

# VideoNet: A Large-Scale Dataset for Domain-Specific Action Recognition

Tanush Yadav<sup>1,2</sup> Mohammadreza Salehi<sup>1,2</sup> Jae Sung Park<sup>1,2</sup> Vivek Ramanujan<sup>1</sup>  
 Hannaneh Hajishirzi<sup>1,2</sup> Yejin Choi<sup>3</sup> Ali Farhadi<sup>1,2</sup> Rohun Tripathi<sup>2</sup> Ranjay Krishna<sup>1,2</sup>

<sup>1</sup>University of Washington <sup>2</sup>Allen Institute for AI <sup>3</sup>Stanford University

## Abstract

*Videos are unique in their ability to capture **actions** which transcend multiple frames. Accordingly, action recognition has long been a quintessential task for video models. Unfortunately, due to a lack of sufficiently diverse and challenging data, modern vision-language models (VLMs) are no longer evaluated on their action recognition capabilities. To revitalize action recognition in the era of VLMs, we advocate for a returned focus on **domain-specific** actions. To this end, we introduce VideoNet, a domain-specific action recognition benchmark covering 1,087 distinct actions from 38 domains. VLMs struggle immensely on VideoNet, with Gemini 2.5 Pro performing only 15.8 percentage points better than random chance. To improve model performance we provide in-context demonstrations, but only see a 3% boost in VLM performance compared to a 13% increase in non-expert human accuracy, suggesting that VLMs are poor few-shot learners. At last, we collect a large-scale training dataset containing nearly 500k video question-answer pairs. Fine-tuning an open-weight 4B model on our data, we surpass all Gemini models on the VideoNet benchmark. We release all of our data, inviting the community to explore new techniques to improve domain-specific action recognition capabilities and few-shot learning in video models.*<sup>1</sup>

## 1. Introduction

Action recognition has proven to be an evergreen goal of the computer vision community. As early as 1992, highly-influential works have highlighted the difficulty of recognizing *domain-specific* actions in particular (e.g., [44] focused on categorizing six distinct tennis strokes). Yet domain-specific data is *notoriously difficult* to collect, so little work has been done on gathering domain-specific data across a wide variety of domains. In the era of large vision-language models (VLMs), where testing generalizability is a key con-

cern of many researchers, the lack of diverse domain-specific data has prevented modern VLMs from being evaluated on this “forgotten” task. Instead, the VLM community has focused on fine-grained actions that are *not* domain-specific, such as whether a ball rotates clockwise or counter-clockwise [35]. While such benchmarks are valuable, they fail to capture the real-world applicability of inquiring about domain-specific actions. Furthermore, they only test perception skills, whereas recognizing actions like a “Biellmann spin” in figure skating requires a model to excel at not only perception but also compositional reasoning (i.e., are all elements of this action present and in the correct order?). In this paper, we introduce the data necessary to make domain-specific action recognition relevant in the VLM era.

To this end, we present a benchmark covering over 1,000 actions across 38 domains. We confirm the validity of our test set labels with expert verification, signaling a near 97% accuracy rate in our data.

VLMs struggle on our benchmark. The best open-weight 7B VLM attains 55.6% accuracy, a slight improvement over random chance at 50%. The best proprietary VLM attains 71.5%, barely surpassing non-expert humans at 69.1%. We extensively ablate our visual and textual inputs to the VLMs to understand why models perform poorly on this task. We hypothesize that a lack of domain-specific action data in these models’ training mixtures is primarily responsible.

Inspired by few-shot learning in humans and LLMs [5, 28], we investigate whether this lack of domain-specific training data can be overcome by providing few-shot examples of actions at test-time. Indeed, non-expert human performance improves by 13 percentage points when given 3 few-shot examples. Yet VLMs only see 2-3 percentage point improvements with few-shot examples, suggesting that they are poor few-shot learners and implying that domain-specific action recognition deficiencies cannot be fixed at test time.

Finally, we explore post-training on domain-specific training data. To this end we collect a training set containing 160,000 clips. Fine-tuning a 4B VLM on our data yields a 12 percentage point improvement on VideoNet. Notably, this nears the performance improvement observed in humans

<sup>1</sup>Data, code, and examples at <https://tanu.sh/videonet>. Please refer to the arXiv paper for updated evaluations on a newer version of the VideoNet benchmark.

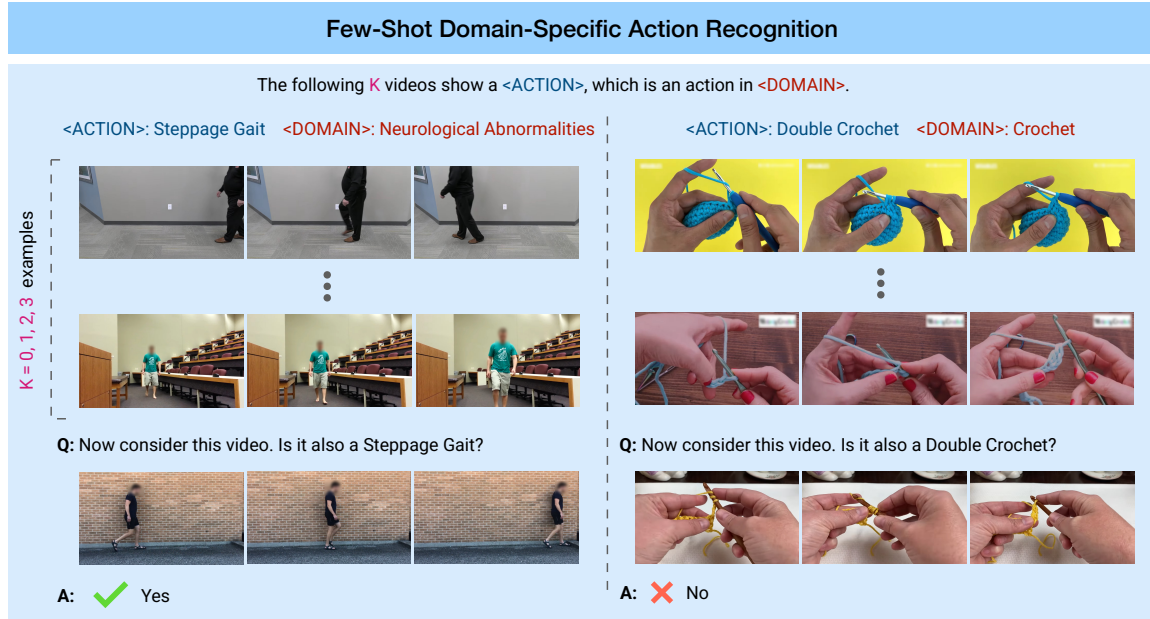


Figure 1. **Examples from the VideoNet benchmark.** Our benchmark includes over 1000 actions across 38 domains, ranging from Crochet to Neurological Abnormalities. It features 5 well-trimmed clips per action. (The prompts shown above are simplified for succinctness.)

when given 3 in-context examples. Our 4B model surpasses all open-weight 7B models and even large proprietary models such as GPT-4o and Gemini 2.5.

