Li, Yanghao Li, Zhenqiang Li, Karttikeya Mangalam, Raghava Modhugu, Jonathan Munro, Tullie Murrell, Takumi Nishiyasu, Will Price, Paola Ruiz Puentes, Merey Ramazanova, Leda Sari, Kiran Somasundaram, Audrey Southerland, Yusuke Sugano, Ruijie Tao, Minh Vo, Yuchen Wang, Xindi Wu, Takuma Yagi, Yunyi Zhu, Pablo Arbelaez, David Crandall, Dima Damen, Giovanni Maria Farinella, Bernard Ghanem, Vamsi Krishna Ithapu, C. V. Jawahar, Hanbyul Joo, Kris Kitani, Haizhou Li, Richard Newcombe, Aude Oliva, Hyun Soo Park, James M. Rehg, Yoichi Sato, Jianbo Shi, Mike Zheng Shou, Antonio Torralba, Lorenzo Torresani, Mingfei Yan, and Jitendra Malik. Ego4d: Around the World in 3,000 Hours of Egocentric Video. In *CVPR*, 2022. 1, 2, 4, 5

[17] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 6

[18] Matthew Honnibal and Ines Montani. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. `spacy.io`, 2017. 5, 7

[19] Gabriel Huang, Bo Pang, Zhenhai Zhu, Clara Rivera, and Radu Soricut. Multimodal pretraining for dense video captioning. In *AACL-IJCNLPÏ*, 2020. 3

[20] Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. Referitgame: Referring to objects in photographs of natural scenes. In *EMNLP*, 2014. 1, 2

[21] Anna Khoreva, Anna Rohrbach, and Bernt Schiele. Video object segmentation with language referring expressions. In *ACCV*, 2018. 3, 6

[22] Ivan Krasin, Tom Duerig, Neil Alldrin, Vittorio Ferrari, Sami Abu-El-Haija, Alina Kuznetsova, Hassan Rom, Jasper Ui-

jlings, Stefan Popov, Shahab Kamali, Matteo Malloci, Jordi Pont-Tuset, Andreas Veit, Serge Belongie, Victor Gomes, Abhinav Gupta, Chen Sun, Gal Chechik, David Cai, Zheyun Feng, Dhyanesh Narayanan, and Kevin Murphy. Open-Images: A public dataset for large-scale multi-label and multi-class image classification. *Dataset available from https://g.co/dataset/openimages*, 2017. 5

[23] Jonathan Krause, Justin Johnson, Ranjay Krishna, and Li Fei-Fei. A hierarchical approach for generating descriptive image paragraphs. In *CVPR*, 2017. 1, 2

[24] Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. Dense-captioning events in videos. In *ICCV*, 2017. 3

[25] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, Michael Bernstein, and Li Fei-Fei. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *IJCV*, 123(1):32–73, 2017. 1, 2

[26] Jie Lei, Licheng Yu, Tamara L Berg, and Mohit Bansal. TVQA+: Spatio-temporal grounding for video question answering. In *ACL*, 2019. 3, 7

[27] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft COCO: Common objects in context. In *ECCV*, 2014. 3, 5

[28] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *NeurIPS*, 2019. 1

[29] Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L Yuille, and Kevin Murphy. Generation and comprehension of unambiguous object descriptions. In *CVPR*, 2016. 1, 2

[30] 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. 3

[31] Mathew Monfort, SouYoung Jin, Alexander Liu, David Harwath, Rogerio Feris, James Glass, and Aude Oliva. Spoken moments: Learning joint audio-visual representations from video descriptions. In *CVPR*, 2021. 3

[32] Arsha Nagrani, Paul Hongsuck Seo, Bryan Seybold, Anja Hauth, Santiago Manen, Chen Sun, and Cordelia Schmid. Learning audio-video modalities from image captions. *arXiv:2204.00679*, 2022. 3

[33] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *CVPR*, 2016. 5

[34] Bryan A. Plummer, Liwei Wang, Christopher M. Cervantes, Juan C. Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. *IJCV*, 123(1):74–93, 2017. 1, 2

[35] J. Pont-Tuset, F. Perazzi, S. Caelles, P. Arbeláez, A. Sorkine-Hornung, and L. Van Gool. The 2017 DAVIS challenge on video object segmentation. *arXiv preprint arXiv:1704.00675*, 2017. 5

[36] Jordi Pont-Tuset, Jasper Uijlings, Soravit Changpinyo, Radu Soricut, and Vittorio Ferrari. Connecting vision and language with localized narratives. In *ECCV*, 2020. 1, 2, 4, 5