We summarize our contributions below.

- A **domain-specific action recognition benchmark** covering 1,087 actions across 38 domains within 7 categories.
- A **domain-specific action training dataset** with 160,000 clips that *enables 4B models to surpass Gemini 2.5 Pro*.
- Two innovative **data pipelines**, for human annotation and synthetic labeling, that break from traditional literature by *entirely circumventing the need for domain experts*.
- Few-shot evaluation of VLMs, highlighting their **deficiencies with in-context learning** relative to LLMs.

Given the expansiveness of our data and the subtle motions innate to domain-specific actions, we are particularly excited about how our data unlocks future research into modeling decisions for perception, visual reasoning, and real-world action understanding.<sup>2</sup>

## 2. Related Work

Action recognition has been extensively explored and fall broadly into three categories. The first set [10, 16, 18, 23, 29, 34, 36] predominantly contain *coarse-grained* labels (e.g., [16] has a single class for "rock climbing", whereas VideoNet contains 23 distinct bouldering actions.) Unsur-

<sup>2</sup>Action understanding is a prerequisite to action quality analysis. Imagine if a VLM could help a new gym goer learn proper squat technique or critique a novice figure skater's lutz jumps.

prisingly, foundation models exceed at recognizing such coarse-grained labels. For instance, InternVideo2 [40] attains 92.1% on Kinetics-400 and 95.9% on ActivityNet. The second set [19, 24, 26, 27, 31, 33, 42, 43] focus on a limited set of sports, rendering them unable to test the generalization promise of foundation models. The third set [7, 8, 22, 35] fixate on fine-grained temporal attributes, such as the direction and trajectory of moving objects. While these works pose interesting perception questions, they focus on details (e.g., does an object move from left to right) that an end user is unlikely to consult a large model for, raising concerns about their real-world utility. Unlike these three groups, VideoNet contains fine-grained labels with real-world applicability across a sufficiently large set of domains.

There are three notable works that collect domain-specific action data across a variety of domains. The first, Ego-Exo4D [15], covers only 8 domains, compared to VideoNet's 38. Our benchmark rivals the size of [15]'s entire dataset, while our training data contains 30x more videos. Perhaps most critically, [15] lacks visual diversity; its 728 bouldering videos, for instance, were filmed at 2 climbing gyms. In contrast, VideoNet sources videos from the web, enabling a great range of visual composition. The second, Ego4D [17], collects fine-grained actions for videos. However, it is restricted to egocentric videos. The third, ActionAtlas [33], collects 934 videos across 56 sports. It is similar to VideoNet in style, but less generalizable due to its exclusive focus on sports. ActionAtlas notably forgoes the question of training data, and even its benchmark is 5 times smaller than VideoNet's.



Figure 2. **Video samples** from all 7 categories and 38 domains in VideoNet. Note the diversity of domains covered by VideoNet.

### 3. Benchmark Construction

#### 3.1. Preparing actions

We employ a top-down approach to generate our taxonomy of actions. First, we formulate a list of categories designed to cover actions that are applicable to daily life (e.g., food), require expert-level knowledge (e.g., medical), and demand a high frame sampling rate for recognizing rapid motions (e.g., sports). Within each category, we find domains that have sufficient videos and trusted expert content online. We then compile actions for each domain from expert-written sources (e.g., skateboarding actions from a respected skateboarding blog) and augment these lists using LLMs (following [33], see Appendix for details). Finally, we search online to verify that we have sufficient potential videos containing each action, removing actions with an insufficient amount of online videos.

Action definitions are used throughout our project to help humans and models classify relevant action videos without specialized domain expertise. To maximize their usefulness to this end, the definitions are written to focus on visual cues or defining characteristics of that action, as well as key differentiators from similar actions. Initially, we used large language models (LLMs) following [33] to generate definitions. However, specialized domains pose challenges,

as LLMs occasionally encode incorrect or outdated domain knowledge [37]. To mitigate this issue, we enable LLMs to perform targeted **web searches** [1], retrieving expert-curated information from reputable online knowledge bases and domain-specific communities. The LLMs then use this information to cross-check and correct inaccuracies, providing a final set of definitions aligned with established domain expertise.

#### 3.2. Collecting well-trimmed clips

Once we have our action lists ready, we launch our three-stage human-annotation pipeline, as visualized in Fig. 3. Our pipeline design is guided by rigorously validated HCI practices [12, 21, 25]. In particular, each clip is reviewed by five distinct annotators throughout the pipeline, and on the second stage the majority vote is taken among three annotators.

**Video collection.** We provide human annotators, sourced from Prolific, with the name of an action within a domain, alongside its definition (§ 3.1). They are told to search for the action online and find seven clips where the action occurs. We require the clips to be sourced from separate videos to increase generalizability.

**Clip verification.** We provide annotators the action name and definition, alongside the seven clips from the previous



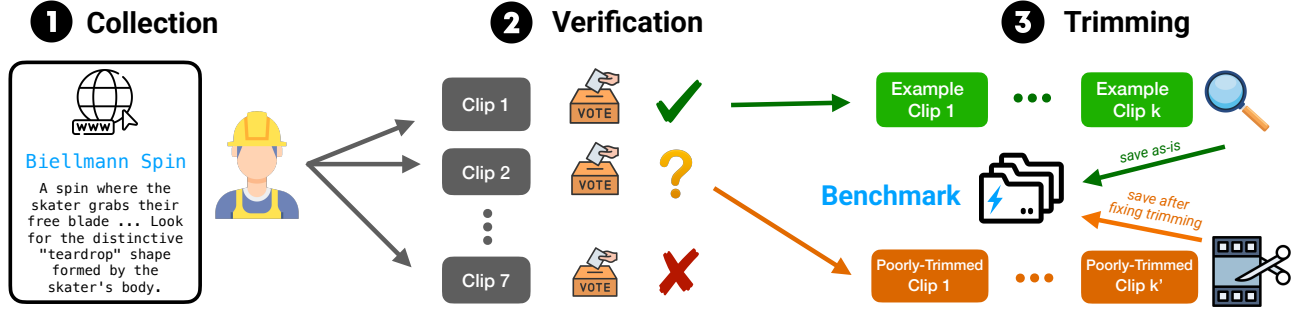


Figure 3. **Benchmark data collection pipeline**, as described in Section 3.2. Given an action name and a definition, humans (1) find clips on the web, (2) remove outliers among these clips, and (3) fix the clip trimmings. This pipeline yields five well-trimmed clips per action.

stage, and ask them to rate each clip as (1) containing the action and being well-trimmed, (2) containing the action but being poorly-trimmed, or (3) not containing the action. Determining if a clip is well-trimmed or not is trivial for humans; however, classifying the clip as containing or not containing the action can be tricky, especially since these annotators are *not* experts in these domains. We solve this dilemma by reducing the problem from  $k$ -way classification to 3-way classification. Where [43] and [27] showed a domain expert a random clip and asked them to classify it as one of  $k$  actions, we ask three non-expert annotators to rate each clip into one of the three categories above and take the majority vote.

**Clip trimming.** We reach this stage with approximately 90% of the actions having 5 or more clips that were deemed to contain the desired action. At least one of these clips was always well-trimmed; in four-fifths of cases, there were at least three well-trimmed clips. To preserve clips that contain the desired action but are poorly trimmed, we conduct an additional trimming stage to refine their temporal boundaries. Here, we show Prolific annotators the action name, action definition, and these well-trimmed examples, thereby training them to be “experts” in the action. We then ask them to fix the trimmings on the poorly-trimmed clips. This leaves us with at least five accurately trimmed clips for the desired action, each verified by annotators trained with relevant in-context examples.

This process yields 1,087 clips, whose statistics are shown in Table 1. Certain domains, like suturing and crochet, contain actions that take longer to demonstrate, causing a noticeable tail in the distribution of video lengths (see Appendix).

### 3.3. Generating (hard) negative examples

With the verified *positive* clips in-hand, we gather suitable *negative* examples to be used in our benchmark. One approach is to gather *random negatives* by randomly sampling clips from different actions within the same domain, but this has notable limitations with different actions often having distinct contexts, backgrounds, or static visual cues. Without

Table 1. **Benchmark Video Duration (seconds)**. The clips are well-trimmed, meaning that they contain the entirety of the action and minimal “fluff” around it. We removed clips longer than 5 minutes after observing open models struggle with context lengths for long videos, especially in the 3-shot setting.

| Mean | Median | Standard Deviation |
|------|--------|--------------------|
| 12.8 | 5.0    | 23.7               |

careful control, models may achieve high performance by exploiting the scene-level details alone (e.g. *alley-oop dunk* vs. *free throw* in basketball), rather than closely watching the entire clip. Instead, we create challenging *hard negatives* by selecting actions that closely resemble the positive clip, only differing in subtle visual or motion-related aspects. Initially, we automatically generate these hard negatives with an LLM (gpt-4.5-preview) following [4, 33]. Different from prior methods, we further refine this candidate set using a reasoning model (o3-mini) by filtering out any clips that could realistically co-occur with the positive action or are otherwise ambiguous. This ensures that our hard negatives are truly valid and challenging (e.g. *alley-oop dunk* vs. *put-back dunk*). Using this strategy, we arrive at a carefully-curated evaluation set consisting of both  $N$  verified positive and  $N$  challenging negative clips for each action (full prompts and refinement details provided in Appendix).

### 3.4. Verifying clip labels

To verify the correctness of our benchmark, we conduct expert verification. We choose one domain from each of our 7 categories for verification, hypothesizing that accuracies for domains within each category should be similar. We verify 620 clips; generalizing human performance from this scale is in line with prior works [32, 45]. When possible, we find experts in our local communities and ask them to verify the data labels, akin to [17, 35, 41]. For domains where we are unable to locate experts, we train someone on a large sample of the domain’s data, before asking them



to verify labels, following [32]. As shown in Table 2, we see 97% accuracy in our data, exceeding MMLU-Pro’s [41] expert accuracy of 85.4% and ImageNet’s [32] estimates of top-5 error at 5.1% and 12.0%. This confirms the validity of our pipeline as a replacement for hiring domain experts during the domain-specific data collection process. It also enables researchers developing future models to confidently use VideoNet as a test bed for domain-specific capabilities.

Table 2. **Expert Verification for VideoNet**. These results confirm that our data collection pipeline is robust to annotator error.

| Category | Domain       | Correct Clips | Total Clips | Percent Correct |
|----------|--------------|---------------|-------------|-----------------|
| Sports   | Tennis       | 92            | 95          | 96.8%           |
| Food     | Coffee       | 75            | 80          | 93.8%           |
| Crafts   | Painting     | 39            | 40          | 97.5%           |
| Medical  | Neuro. Exams | 75            | 75          | 100.0%          |
| Dance    | Break Dance  | 162           | 165         | 98.2%           |
| Hobbies  | Gym          | 107           | 110         | 97.3%           |
| Beauty   | Spa Massage  | 55            | 55          | 100%            |
| All      | All          | 605           | 620         | 97.6%           |

## 4. Model Training

We create a large-scale training dataset using a fully automated pipeline for our domain-specific action recognition task. By further training an open VLM, Molmo2-4B [11], on this dataset, we demonstrate a significant improvement in the base model’s performance for our task.

### 4.1. Training Data

While the data collection pipeline described in Section 3.2 leads to extremely high-quality clips, its reliance on human annotators renders it *prohibitively expensive* for collecting training-scale data. A common solution in such cases is to rely on synthetic labels generated by foundation models [13, 20]. As shown in Section 5.1, VLMs struggle to recognize domain-specific actions, so distilling directly from even the best-performing VLM is unideal. Instead, we choose to rely on signals contained within the video data, specifically in the video’s title and transcript.

We build up our training data one domain at a time for each of the 38 domains covered by our benchmark. For a given domain, we begin by crawling relevant videos from the web. To do so, we construct queries from our action list, e.g. from “laser flip” we construct queries like “skateboarding laser flip” and “how to laser flip”. Once we have a pool of relevant videos, we extract clips using Gemini 2.5 Flash as a localizer. For instance, we ask Gemini to provide start and end timestamps for each clip where a skateboarding action occurs. Critically, even though Gemini struggles to *label* the actions in these clips, it excels at *localizing* them. This is likely because the localization task can be performed by leveraging signals such as the speech and scene changes

within the video. Once we have a set of domain-specific action clips extracted from our pool of domain-specific videos, we must filter and label these clips. The video’s audio can be helpful for labeling clips, so we extract word-level timestamps using whisperX [3]. We experiment with three filtering and labeling strategies.