[37] Jiyang Qi, Yan Gao, Yao Hu, Xinggang Wang, Xiaoyu Liu, Xiang Bai, Serge Belongie, Alan Yuille, Philip Torr, and Song Bai. Occluded video instance segmentation: A benchmark. *IJCV*, 2022. 1, 4, 5

[38] 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. 1

[39] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *ICML*, 2021. 1

[40] Anna Rohrbach, Marcus Rohrbach, Siyu Tang, Seong Joon Oh, and Bernt Schiele. Generating descriptions with grounded and co-referenced people. In *CVPR*, 2017. 4

[41] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet large scale visual recognition challenge. *IJCV*, 2015. 6

[42] Arka Sadhu, Kan Chen, and Ram Nevatia. Video question answering with phrases via semantic roles. In *NAACL*, 2021. 3

[43] Seonguk Seo, Joon-Young Lee, and Bohyung Han. URVOS: Unified referring video object segmentation network with a large-scale benchmark. In *ECCV*, 2020. 3, 6

[44] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *ACL*, 2018. 1, 2

[45] Gunnar A Sigurdsson, Gül Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *ECCV*, 2016. 3

[46] Ashish Thapliyal, Jordi Pont-Tuset, Xi Chen, and Radu Soricut. Crossmodal-3600: A Massively Multilingual Multimodal Evaluation Dataset. In *EMNLP*, 2022. 1, 2

[47] Weiyao Wang, Matt Feiszli, Heng Wang, and Du Tran. Unidentified video objects: A benchmark for dense, open-world segmentation. In *ICCV*, 2021. 1, 4, 5

[48] Zirui Wang, Jiahui Yu, Adams Wei Yu, Zihang Dai, Yulia Tsvetkov, and Yuan Cao. Simvlm: Simple visual language model pretraining with weak supervision. *arXiv preprint arXiv:2108.10904*, 2021. 1

[49] Jiannan Wu, Yi Jiang, Peize Sun, Zehuan Yuan, and Ping Luo. Language as queries for referring video object segmentation. In *CVPR*, 2022. 6

[50] Dejing Xu, Zhou Zhao, Jun Xiao, Fei Wu, Hanwang Zhang, Xiangnan He, and Yueting Zhuang. Video question answering via gradually refined attention over appearance and motion. In *ACM Multimedia*, 2017. 3

[51] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. Msr-vtt: A large video description dataset for bridging video and language. In *CVPR*, 2016. 3

[52] Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. mT5: A massively multilingual pre-trained text-to-text transformer. In *NAACL*, 2021. 7

[53] Antoine Yang, Antoine Miech, Josef Sivic, Ivan Laptev, and Cordelia Schmid. Just ask: Learning to answer questions from millions of narrated videos. In *ICCV*, 2021. 3, 7

[54] Antoine Yang, Antoine Miech, Josef Sivic, Ivan Laptev, and Cordelia Schmid. Learning to answer visual questions from web videos. *PAMI*, 2022. 3

[55] Yunan Ye, Zhou Zhao, Yimeng Li, Long Chen, Jun Xiao, and Yueting Zhuang. Video question answering via attribute-augmented attention network learning. In *SIGIR*, 2017. 3

[56] Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *TACL*, 2:67–78, 2014. 1, 2

[57] 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. In *American Ass. of Art. Intelligence*, 2019. 3, 7

[58] Heeseung Yun, Youngjae Yu, Wonsuk Yang, Kangil Lee, and Gunhee Kim. Pano-avqa: Grounded audio-visual question answering on 360deg videos. In *ICCV*, 2021. 3, 7

[59] Kuo-Hao Zeng, Tseng-Hung Chen, Ching-Yao Chuang, Yuan-Hong Liao, Juan Carlos Niebles, and Min Sun. Leveraging video descriptions to learn video question answering. In *American Ass. of Art. Intelligence*, 2017. 3, 7

[60] Yaoyao Zhong, Wei Ji, Junbin Xiao, Yicong Li, Weihong Deng, and Tat-Seng Chua. Video question answering: Datasets, algorithms and challenges. *arXiv*, 2022. 3, 7

[61] Luowei Zhou, Yannis Kalantidis, Xinlei Chen, Jason J Corso, and Marcus Rohrbach. Grounded video description. In *CVPR*, pages 6578–6587, 2019. 3, 4, 5

[62] Luowei Zhou, Nathan Louis, and Jason J Corso. Weakly-supervised video object grounding from text by loss weighting and object interaction. *arXiv preprint arXiv:1805.02834*, 2018. 1, 3, 4, 5

[63] Luowei Zhou, Chenliang Xu, and Jason J Corso. Towards automatic learning of procedures from web instructional videos. In *American Ass. of Art. Intelligence*, 2018. 3