1. **TRANSCRIPTLOCALIZED**. If an action name appears in the video’s transcript within  $T = 1$  seconds of a localized clip, the clip is labeled with that action.
2. **TRANSCRIPTLOCALIZEDTITLEMATCH**. Refining on top of **TRANSCRIPTLOCALIZED**, we further require the action appear in the video’s title.
3. **SINGLEACTION**. If an action appears in the video’s title, and the localizer identified *only one clip* in the entire video, that clip is labeled with the action from the title.

In total, we crawl 8 million videos before localizing and transcribing 1.5 million videos. This results in 6 million clips, which we then filter into training sets ranging in size from 160,000 clips to 500,000 clips. Training results for the different data filtering strategies are provided in Sec. 5.3.

### 4.2. Training Details

We fine-tune Molmo2-4B, an instruction-tuned VLM on our filtered training dataset. The model’s architecture follows the popular paradigm of connecting a vision transformer (ViT) [14] to a pre-trained language model (LLM) via an MLP connector module [11]. During training, the frames are sampled up to max sampling rate of  $S$  and up to max frames  $F$ . If the video has more than  $F/S$  frames, we use uniformly sample  $F$  frames. To preserve the temporal information, we encode the frame timestamp in seconds before each sampled frame as text input to the LLM. We train the model for 8K steps, with a batch size of 128, max frames  $F = 64$  and a maximum sampling rate  $S = 4$  in all our experiments. Additional training details are reported in the Appendix.

## 5. Experiments

We evaluate state-of-the-art video-language models on VideoNet, finding that the best models rival human performance in a zero-shot setting but lag considerably behind humans when given in-context examples.

For open models, we use InternVL3-8B [48], Qwen2.5-VL-7B [2], and LLaVA-Video-7B [47]. For proprietary models, we use Gemini 2.5 Flash, Gemini 2.5 Pro, GPT-4o, GPT-4.1, and GPT-5.<sup>3</sup> A discussion of their context lengths—which may impact few-shot performance—is in the Appendix. Also in the Appendix are results for Qwen2.5-VL-72B, CLIP models [30, 38, 39, 46], and optical flow models [9].

When passing videos as inputs, we use the models’ official sampling strategies: uniform sampling for LLaVA-Video-7B (max 110 frames) and InternVL3-8B (max 64

<sup>3</sup>Snapshots/versions of proprietary models available in Appendix.

Table 3. **Zero-shot evaluation results.** Most open-source models performance little better than chance on the benchmark. We derive the “Molmo2-4B (FT)” model by fine-tuning the Molmo2-4B Instruct model on our **SINGLEACTION** dataset. This boosts performance by about 12%-points. Highest overall accuracy for each column is in bold; second-highest is underlined.

|                    | Model Name            | Beauty       | Crafts       | Dance        | Food         | Hobbies      | Medical      | Sports       | Overall      |
|--------------------|-----------------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
|                    | Random                | 50.00        | 50.00        | 50.00        | 50.00        | 50.00        | 50.00        | 50.00        | 50.00        |
| <i>Closed</i>      | Gemini 2.5 Flash      | 70.18        | 72.69        | 59.90        | 86.05        | 63.85        | 69.15        | 60.72        | 65.14        |
|                    | Gemini 2.5 Pro        | <u>74.19</u> | 73.51        | 62.26        | <u>87.37</u> | 62.71        | 72.36        | 60.99        | 65.78        |
|                    | GPT-4o                | 71.90        | 73.25        | 61.10        | 86.49        | 63.79        | 66.15        | 65.72        | 66.76        |
|                    | GPT-4.1               | 73.39        | <u>75.00</u> | 64.18        | <u>87.37</u> | 65.57        | <u>74.00</u> | <u>67.59</u> | <u>69.02</u> |
|                    | GPT-5                 | <b>75.00</b> | <b>77.53</b> | <b>70.07</b> | <b>88.40</b> | <b>67.61</b> | <b>79.50</b> | <b>68.32</b> | <b>71.51</b> |
| <i>Open Weight</i> | InternVL3-8B          | 54.03        | 51.12        | 54.69        | 64.18        | 47.25        | 54.00        | 51.63        | 52.16        |
|                    | Qwen2.5-VL-7B         | 50.00        | 51.20        | 50.00        | 72.16        | 55.58        | 58.00        | 53.26        | 55.01        |
|                    | LLaVA-Video-7B        | 58.87        | 57.84        | 51.32        | 70.36        | 54.74        | 58.00        | 54.89        | 55.98        |
| <i>Fully Open</i>  | Molmo2-4B (base)      | 50.00        | 51.20        | 50.00        | 72.16        | 55.58        | 58.00        | 53.26        | 55.01        |
|                    | <b>Molmo2-4B (FT)</b> | <b>75.00</b> | 69.66        | <u>66.84</u> | 76.03        | <u>66.51</u> | 71.00        | 66.33        | 67.36        |

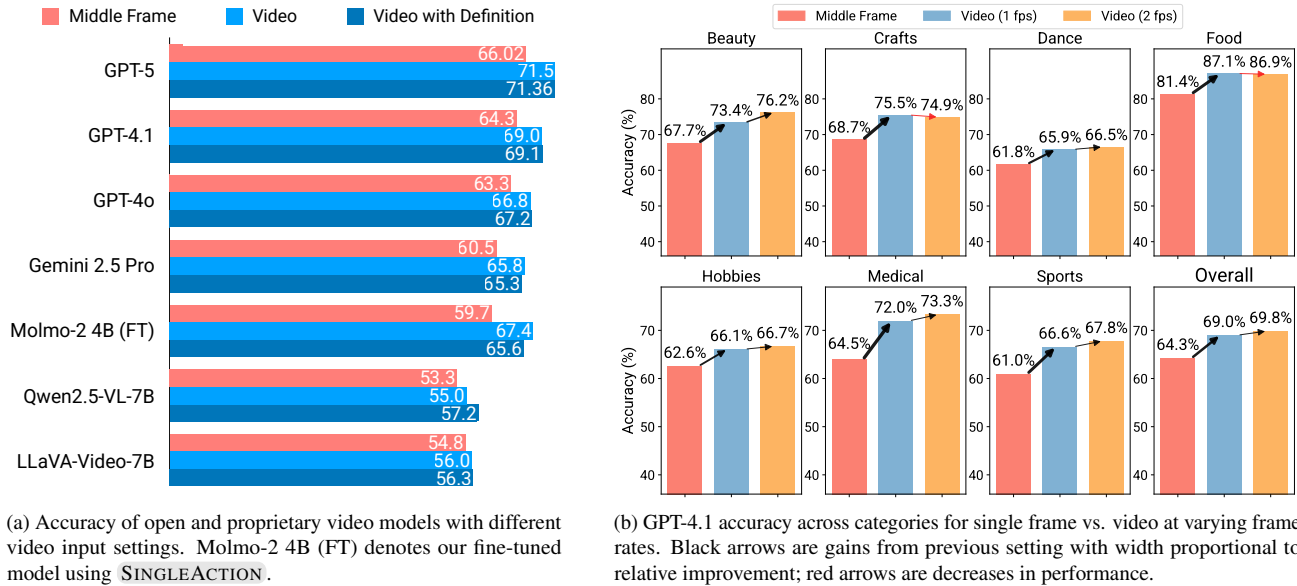


Figure 4. **Ablations on video input configurations.** Open models show limited gains from full-video input, indicating difficulty in effectively leveraging video context. (A notable exception is our model, which benefits significantly.) GPT-4.1, the GPT model with the longest context length, shows minimal improvement at higher fps, suggesting that it faces challenges in capturing subtle or rapid motions.

frames); fps sampling for Qwen2.5-VL, GPT, and Gemini 2.5 (1 fps). For our model, we use 4 fps sampling up to a max of 64 frames, with frame ablations in the Appendix.

### 5.1. Zero-shot evaluation

Given a video along with an action name and domain, we prompt models to answer whether or not the video depicts the specified action. We use a balanced set of  $N = 2$  positive and  $N = 2$  negative clips per action (Figure 1). Models are optionally prompted to explicitly reason or analyze the video before providing their answer. We use binary accuracy as our

metric, where random chance gets 50%. Table 3 shows that previous open models generally struggle with fine-grained action recognition, performing only slightly above random chance. Interestingly, LLaVA-Video-7B achieves the highest overall accuracy among the previous open models (55.98%), outperforming both InternVL3-8B (52.16%) and Qwen2.5-VL-7B (55.01%), despite the latter models typically performing better in standard video benchmarks. Comparing across categories, LLaVA-Video-7B notably excels in the Beauty and Crafts categories, whereas Qwen2.5-VL-7B is the strongest in the Food category. These divergent results

suggest that previous open models have varied expertise across domains, implying that systematic evaluations can help identify their specific weaknesses and inform which domains to prioritize in future training. In contrast, the model fine-tuned on our data outperforms previous open models by a significant margin. See § 5.3 for more details.

Next, we see that closed-source proprietary models consistently achieve higher performance than open-source models across all categories. GPT-5 achieves the highest overall accuracy, 71.51%, and clearly outperforms both GPT-4.1 at 69.02% and GPT-4o at 66.76%, whereas Gemini 2.5 Pro and Flash perform similarly (65.78% vs. 65.14%). Notably, the Dance and Medical categories benefit most from stronger reasoning capabilities in advanced models (GPT-4.1 vs. GPT-4o; Gemini 2.5 Pro vs. Flash). Categories such as Beauty, Food, and Craft which involve less rapid motion consistently show higher model performance compared to Dance, Hobbies, and Sports, suggesting that visual cues in the former group may be easier for models to recognize.

We conduct ablation studies to identify factors limiting model performance. Specifically, we examine whether performance issues arise from insufficient motion understanding or limited action knowledge (Figure 4). Figure 4a compares performances when provided with (1) a single middle frame, (2) the entire video (default setup), and (3) the video with action definitions (see § 3.1). Figure 4b shows GPT-4.1 accuracy across categories and varied frame rates. Detailed category-level results are available in the Appendix, as are 4fps results. Here, we analyze the following:

**Image bias vs. motion understanding.** Existing open-source models show only slight improvements when moving from a single middle frame to the entire video, implying that they struggle to effectively ground actions in detailed motion cues and instead rely heavily on static visual biases (Figure 4a). In contrast, our fine-tuned model and all proprietary models benefit from full-video input indicating their stronger capability to utilize video information for action recognition.

**Impact of action definitions.** Figure 4a shows that providing explicit action definitions yields minimal gains for proprietary models. This is likely because these models already possess sufficient inherent knowledge about actions, likely comparable to expert community sources from the web, and their primary limitation is effectively mapping this knowledge to subtle motion details. In contrast, Qwen2.5-VL-7B shows a moderate improvement from definitions (55.0% to 57.2%), indicating that open models can still benefit from additional action-contextual information.

**Higher FPS.** Across action categories, GPT-4.1 significantly improves from single-frame to full-video inputs (see Figure 4b). However, increasing the fps further does not improve results, even in motion-intensive categories (Dance, Hobbies, Sports), suggesting that models fail to leverage higher temporal resolution for capturing subtle or rapid motions.

## 5.2. Few-shot evaluation

Prior studies have shown that LLMs can greatly benefit from well-curated in-context examples [5, 28]. We ask whether current video-language models share this capability when it comes to processing visual information. To investigate this, we evaluate each model’s performance when provided with up to  $k = 3$  ground truth video demonstrations as *visual* in-context examples. These in-context examples are drawn separately and are excluded from the original zero-shot evaluation set for fairness, but we reuse the same zero-shot evaluation set to ensure consistency.

**Video models fail to effectively utilize in-context examples.** Figure 5 shows overall model accuracy as more in-context examples are provided (exact numbers in Appendix). We see that open models improve modestly from 0-shot to 3-shot conditions. Among proprietary models, GPT-4o displays steady improvement (from 66.76% at 0-shot to 70.12% at 3-shot), surpassing zero-shot GPT-4.1. Meanwhile GPT-4.1 improves notably at first (+2.6% from 0-shot to 1-shot), but quickly saturates with additional examples (71.7% at 1-shot to 72.7% at 3-shot). GPT-5, the best zero-shot model, sees only a 1% improvement from 0-shot to 3-shot. This trend suggests that as the base model gets stronger and has a better innate understanding of the action at hand, its ability to effectively utilize few-shot examples decreases. Surprisingly, Gemini 2.5 Pro shows a slight decline after one-shot example, possibly due to context length overload introduced by more in-context examples. Overall, the slight few-shot gains—especially when compared to the substantial gains yielded by post-training on our dataset (see § 5.3)—suggest that video models struggle to fully exploit visual demonstrations. This is likely because their training focuses on scenarios with single-video inputs rather than multiple in-context videos.

**Humans are far more effective few-shot learners.** To better understand the upper bounds of our few-shot setup, we conduct human evaluations. We sample 698 questions across our dataset and ask three annotators to answer each binary question, taking the majority vote as the final answer (details in Appendix). Figure 5 shows that in the zero-shot setup, **average humans without definition perform worse than the proprietary models**, likely due to limited domain knowledge. When given definitions, zero-shot human performance improves (+4.5%) to 69.1%, comparable to GPT-4.1. We see the most striking difference in the 3-shot setting, where **humans 3-shot with definitions improve significantly** to 82.7% (+13.6% from 0-shot), while even those without definitions achieve 78.8% with examples alone. This suggests that humans are highly efficient few-shot learners, quickly generalizing visual patterns from few demonstrations. The large gap between human and model few-shot performance indicates current video models may lack the perceptual mechanisms underlying such human visual learning [6].

**Random vs. hard negatives.** While humans achieve high



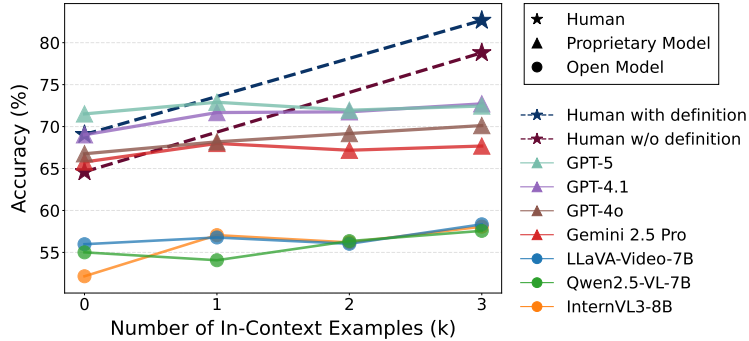


Figure 5. **Few-shot accuracy** of video models and humans with  $k$  in-context video demonstrations. Humans benefit significantly more than models do.

Table 4. **Accuracy with hard vs. random negatives.** As intended, our selection of hard negatives makes the benchmark more difficult, especially for humans.

| Models                | Hard |      | Random |      |
|-----------------------|------|------|--------|------|
|                       | k=0  | k=3  | k=0    | k=3  |
| Qwen2.5-VL-7B         | 55.0 | 57.6 | 57.7   | 60.5 |
| LLaVA-Video-7B        | 56.0 | 58.3 | 60.2   | 62.5 |
| InternVL3-8B          | 52.2 | 56.9 | 58.1   | 61.1 |
| Gemini 2.5 Pro        | 65.8 | 67.8 | 72.9   | 73.3 |
| GPT-4o                | 66.8 | 70.1 | 75.2   | 76.3 |
| GPT-4.1               | 69.0 | 72.7 | 75.9   | 77.4 |
| GPT-5.1               | 71.5 | 72.5 | 77.5   | 76.2 |
| Human with definition | 69.1 | 82.7 | 81.5   | 93.5 |

Table 5. **Data Filtering Strategy.** Performance on post-trained Molmo2 [11] using different data filtering strategies. Bold represents best in column. At large scales, data quality seems most important. At small scales, data coverage seems to play a larger factor.

| Filter                        | Size | Beauty | Crafts | Dance | Food  | Hobbies | Medical | Sports | Overall |
|-------------------------------|------|--------|--------|-------|-------|---------|---------|--------|---------|
| Molmo2-4B (base)              | -    | 50.00  | 51.20  | 50.00 | 72.16 | 55.58   | 58.00   | 53.26  | 55.01   |
| SINGLEACTION                  | 163K | 75.00  | 69.66  | 66.84 | 76.03 | 66.51   | 71.00   | 66.33  | 67.36   |
| TRANSCRIPTLOCALIZED           | 501K | 74.19  | 63.72  | 64.97 | 78.35 | 66.51   | 69.50   | 59.32  | 65.11   |
| TRANSCRIPTLOCALIZEDTITLEMATCH | 208K | 73.39  | 62.35  | 64.29 | 75.26 | 63.90   | 63.50   | 61.36  | 64.21   |

few-shot accuracy (82.7%), their performance remains imperfect. One plausible cause is the lack of domain-specific expertise needed to distinguish difficult negatives, as we observe that few-shot human annotators achieve 94.4% on positive examples but only 71.9% on our hard negatives (see Appendix). We compare performance on our default hard vs. random negatives (actions randomly chosen within the same domain, as described in § 3.3), and report results in Table 4. As expected, **random negatives yield consistently higher performance for models and humans in both zero-shot and few-shot settings**, with GPT-4.1 (3-shot) performing best at 77.4%. Meanwhile, humans achieve 93.5% at 3-shot with much higher gains in performance. The accuracy drop from random to hard negatives – especially for humans – suggests that VideoNet **contains challenging, fine-grained visual distinctions that require expertise to solve**.

### 5.3. Training Results

We finetune a Molmo2-4B [11] model using our filtered data according to strategies detailed in Section 4.1. As reported in Table 5, training with any of our subsets improves results over the base model, substantiating the claim that open-source models suffer from a lack of domain-specific action training data. Using **SINGLEACTION** as the filtering strategy provides the most gain over the base model, improving by roughly 12 percentage points over the base model. Notably, all of our models beat all open-weight 7B models, and our best two models surpass all Gemini models.

Our results indicate that the quality of data *generally* influences the performance more than the quantity, since **SINGLEACTION** is the strictest filter—in terms of samples selected—and leads to the highest accuracy. However, for domains in the long-tail, *coverage* becomes an important factor. For instance, **SINGLEACTION** yields only 348 clips for juggling, whereas **TRANSCRIPTLOCALIZED** yields 1,582. Indeed, the juggling accuracy for the model trained on the latter filter surpasses that for the model trained on the former filter (64.42% vs. 62.50%; refer to Appendix.) It is unclear the scale at which coverage becomes less important than quality; we leave this to future work.

## 6. Conclusion

We introduce VideoNet, a benchmark to evaluate domain-specific, fine-grained action understanding of large-video language models. Our findings reveal that models struggle with recognizing actions in the zero-shot setup. In order to improve models, we collect a training dataset of automatically labeled clips of fine-grained, domain-specific actions. Post-training a 4B VLM model on this data surpasses all Gemini models and GPT-4o on our benchmark. Further, we explore a few-shot evaluation setting where even the best-performing model, GPT-5, struggles, implying that video language models are currently not effective few-shot learners—unlike their text-only counterparts. With this benchmark, we aim to inspire future work improving these models’s fine-grained recognition and few-shot learning capabilities.

## 7. Acknowledgements

This project was partially funded by a grant from Apple.

We thank the Hyak and Beaker teams at UW and Ai2 for maintaining their respective compute clusters.

We thank Oncel Tuzel and Chun-Liang Li for their feedback and guidance.

We thank members of the UW RAIIVN Lab and the Ai2 PRIOR team for insightful discussions and morale boosts, including but not limited to (in alphabetical order) Chris Dongjoo Kim, Etash Guha, Ethan Shen, George Stoica, Haoquan Fang, Jason Lee, Kevin Farhat, Kevin Zhang, Madeline Brumley, Matthew Wallingford, Peter Sushko, and Sarah Pratt. The author likewise thanks Hayoung Jung.

## References

- [1] Anthropic. Claude can now search the web, 2025. [3](#)
- [2] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report, 2025. [5](#)
- [3] Max Bain, Jaesung Huh, Tengda Han, and Andrew Zisserman. Whisperx: Time-accurate speech transcription of long-form audio. *INTERSPEECH 2023*, 2023. [5](#)
- [4] Hritik Bansal, Yonatan Bitton, Idan Szepktor, Kai-Wei Chang, and Aditya Grover. Videocon: Robust video-language alignment via contrast captions, 2023. [4](#)
- [5] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020. [1](#), [7](#)
- [6] Giovanni Buccino, Ferdinand Binkofski, and Lucia Riggio. The mirror neuron system and action recognition. *Brain and Language*, 89:370–376, 2004. [7](#)
- [7] James Burgess, Xiaohan Wang, Yuhui Zhang, Anita Rau, Alejandro Lozano, Lisa Dunlap, Trevor Darrell, and Serena Yeung-Levy. Video action differencing, 2025. [2](#)
- [8] Mu Cai, Reuben Tan, Jianrui Zhang, Bocheng Zou, Kai Zhang, Feng Yao, Fangrui Zhu, Jing Gu, Yiwu Zhong, Yuzhang Shang, Yao Dou, Jaden Park, Jianfeng Gao, Yong Jae Lee, and Jianwei Yang. Temporalbench: Towards fine-grained temporal understanding for multimodal video models. *arXiv preprint arXiv:2410.10818*, 2024. [2](#)
- [9] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset, 2018. [5](#)
- [10] Joao Carreira, Eric Noland, Chloe Hillier, and Andrew Zisserman. A short note on the kinetics-700 human action dataset, 2022. [2](#)
- [11] Christopher Clark, Jieyu Zhang, Zixian Ma, Jae Sung Park, Rohun Tripathi, Sangho Lee, Mohammadreza Salehi, Jason Ren, Chris Dongjoo Kim, YINUO Yang, Vincent Shao, Yue Yang, Weikai Huang, Ziqi Gao, Taira Anderson, Jianrui Zhang, Jitesh Jain, George Stoica, Ali Farhadi, and Ranjay Krishna. Molmo 2: Open weights and open data for state-of-the-art video and image models, 2025. Technical Report. [5](#), [8](#)
- [12] Florian Daniel, Pavel Kucherbaev, Cinzia Cappiello, Boualem Benatallah, and Mohammad Allahbakhsh. Quality control in crowdsourcing: A survey of quality attributes, assessment techniques, and assurance actions. *ACM Comput. Surv.*, 51(1), 2018. [3](#)
- [13] Bosheng Ding, Chengwei Qin, Linlin Liu, Yew Ken Chia, Boyang Li, Shafiq Joty, and Lidong Bing. Is GPT-3 a good data annotator? In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11173–11195, Toronto, Canada, 2023. Association for Computational Linguistics. [5](#)
- [14] 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. [5](#)
- [15] Kristen Grauman et al. Ego-exo4d: Understanding skilled human activity from first- and third-person perspectives, 2024. [2](#)
- [16] 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. [2](#)
- [17] 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, Abraham Gebreselasie, Cristina González, James Hillis, Xuhua Huang, Yifei Huang, Wenqi Jia, Weslie Khoo, Jáchym Kolář, 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, Merey Ramazanova, Leda Sari, Kiran Somasundaram, Audrey Southerland, Yusuke Sugano, Ruijie Tao, Minh Vo, Yuchen Wang, Xindi Wu, Takuma Yagi, Ziwei Zhao, Yunyi Zhu, Pablo Arbeláez, David Crandall, Dima Damen, Giovanni Maria Farinella, Christian Fuegen, 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18995–19012, 2022. 2, 4
- [18] Chunhui Gu, Chen Sun, David A. Ross, Carl Vondrick, Caroline Pantofaru, Yeqing Li, Sudheendra Vijayanarasimhan, George Toderici, Susanna Ricco, Rahul Sukthankar, Cordelia Schmid, and Jitendra Malik. Ava: A video dataset of spatio-temporally localized atomic visual actions, 2018. 2
- [19] Xiaofan Gu, Xinwei Xue, and Feng Wang. Fine-grained action recognition on a novel basketball dataset. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2563–2567, 2020. 2
- [20] Etash Guha, Ryan Marten, Sedrick Keh, Negin Raoof, Georgios Smyrnis, Hritik Bansal, Marianna Nezhurina, Jean Mercat, Trung Vu, Zayne Sprague, Ashima Suvarna, Benjamin Feuer, Liangyu Chen, Zaid Khan, Eric Frankel, Sachin Grover, Caroline Choi, Niklas Muennighoff, Shiye Su, Wanjia Zhao, John Yang, Shreyas Pimpalgaonkar, Kartik Sharma, Charlie Cheng-Jie Ji, Yichuan Deng, Sarah Pratt, Vivek Ramanujan, Jon Saad-Falcon, Jeffrey Li, Achal Dave, Alon Albalak, Kushal Arora, Blake Wulfe, Chinmay Hegde, Greg Durrett, Sewoong Oh, Mohit Bansal, Saadia Gabriel, Aditya Grover, Kai-Wei Chang, Vaishaal Shankar, Aaron Gokaslan, Mike A. Merrill, Tatsunori Hashimoto, Yejin Choi, Jenia Jitsev, Reinhard Heckel, Maheswaran Sathiamoorthy, Alexandros G. Dimakis, and Ludwig Schmidt. Openthoughts: Data recipes for reasoning models, 2025. 5
- [21] Derek L. Hansen, Patrick J. Schone, Douglas Corey, Matthew Reid, and Jake Gehring. Quality control mechanisms for crowdsourcing: peer review, arbitration, & expertise at familysrch indexing. In *Proceedings of the 2013 Conference on Computer Supported Cooperative Work*, page 649–660, New York, NY, USA, 2013. Association for Computing Machinery. 3
- [22] Wenyi Hong\*, Yean Cheng\*, Zhuoyi Yang\*, Weihai Wang, Lefan Wang, Xiaotao Gu, Shiyu Huang, Yuxiao Dong, and Jie Tang. Motionbench: Benchmarking and improving fine-grained video motion understanding for vision language models, 2024. 2
- [23] Hildegard Kuehne, Hueihan Jhuang, Estíbaliz Garrote, Tomaso Poggio, and Thomas Serre. Hmdb: a large video database for human motion recognition. In *2011 International conference on computer vision*, pages 2556–2563. IEEE, 2011. 2
- [24] H. Kuehne, A. B. Arslan, and T. Serre. The language of actions: Recovering the syntax and semantics of goal-directed human activities. In *Proceedings of Computer Vision and Pattern Recognition Conference (CVPR)*, 2014. 2
- [25] Hongwei Li and Bin Yu. Error rate bounds and iterative weighted majority voting for crowdsourcing, 2014. 3
- [26] Yingwei Li, Yi Li, and Nuno Vasconcelos. Resound: Towards action recognition without representation bias. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018. 2
- [27] 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 (ICCV)*, pages 13536–13545, 2021. 2, 4
- [28] Sewon Min, Mike Lewis, Luke Zettlemoyer, and Hannaneh Hajishirzi. MetaICL: Learning to learn in context. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2791–2809, Seattle, United States, 2022. Association for Computational Linguistics. 1, 7
- [29] Mathew Monfort, Alex Andonian, Bolei Zhou, Kandan Ramakrishnan, Sarah Adel Bargal, Tom Yan, Lisa Brown, Quanfu Fan, Dan Gutfrueud, Carl Vondrick, and Aude Oliva. Moments in time dataset: one million videos for event understanding, 2019. 2
- [30] Bolin Ni, Houwen Peng, Minghao Chen, Songyang Zhang, Gaofeng Meng, Jianlong Fu, Shiming Xiang, and Haibin Ling. Expanding language-image pretrained models for general video recognition. 2022. 5
- [31] Marcus Rohrbach, Anna Rohrbach, Michaela Regneri, Sikan-dar Amin, Mykhaylo Andriluka, Manfred Pinkal, and Bernt Schiele. Recognizing fine-grained and composite activities using hand-centric features and script data. *International Journal of Computer Vision*, 119(3):346–373, 2015. 2
- [32] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge, 2015. 4, 5
- [33] Mohammadreza Salehi, Jae Sung Park, Tanush Yadav, Aditya Kusupati, Ranjay Krishna, Yejin Choi, Hannaneh Hajishirzi, and Ali Farhadi. Actionatlas: A videoqa benchmark for domain-specialized action recognition, 2024. 2, 3, 4
- [34] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: a local svm approach. In *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, pages 32–36 Vol.3, 2004. 2
- [35] Ziyao Shangguan, Chuhan Li, Yuxuan Ding, Yanan Zheng, Yilun Zhao, Tesca Fitzgerald, and Arman Cohan. Tomato: Assessing visual temporal reasoning capabilities in multimodal foundation models, 2024. 1, 2, 4
- [36] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild, 2012. 2
- [37] S. M Towhidul Islam Tonmoy, S M Mehedi Zaman, Vinija Jain, Anku Rani, Vipula Rawte, Aman Chadha, and Amitava Das. A comprehensive survey of hallucination mitigation techniques in large language models. *ArXiv*, abs/2401.01313, 2024. 3
- [38] Jiapeng Wang, Chengyu Wang, Kunzhe Huang, Jun Huang, and Lianwen Jin. Videoclip-xl: Advancing long description understanding for video clip models, 2024. 5
- [39] Yi Wang, Yanan He, Yizhuo Li, Kunchang Li, Jiashuo Yu, Xin Ma, Xinhao Li, Guo Chen, Xinyuan Chen, Yaohui Wang, et al. Internvid: A large-scale video-text dataset for multimodal understanding and generation. In *ICLR*, 2023. 5



- [40] Yi Wang, Kunchang Li, Xinhao Li, Jiashuo Yu, Yinan He, Chenting Wang, Guo Chen, Baoqi Pei, Rongkun Zheng, Jilan Xu, Zun Wang, et al. Internvideo2: Scaling video foundation models for multimodal video understanding. *arXiv preprint arXiv:2403.15377*, 2024. [2](#)
- [41] Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyang Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhui Chen. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark, 2024. [4](#), [5](#)
- [42] Jinglin Xu, Yongming Rao, Xumin Yu, Guangyi Chen, Jie Zhou, and Jiwen Lu. Finediving: A fine-grained dataset for procedure-aware action quality assessment, 2022. [2](#)
- [43] Jinglin Xu, Guohao Zhao, Sibao Yin, Wenhao Zhou, and Yuxin Peng. Finesports: A multi-person hierarchical sports video dataset for fine-grained action understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21773–21782, 2024. [2](#), [4](#)
- [44] J. Yamato, J. Ohya, and K. Ishii. Recognizing human action in time-sequential images using hidden markov model. In *Proceedings 1992 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 379–385, 1992. [1](#)
- [45] Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruochi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, Cong Wei, Botao Yu, Ruibin Yuan, Renliang Sun, Ming Yin, Boyuan Zheng, Zhenzhu Yang, Yibo Liu, Wenhao Huang, Huan Sun, Yu Su, and Wenhui Chen. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi, 2024. [4](#)
- [46] Beichen Zhang, Pan Zhang, Xiaoyi Dong, Yuhang Zang, and Jiaqi Wang. Long-clip: Unlocking the long-text capability of clip. *arXiv preprint arXiv:2403.15378*, 2024. [5](#)
- [47] Yuanhan Zhang, Jinming Wu, Wei Li, Bo Li, Zejun Ma, Ziwei Liu, and Chunyuan Li. Video instruction tuning with synthetic data, 2024. [5](#)
- [48] Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, Zhangwei Gao, Erfei Cui, Xuehui Wang, Yue Cao, Yangzhou Liu, Xingguang Wei, Hongjie Zhang, Haomin Wang, Weiye Xu, Hao Li, Jiahao Wang, Nianchen Deng, Songze Li, Yinan He, Tan Jiang, Jiapeng Luo, Yi Wang, Conghui He, Botian Shi, Xingcheng Zhang, Wenqi Shao, Junjun He, Yingtong Xiong, Wenwen Qu, Peng Sun, Penglong Jiao, Han Lv, Lijun Wu, Kaipeng Zhang, Huipeng Deng, Jiaye Ge, Kai Chen, Limin Wang, Min Dou, Lewei Lu, Xizhou Zhu, Tong Lu, Dahua Lin, Yu Qiao, Jifeng Dai, and Wenhui Wang. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models, 2025. [5](#)